KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

A geometric approach to symmetric-key cryptanalysis

Tim Beyne

Supervisor:
Prof. dr. ir. Vincent Rijmen

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

June 2023

A geometric approach to symmetric-key cryptanalysis

Tim BEYNE

Examination committee:
Prof. dr. ir. Paul Sas, chair
Prof. dr. ir. Vincent Rijmen, supervisor
Prof. dr. ir. Hugo Van hamme
Prof. dr. ir. Fréderik Vercauteren
Dr. Anne Canteaut
(INRIA-Paris, France)
Prof. dr. ir. Joan Daemen
(Radboud Universiteit, The Netherlands)
Prof. dr. Gregor Leander
(Ruhr-Universität Bochum, Germany)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

June 2023

Ter herinnering aan Karel Vanelstraete

Preface

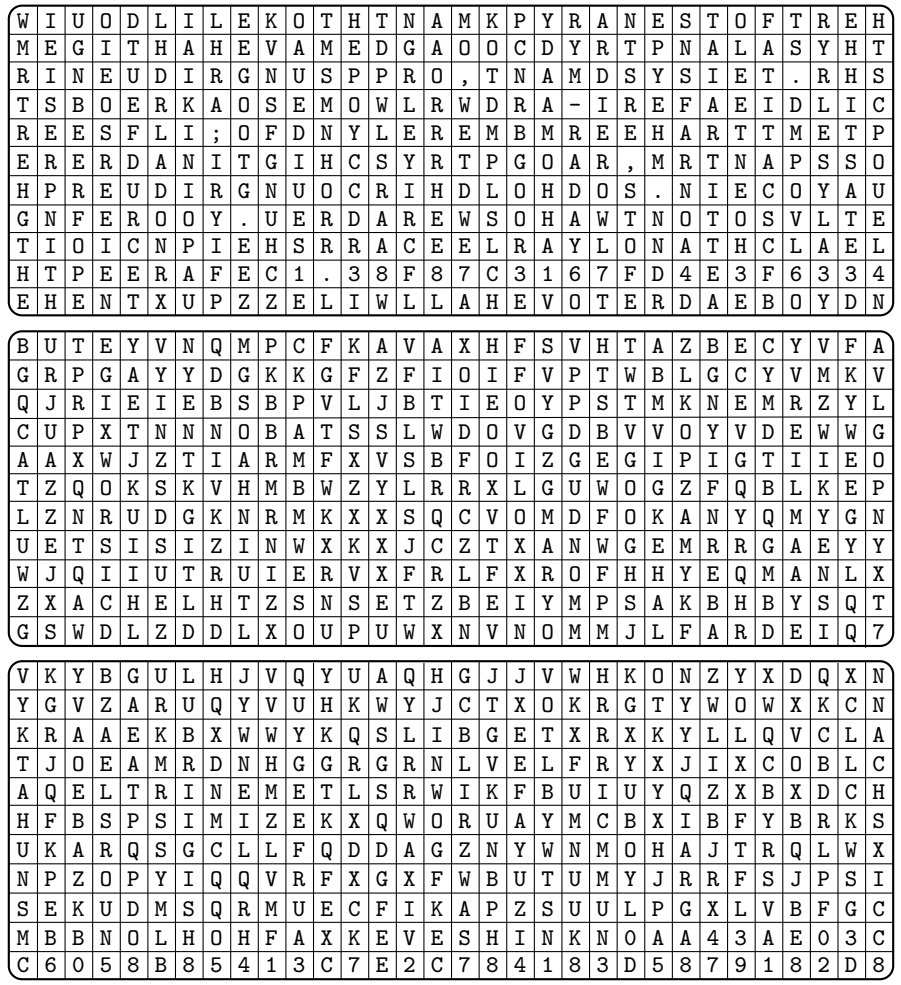
When Edward Snowden betrayed the world's largest cryptanalytic organization in June 2013, he concluded his missive to journalists with the words "Let us speak no more of faith in man, but bind him down from mischief by the chains of cryptography". True or not, they left their mark on a high-schooler interested in cryptography. Following a sequence of events so unlikely that their description cannot be compressed to fit this page, I was told by a teacher that she had arranged for me to meet professor Vincent Rijmen. Given that it was April 1, 2014, my first thought was that this was an elaborate prank – apparently not realizing that not every language teacher knows what the advanced encryption standard is. Two months later, I began my first internship at COSIC. At the end of my stay, Vincent Rijmen suggested that I look into correlation matrices. This profound but unsung idea, due to professor Joan Daemen, led to my master's thesis and ultimately to this PhD thesis. I am grateful to my promotor Vincent for his confidence in me and for all his advice, not least this initial suggestion almost a decade ago.

I would like to thank doctor Anne Canteaut, professors Joan Daemen, Gregor Leander, Hugo Van hamme and Fréderik Vercauteren for being part of my examination committee, as well as professor Paul Sas for chairing it. In September 2017, I had the opportunity to visit Anne Canteaut at INRIA-Paris. I hope that Chapter 4 solves the problem that we worked on to her satisfaction. Her work with Christina Boura on parity sets inspired Chapter 5. As already mentioned, Joan Daemen's correlation matrices started everything. His ideas about side-channel countermeasures influenced Chapter 11. Gregor Leander invited me to Bochum during the first week of my doctoral studies in September 2021. Unaware of his work on invariants, I had started to investigate the eigenvectors of correlation matrices. The connection with his work led to the first application (Chapter 6) of my theoretical ideas. Our joint work on backdoored ciphers appears in Chapter 12. Fréderik Vercauteren might recall some of our discussions when reading Section 10.4 on the Legendre PRF. I will not be surprised if he comes up with a subexponential attack some day.

I also want to acknowledge my coauthors that were not yet mentioned, in degree reverse lexicographical order: Yu Long Chen, Bart Mennink, Christoph Dobraunig, Siemen Dhooghe, Friedrich Wiemer, María Naya-Plasencia, Léo Perrin, Zhenda Zhang, Giuseppe Vitto, Aleksei Udovenko, Yosuke Todo, Danilo Šijačić, Aein Rezaei Shahmirzadi, Yu Sasaki, Adrián Ranea, Amir Moradi, Yunwen Liu, Chaoyun Li, Gaëtan Leurent, Patrick Felke, Maria Eichlseder, Itai Dinur, Begül Bilgin, Ward Beullens, Christof Beierle and Tomer Ashur.

A few more acknowledgments are in order. Michiel Verbauwhede completed a successful master's thesis under the supervision of Chaoyun Li and myself, and might have made the previous list if this thesis had been finished a few months later. Wouter Castryck has been incredibly generous with his time, answering my questions about exponential sums on more than one occasion. Taking his courses on algebraic number theory and combinatorics has been a pleasure. The Fonds Wetenschappelijk Onderzoek (FWO) supported my work financially through a PhD fellowship for fundamental research.

I wanted to write another paragraph to thank some people personally, but it seems more prudent to do this the right way:



Abstract

The first part of this thesis develops a general approach to symmetric-key cryptanalysis. It brings together linear, differential and integral cryptanalysis in a single framework, and extends these techniques in several ways. A universal notion of trails is introduced, leading to a systematic method to evaluate the properties of iterated functions. The theory provides a unified description of extensions of linear cryptanalysis, and clarifies the connections between them. For differential cryptanalysis, it leads to the definition of quasidifferential trails, which make it possible to estimate the probability of differential characteristics without relying on assumptions of probabilistic independence. In integral cryptanalysis, it suggests a spectrum of properties between zero-sums and saturation. These can be obtained and analyzed using a new theory of ultrametric trails, that generalizes division or monomial trails.

The second part of this thesis turns to applications of cryptanalysis. Using a characterization of invariants as eigenvectors of correlation matrices that follows from the first part, weak key attacks on reduced-round Midori-64 and MANTIS are given. The South-Korean and American format-preserving encryption standards FEA and FF3-1 are broken using multidimensional linear cryptanalysis. Differential attacks on Rectangle, KNOT and Speck are reevaluated using quasidifferential trails, showing that some of these attacks were invalid and that others work only for a subset of keys. A new generic attack on contracting Feistel ciphers leads to attacks on the Chinese commercial encryption standard SM4 with a reduced number of rounds. The security of several arithmetization-oriented primitives is analyzed, leading to attacks on some instances of GMiMC-erf, GMiMC-crf, HadesMiMC and the Legendre PRF. An attack on the backdoored cipher LowMC-M is given, and two new backdoored ciphers that follow more standard design principles are proposed. Finally, it is shown how linear cryptanalysis can be used to analyze the security of side-channel countermeasures.

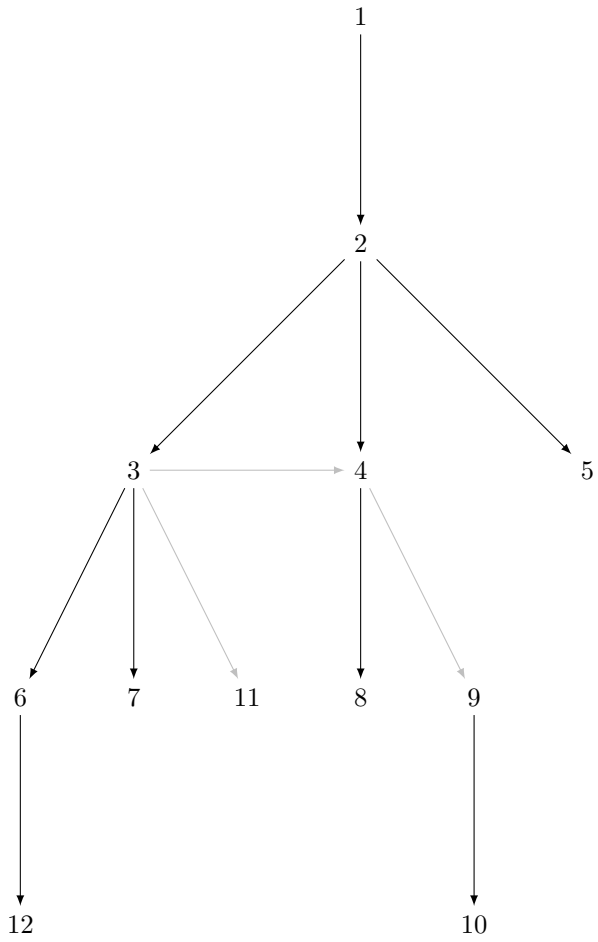
Beknopte samenvatting

Het eerste deel van deze thesis ontwikkelt een algemene theorie van symmetrische-sleutel cryptanalyse. Ze brengt lineaire, differentiële en integrale cryptanalyse samen in eenzelfde kader. Een universele definitie van paden maakt het mogelijk om de eigenschappen van geïtereerde functies op een systematische manier te bepalen. De theorie laat toe om de uitbreidingen van lineaire cryptanalyse op een uniforme manier te beschrijven. Voor differentiële cryptanalyse leidt ze tot de definitie van quasidifferentiële paden, die het mogelijk maken om de kans van differentiële karakteristieken te bepalen zonder gebruik te maken van probabilistische onafhankelijkheidsveronderstellingen. In integrale cryptanalyse geeft de theorie aanleiding tot een spectrum van eigenschappen tussen nulsommen en saturatie. Deze eigenschappen kunnen ontdekt en onderzocht worden met behulp van een nieuwe theorie van ultrametrische paden, een veralgemening van monomiale paden.

In het tweede deel van deze thesis worden cryptanalytische toepassingen besproken. Op basis van een karakterisatie van invarianten als eigenvectoren van correlatiematrices die uit het eerste deel volgt, worden zwakke sleutel aanvallen op Midori-64 en MANTIS met een verminderd aantal ronden gevonden. De Zuid-Koreaanse en Amerikaanse standaarden voor formaat-bewarende encryptie FEA en FF3-1 worden gebroken met behulp van meerdimensionale lineaire cryptanalyse. Differentiële aanvallen op Rectangle, KNOT en Speck worden herbekeken met behulp van quasidifferentiële paden, waaruit blijkt dat sommige fout zijn en andere enkel voor een deel van de sleutels werken. Een nieuwe generische aanval op samentrekkende Feistel constructies leidt tot aanvallen op de Chinese commerciële blokcijferstandaard SM4 met een verminderd aantal ronden. Een analyse van de veiligheid van verschillende aritmetisatiegeoriënteerde primitiven leidt tot aanvallen op sommige voorbeelden van GMiMC-erf, GMiMC-crf, HadesMiMC en de Legendre PRF. Er wordt een aanval gegeven op het blokcijfer LowMC-M, dat een achterdeurtje bevat. Bovendien worden twee nieuwe blokcijfers die meer gangbare ontwerpprinciples volgen, maar toch een achterdeurtje bevatten, ontwikkeld. Ten slotte wordt aangetoond hoe lineaire cryptanalyse kan gebruikt worden om de veiligheid van beschermingsmechanismen tegen nevenkanaalaanvallen te analyseren.

Contents

Interdependence of chapters



1

Introduction

Unlike the four puzzles or 'cryptograms' in the preface, real cryptography protects important information that is not meant to be recovered so easily. In the 1920s, William Friedman defined cryptanalytics as "the science which embraces all the principles, methods and means employed in the *analysis* of cryptograms, that is, their reduction or solution without a knowledge of the system or the key, or the possession of the code book, by a detailed study of the cryptograms themselves" [144], and *cryptanalysis* as the application of cryptanalytics to cryptograms.

Nevertheless, no puzzle accurately reflects the nature of modern cryptanalysis. Contemporary research in cryptanalysis is concerned with the general analysis of cryptosystems, rather than the solution of specific cryptograms. The end goal of this research is either a general method to break the system, or additional insight that can be used to improve its design.

This chapter sketches the context of this thesis. Symmetric-key cryptography is introduced in Section 1.1. The term 'symmetric' refers to the fact that the same key is used for encryption and decryption. Section 1.2 clarifies what is meant by a secure cryptosystem. Constructions of symmetric-key cryptography are discussed in Section 1.3, and the main techniques for their analysis are introduced in Section 1.4. Section 1.5 outlines the goals of this thesis.

1.1 Symmetric-key cryptography

This thesis is concerned with the cryptanalysis of symmetric-key cryptography, and more specifically with the analysis of *primitives*. These are the elementary components from which all practical symmetric-key constructions are built.

There are three types of primitives that underpin the majority of applications, and that play an important role in this thesis: (i) cryptographic permutations, (ii) block ciphers, and (iii) tweakable block ciphers. The principles behind their construction are largely the same, and they can be constructed from each other in various ways. Permutations are the most bare-bones primitives: they provide an invertible, unstructured mapping between inputs and outputs. The

meaning of 'unstructured' depends on the context and is discussed extensively in Section 1.2. Block ciphers are permutations parameterized by a key: they provide an invertible, secret mapping between inputs and outputs. Finally, tweakable block ciphers include an additional non-secret *tweak* input. Every choice of the tweak yields a new block cipher.

The primitives (i) to (iii) do not exist in isolation. This section provides an overview of the broader context in which they are used. In particular, Sections 1.1.1 to 1.1.3 describe how primitives are used to achieve the main goals of cryptography.

1.1.1 Confidentiality

Confidentiality refers to the ability to keep secrets. Encryption schemes provide this functionality: given a secret key, they convert arbitrary-length messages or *plaintext* into *ciphertext*. Decrypting the ciphertext to recover the original message should be infeasible unless the secret key is known.

In the past, encryption schemes were often constructed using a *code book*: a table with sequences of symbols in one column and their code equivalents in another. Generating code books is harder than it might appear to be, and storing large tables is not really practical either. Block ciphers address both problems by replacing the traditional code book with a mathematical function E_k parameterized by a secret key k.

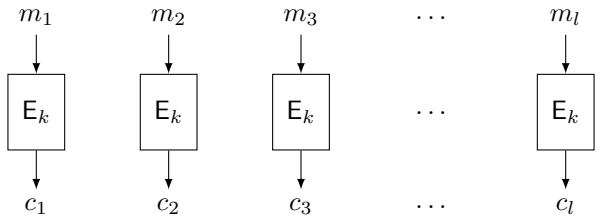


Figure 1.1: Electronic code book encryption.

Building an encryption scheme using a code book or block cipher might seem obvious: split the message into blocks $m_1, \ldots, m_l$ and compute the corresponding ciphertext blocks as shown in Figure 1.1. Unfortunately, this approach is not secure because duplicate message blocks result in duplicate ciphertext blocks. Nevertheless, it was widely used before the widespread use of machine- and computer-based cryptography – although policies such as frequent replacement of keys ('supersession') and compartmentation were used to control the damage [69]. To achieve modern standards of security, each message block in Figure 1.1 should

really be encrypted using a different block cipher. This goal can be achieved more efficiently by using a tweakable block cipher E_k^t. For example, if the tweak t takes integer values, then $\mathsf{E}_k^1, \ldots, \mathsf{E}_k^l$ are l different block ciphers.

There are other ways to construct encryption schemes. One particularly common example is *counter mode* [117]. The basic principle is shown in Figure 1.2: rather than encrypting the message blocks, a public value known as the *nonce* is encrypted to generate a keystream. The ciphertext is obtained by adding the message to this keystream. The addition is usually in a vector space of the form $\mathbb{F}_2^n$, *i.e.* using exclusive or. Every nonce can be used at most once, for similar reasons as above. The variant of counter mode in Figure 1.2 was first proposed by Peyrin and Seurin in 2016 [233]. Conventional counter mode includes the counter in the input of the block cipher. AES-GCM [130] is a widely-used example and is part of TLS[1]. This avoids the use of a tweakable block cipher, but achieves lower security because distinct messages *always* result in distinct ciphertexts. Yet another approach uses the key, nonce and counter as the input to a cryptographic permutation with feedforward. This is used in Chacha20-Poly1305 [193], another mode supported by TLS.

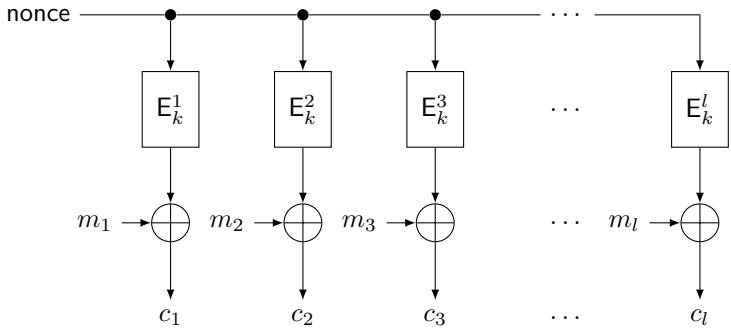


Figure 1.2: Counter mode encryption using a tweakable block cipher.

Instead of using a permutation or (tweakable) block cipher in counter mode, it is also possible to build a dedicated primitive to generate a keystream. This is called a *stream cipher*. Rather than generating blocks, these primitives typically generate the output bit by bit. They do not play a role in this thesis.

[1] *Transport layer security* is used for secure web browsing, as well as other applications.

1.1.2 Integrity

Encrypting a message does not guarantee that the integrity of the plaintext is protected. For example, if the adversary flips a bit in the ciphertext of a message encrypted using counter mode, then the corresponding bit in the decrypted message will also be flipped. This can have disastrous consequences.

To address the integrity problem, *message authenticated codes* (MACs) can be used. Given a secret key, a MAC function outputs a *tag*. Creating a tag without the key ('forgery') should be infeasible. Like encryption schemes, MACs can be constructed from permutations, block ciphers or tweakable block ciphers.

A wide variety of MAC functions have been proposed. One example is shown in Figure 1.3. It is a variant of the LightMAC construction [211]. The output of the last block cipher call can be truncated to reduce the length of the tag, provided that it is long enough to avoid forgery by trial and error.

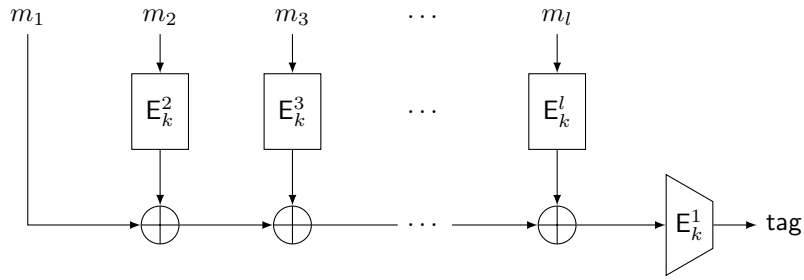


Figure 1.3: Variant of LightMAC using a tweakable block cipher.

Confidentiality and integrity are often combined into a single security notion: *authenticated encryption*. This functionality can be achieved by applying a MAC function to the plaintext ('mac-then-encrypt') or by generating the tag from the ciphertext ('encrypt-then-mac'), but there are also dedicated constructions. In August 2018, the United States *national institute of standards and technology* (NIST) issued a call for submissions to a standardization project focused on lightweight authenticated encryption.

Together with Yu Long Chen, Christoph Dobraunig and Bart Mennink, the author of this thesis submitted Elephant [49]. It is an encrypt-then-mac construction that combines the encryption scheme from Figure 1.2 and the MAC from Figure 1.3. It uses a tweakable block cipher constructed from a cryptographic permutation. The design rationale of Elephant is discussed in the joint papers [47, 50]. In March 2021, Elephant was selected as one of the finalists. NIST announced Ascon [125] as the winner in February 2023.

1.1.3 Other functionalities

Encryption schemes and message authentication codes are not the only functionalities provided by symmetric-key cryptography. Listing all applications is beyond the scope of this section, but two examples are worth mentioning.

As discussed in Section 1.1.1, counter mode can be used to generate an unpredictable stream of bits. This is in itself an important application of symmetric-key cryptography, and there are several other constructions to achieve the same functionality. Unpredictable bits are necessary to generate keys and are an important resource in many protocols.

Hash functions are particularly important and several examples of them will be encountered in Part II of this thesis. A hash function is a public function that maps arbitrary length messages to fixed-length *digests*. Depending on the application, one or more of the following security properties are desired: (i) it is infeasible to find a message that hashes to a given digest, (ii) given a message it is infeasible to find another message with the same digest, and (iii) it is infeasible to find two different messages with the same digest. These requirements are called *preimage resistance*, *second preimage resistance* and *collision resistance* respectively. Depending on the application, additional properties may be required. Hash functions are widely used; for example, most digital signature schemes hash the message before producing a signature.

The hash functions discussed in this thesis are based on the sponge construction. The most important example is the NIST standard SHA-3. As shown in Figure 1.4, the message blocks are added to the top r input bits of an n-bit cryptographic permutation P. This is called the absorption phase. After all message blocks have been processed, the digest is extracted by truncating the output of the permutation. This is called the squeezing phase. In fact, it is possible to extend this process to extract arbitrary-length digests. Constructions that provide this functionality are called *extendable-output functions*.

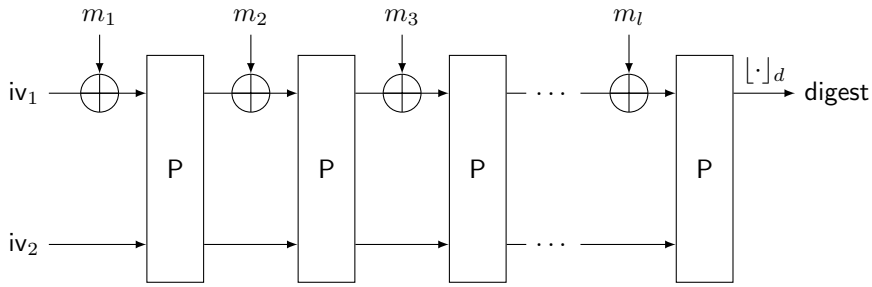


Figure 1.4: Sponge construction based on a cryptographic permutation P.

The parameter r is called the *rate*, and $c = n - r$ is the *capacity*. The domain of the permutation P typically needs to be quite big, because the capacity must be large enough to avoid internal collisions.

1.2 Defining security

The security of all the examples in Section 1.1 relies on the assumption that solving some problem is infeasible. The feasibility of solving a problem clearly depends on the capabilities of the adversary. However, formalizing these capabilities turns out to be difficult.

Formalizing security is important because it makes it possible to formulate precise security claims, which can then be proved or disproved. Unfortunately, as pointed out below, all existing definitions have significant limitations. Sections 1.2.1 to 1.2.3 briefly introduce the main approaches to defining security. The discussion follows an anti-chronological order.

Remark 1.1. Security definitions generally start from the assumption that the adversary lacks some knowledge, such as what key was used for encryption. Probability theory provides a convenient mathematical model for this situation.

A finite probability space consists of a finite set $\mathfrak{S}$, the sample space, and a probability function that maps the subsets of $\mathfrak{S}$ to $[0, 1]$. The probability of a subset A of $\mathfrak{S}$ in denoted by $\Pr[A]$, and $\Pr[\mathfrak{S}] = 1$. Furthermore, if A and B are disjoint subsets of $\mathfrak{S}$, then $\Pr[A \cup B] = \Pr[A] + \Pr[B]$. Throughout this thesis, the probability space will often be implicit.

A random variable $\boldsymbol{x}$ is a function $\boldsymbol{x} : \mathfrak{S} \to X$ to a set X. For x in X, the probability $\Pr[\{s \in \mathfrak{S} \mid \boldsymbol{x}(s) = x\}]$ is denoted by $\Pr[\boldsymbol{x} = x]$. For familiar concepts such as the average $\mathsf{E}\,\boldsymbol{x}$ and the variance $\mathsf{Var}\,\boldsymbol{x}$ of a random variable, the reader is referred to standard references such as the first volume of Feller [142]. A uniform random permutation is a random variable taking values in the set of permutations (on a finite set), so that all permutations are equally probable. ▷

1.2.1 Reductionist security

Under the influence of computational complexity theory, reductionist security notions were introduced in cryptography in the 1980s beginning with Goldwasser and Micali [145]. Bellare *et al.* [30] proposed the first non-asymptotic definitions. These definitions also keep track of more detailed information about the adversary, such as the number of queries it can make. Indeed, the concrete

security of an encryption scheme typically depends on the number of messages (chosen by the adversary itself) for which the ciphertext is known.

A reductionist security proof shows that every attack on a cryptographic system solves a problem that is conjectured to be hard. Confidence in the difficulty of the problem then leads to confidence in the security of the system. There are, however, a few practical issues such as the potential overhead incurred by reductions and the fact that no problem has ever been proven to be hard. Furthermore, this approach to security is not useful in symmetric-key cryptography because designing primitives is about constructing rather than repurposing hard problems.

A more fundamental problem is the fact that reductionist security definitions do not lead to meaningful assumptions on symmetric-key primitives. To illustrate this, consider the *standard model* for constructions based on block ciphers [30, 209]. This model reduces the security of an encryption scheme or message authentication code to the *pseudorandomness* of a block cipher. A block cipher is a (q, t, ε)-secure *pseudorandom permutation* if no algorithm with runtime t and making q queries can tell the difference between the cipher with a uniform random key and a uniform random permutation with advantage greater than ε. The advantage is the absolute value of the difference between the probability that the algorithm correctly recognizes the block cipher and the probability that it falsely recognizes the uniform random permutation.

For an n-bit key, exhaustive search shows that $\varepsilon \geq t/2^n$ if t is measured in block cipher evaluations. One might conjecture that this is essentially optimal for a secure block cipher. However, the following example shows that $\varepsilon \geq \sqrt{t/2^n}$. This does not imply that all block ciphers are broken, but rather highlights the issues with computational security definitions.

Example 1.1. Koblitz and Menezes [186] and Bernstein and Lange [34] point out that a block cipher E_k with an n-bit key is at most a $(1, 1, 2^{-n/2})$-secure pseudorandom permutation. Let f be a Boolean function that is easy to compute and balanced, *i.e.* taking the values zero and one an equal number of times. If $f(c)$ is zero, then conclude that the ciphertext c was encrypted using the block cipher. Otherwise, conclude that it is the output of a permutation sampled uniformly at random from all permutations.

This attack works because of the typical properties of the function $k \mapsto f(\mathsf{E}_k(p))$ for a fixed plaintext p. If $\frac{1}{2} + \gamma$ is the fraction of keys k that are mapped to zero, then the value of γ depends on f but $\gamma \approx \pm 2^{-n/2}$ is common. Hence, the advantage is $\left|(\frac{1}{2} \pm \gamma) - \frac{1}{2}\right| \approx 2^{-n/2}$. Although prp-security does not rule out this attack, it is not meaningful in practice because the sign of γ is unknown unless excessive precomputation is performed.

Using a cryptographic hash function, it is easy to construct several functions $f_1, \ldots, f_t$ with the same properties as f from above. Furthermore, with enough precomputation it can be ensured that every function f_i results in a positive bias $\gamma_i \geq \sqrt{2\pi}\, 2^{-n/2}$. For a uniform random key, the majority of the outputs of $f_1, \ldots, f_t$ are zero with probability approximately $\Phi(\sqrt{2\pi t\, 2^n})$, where Φ is the cumulative standard normal distribution. Since $\Phi(\sqrt{2\pi t\, 2^n}) - \frac{1}{2} \approx \sqrt{t/2^n}$, the parameter ε satisfies $\varepsilon \geq \sqrt{t/2^n}$. ▷

A similar problem occurs for collision-resistant hash functions: since a hash function is public, there is clearly an efficient algorithm that outputs a collision. Like in Example 1.1, the problem is that finding this algorithm is infeasible.

In practice, prp-security is used as a (poor) justification for replacing block ciphers with uniform random permutations during the security-analysis of a construction. This turns the analysis into a purely probabilistic problem and is known as information-theoretical security.

1.2.2 Information-theoretical security

Information theory and its application to cryptography were introduced in two influential papers of Shannon [250, 251]. He formalized the intuitive idea that ciphertext should not reveal anything about plaintext and called this security notion *perfect secrecy* [251, §10]. Vernam had already proposed such an encryption scheme thirty years earlier [172, §13]. His telegraph-based system uses the exclusive-or of the message and a keystream as the ciphertext. Mauborgne realized that this system is perfectly secure if the keystream is uniform random.

The idea of perfect secrecy is a simplification of reality, because it hinges on the availability of uniform random bits. A lack of careful analysis of the methods used to generate these bits can easily turn the lofty ideal of perfect secrecy into a security disaster. From a cryptanalytic viewpoint, the analysis of most 'true random number generators' is at best flimsy, and cryptographic post-processing is required in any case. The main difficulty is not so much to generate some data that the adversary cannot predict, but rather making it difficult to tell the difference between this data and uniform random bits.

High-level constructions such as encryption schemes are usually analyzed in the so-called *ideal model* [209]. This model replaces block ciphers by uniform random permutations and tweakable block ciphers by families of independent and uniform random permutations. Cryptographic permutations are also modelled as uniform random permutations, but the adversary has the ability to query them. A stronger model of block ciphers, known as the *ideal cipher model*, gives

the adversary query access to the cipher (with the key as an input) and assumes that each key results in an independent and uniform random permutation.

The ideal model is information-theoretical, but the most common security notion is *indistinguishability* rather than perfect secrecy. For example, it should not be possible to reliably distinguish an encryption scheme from an idealization that outputs uniform random ciphertext. Like in the definition of prp-security from Section 1.2.1, the insecurity of the system is quantified by the maximum possible advantage over all possible adversaries. The advantage is a function of the number of queries made by the adversary, but the ideal model does not restrict computational resources. In other words, an adversary is a hypothesis test and its advantage is equal to the absolute value of the difference between its success-probability and its false-positive rate.

Although the information-theoretical approach is useful to show the absence of certain generic attacks on high-level constructions, its applications to the analysis of primitives are limited. Two exceptions are worth mentioning. The first exception is the use of information-theoretical proofs as sanity checks for the overall internal structure of a primitive. Two examples of this will be mentioned in Section 1.3. The second exception is Vaudenay's *decorrelation theory* [275], which is an information-theoretical approach to block cipher design.

1.2.3 Cryptanalytic security

Given that that neither the reductionist nor the information-theoretical approach provide adequate security definitions, one is left with the informal adage that a cryptosystem is secure if no attack on it is known to the adversary. Hence, confidence in the security of primitives comes from their analysis.

Nevertheless, it is important to be precise about the resources required by an attack. Likewise, designers must make specific security claims even if these cannot be formalized. These requirements help to avoid needless debate about whether or not the security of a primitive has been violated. The most important aspects of an attack on a primitive are its goal, its access model, and its cost.

For (tweakable) block ciphers, the goal of the adversary is often either to recover the key or to distinguish the ciphertext from the output of a uniform random permutation. However, neither of these goals is meaningful for cryptographic permutations. An attack on a cryptographic permutation should exhibit some unexpected structure that can be exploited to attack one of the constructions in which the permutation might be used.

The input and output access models listed in the first column of Table 1.1 are

Table 1.1: Common access models for primitives.

Input and output	Key	Tweak
ciphertext only	single key	known tweak
known plaintext	weak key	weak tweak
chosen plaintext	related key [57]	related tweak
chosen ciphertext	known key [182]	chosen tweak

self-explanatory. It is common practice to assume that a chosen ciphertext attack can also use chosen plaintexts. An adversary is called non-adaptive if the plaintexts and ciphertexts it chooses are predetermined. For block ciphers, the role of the key must also be specified. This leads to the models in the second column of Table 1.1. Somewhat confusingly, attacks in the single key model are supposed to work for all or most keys – as opposed to weak key attacks. In a related key attack, the adversary can query the cipher under two different keys that are somehow related. Not all relations can be allowed [242]. The related key and known key models are mainly meaningful in the context of block cipher based hash functions. In addition to the access models listed in Table 1.1, it is possible that the implementation of the primitive unintentionally reveals information. This is called side-channel leakage.

The cost of an attack is determined by several parameters, the most important of which are (i) its success-probability and false-positive rate, (ii) its data complexity, (iii) its time complexity, and (iv) its memory requirements. For a key-recovery attack, the false-positive rate is equal to the fraction of remaining candidate keys. Data complexity (ii) refers to the number of known or chosen plaintexts and ciphertexts required by an attack. The time complexity is traditionally measured in terms of equivalent evaluations of the primitive.

1.3 Construction of cryptographic primitives

Sections 1.3.1 to 1.3.3 introduce common approaches to the construction of symmetric-key primitives. Generally speaking, the same principles are used for permutations, block ciphers and tweakable block ciphers. Design decisions are influenced by two factors: the ability to argue that *known* cryptanalytic attacks will not be successful, and the efficiency of the primitive (implementation size, latency, throughput, ...) on the target platform.

All of the constructions described below are *iterated*. That is, the final primitive F is obtained as a composition $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$ of simple functions or *rounds*.

The idea of composing several weak systems to obtain a stronger one dates back to premodern cryptography, when it was known as *superencipherment* [172]. Shannon explicitly proposed to build 'mixing transformations' as compositions of non-commuting transformations [251]. It is common practice to use more rounds than suggested by the preliminary security analysis. As in other engineering disciplines such as structural and mechanical engineering, this is meant to provide a safety margin against oversights in the analysis.

1.3.1 Feistel networks

Figure 1.5 shows two rounds of a Feistel network. This structure is always invertible, even if the functions F_1 and F_2 are not. Feistel networks are named after Horst Feistel, who was part of the IBM team that designed the first commercial encryption standard DES for the United States. They have the interesting feature that encryption and decryption are identical up to reversing the order of the round functions. This saves area in hardware implementations, since the same circuit can be used for encryption and decryption.

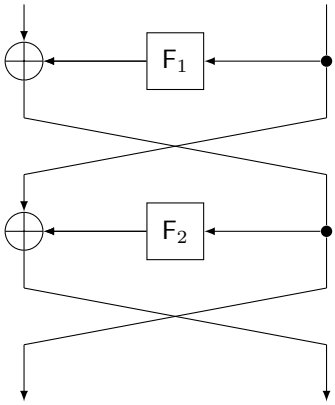


Figure 1.5: Two rounds of a Feistel network.

The Feistel structure has been generalized in various ways, for example by extending the number of branches to more than two. In addition to DES, there are numerous other primitives based on (generalized) Feistel networks. For example, this thesis includes cryptanalytic results on FEA [198], FF3-1 [129], Speck [24], SM4 [118] and GMiMC [6].

The information-theoretical analysis of Feistel constructions was initiated by Luby and Rackoff [208]. Although these results indicate that the Feistel structure is generally sound, they assume that $\mathsf{F}_1, \mathsf{F}_2, \ldots$ are uniform random functions.

As discussed in Section 1.2.2, such results say little about the concrete security of a primitive. In practice, most attacks use the fact that the functions $\mathsf{F}_1, \mathsf{F}_2, \ldots$ have exploitable structure. One way to instantiate these functions is using a combination of modular addition, bitwise rotation and exclusive-or operations ('ARX'). Another approach is to use a substitution-permutation network.

1.3.2 Substitution-permutation networks

A round of a substitution-permutation network consists of an *S-box layer* and a reordering of the bits of the state. An S-box is a permutation that is applied to a small part of the state. Substitution-permutation networks can be used as a part of a construction such as a Feistel cipher, or as a standalone primitive.

An archetypical example is shown in Figure 1.6. The figure shows one round of the ISO-standardized block cipher PRESENT [71]. In addition to the S-box and bit-permutation layers, the round function of PRESENT includes a round key addition. The round keys are derived from the key using a *key-scheduling algorithm*.

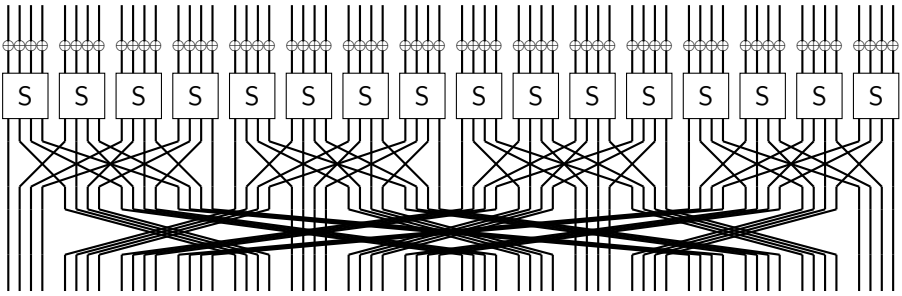


Figure 1.6: One round of PRESENT.

It has become conventional to expand the meaning of 'substitution-permutation network' to more general structures such as the one shown in Figure 1.7. That is, the bit-permutation is replaced by a more general linear layer L. The term 'linear' refers to the fact that L is a linear map on the state space, usually $\mathbb{F}_2^n$.

A carefully chosen linear layer can improve the security of the primitive against certain attacks. The designers of SHARK [243] first proposed to choose a linear function $\mathsf{L} : \mathbb{F}_{2^m}^n \to \mathbb{F}_{2^m}^n$ that maximizes

$$b_{\mathsf{L}} = \min_{\substack{x \in \mathbb{F}_{2^m}^n \\ x \neq 0}} \mathsf{wt}(x) + \mathsf{wt}(\mathsf{L}(x)) \,,$$

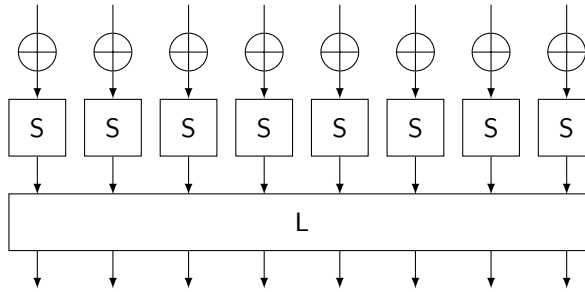


Figure 1.7: One round of SHARK.

with $\mathsf{wt}(x) = |\{1 \leq i \leq n \mid x_i \neq 0\}|$ the Hamming weight of x. The largest possible value of b_{L} is $n+1$ and is achieved if $x \mapsto (x, \mathsf{L}(x))$ generates a maximum distance separable (MDS) code. Intuitively, b_{L} is a measure of 'diffusion': if i coordinates of the input are changed, then at least $b_{\mathsf{L}} - i$ coordinates of the output must change. More importantly, b_{L} is the *differential branch number* and leads to a security argument against differential cryptanalysis (see Section 1.4.2).

The ideas described in the previous paragraph are part of the *wide trail strategy*, which was introduced by Daemen [99] and extended by Daemen and Rijmen [104, 107, 241]. The downside of SHARK's linear layer is that it is relatively expensive to implement. To address this issue, the block ciphers Square [102] and BKSQ [103] represent the state as a two-dimensional array and apply an MDS matrix only to either the rows or the columns. To ensure mixing in both dimensions, the linear layer additionally includes a cell-permutation. This work led to the design of Rijndael, the 128-bit variant of which was selected as the *advanced encryption standard* or AES by NIST. Its round function is depicted in Figure 1.8, along with the standard names for each of the four steps. The linear layer consists of ShiftRows (cell-permutation) and MixColumns (column-wise multiplication by an MDS matrix).

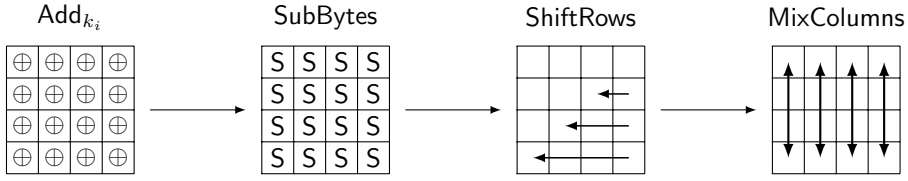


Figure 1.8: One round of the AES.

Aside from PRESENT, SHARK, Square and the AES, there are many

other substitution-permutation networks. For example, this thesis includes cryptanalytic results on Midori [18], MANTIS [29], Rectangle [292], KNOT [293], HadesMiMC [152] and LowMC-M [234]. All of these follow the wide-trail strategy in one form or another.

Unlike Feistel structures, the inverse of a substitution-permutation network can in general not be computed with the same circuit. For most applications, this is not an important downside. In fact, many encryption schemes (such as counter mode) never use the inverse of the primitive. Nevertheless, substitution-permutation networks with a similar self-inverse property have been proposed. A first approach is based on using involutive S-boxes and linear layers. This idea is used in Midori. Another approach is described in the next section.

1.3.3 Reflection ciphers

Reflection ciphers are (tweakable) block ciphers for which decryption is the same as encryption under a related key. The high-level structure is shown in Figure 1.9: decryption is the same as encryption up to adding a constant α to the round keys. This construction was introduced by the low-latency cipher PRINCE [76]. The tweakable block cipher MANTIS is based on the same idea.

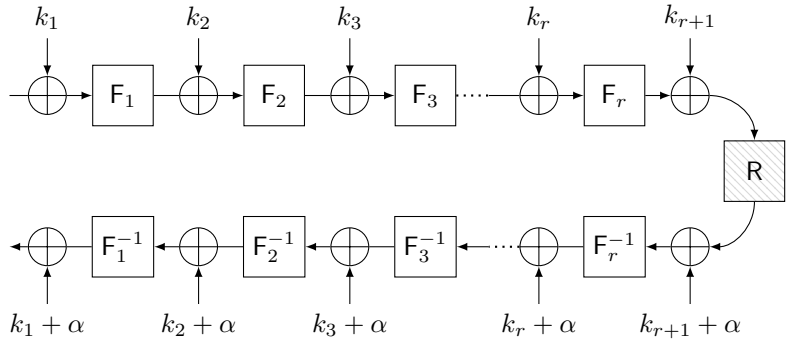


Figure 1.9: Reflection cipher.

In joint work with Yu Long Chen, the author of this thesis has analyzed the information-theoretical security of two-round reflection ciphers. This result was published at Crypto 2022 [45]. Similar to the results of Luby and Rackoff [208] about Feistel ciphers, it says little about the concrete security of reflection ciphers.

1.4 Analysis of cryptographic primitives

As discussed in Section 1.2.3, confidence in the security of primitives is ultimately derived from their cryptanalysis. An outline of the main principles of modern cryptanalysis is given Section 1.4.1. Sections 1.4.2 to 1.4.4 introduce three specific techniques in chronological order of discovery: differential, linear and integral cryptanalysis. They dominate contemporary research, both in applied and in theoretical work. Although there are several other important methods, none of them have led to a comparable wealth of applications and theory.

1.4.1 General principles

Cryptanalytical attacks usually exploit unexpected properties of primitives to break functionalities such as confidentiality, integrity, collision-resistance, etc. that a cryptographic system might claim to provide. A property is considered to be 'unexpected' if it does not hold for the information-theoretical idealization of the primitive (see Section 1.2.2). This is an informal description, since not all such properties are actually meaningful. A formal definition would have to avoid the theoretical issues that were discussed in Section 1.2.1. Furthermore, some properties are meaningful but do not break the functionality of the system.

The first step in almost every attack is the identification and analysis of a useful property. Due to the size of the search space, and the fact that cryptanalytic properties are often subtle, black-box methods do not suffice. Instead, techniques that exploit the structure of the primitive must be used. The most influential methods are described in Sections 1.4.2 to 1.4.4. All of them rely on the iterative structure of primitives, which enables the round-by-round analysis or equivalently *propagation* of properties.

Once a useful property has been identified, it can be used as a *distinguisher* between the primitive and its idealization. In many cases, it is possible to test the property by random sampling. This approach is sometimes called *statistical cryptanalysis*. In statistical terms, a distinguisher is a hypothesis test between two alternatives: were the samples obtained from the real primitive, or from its idealized variant? The hypothesis test is based on a *test statistic* that counts the number of samples for which the property is valid. The number of samples is related to the data complexity of the distinguisher, but not necessarily the same. In particular, in some cases such as for attacks based on pairs of inputs, the number of samples can exceed the amount of data.

The following result estimates the number of samples that are sufficient to obtain a constant distinguishing advantage. It assumes nothing about the

sampling method, other than that the standard deviation of the test statistic is proportional to $1/\sqrt{q}$ for q samples. Stronger results can be obtained if more assumptions about the distribution of the test statistic are made. Nevertheless, Theorem 1.1 is often close to optimal for intermediate advantages (such as 1/2).

Theorem 1.1 (Data-complexity)**.** *Let* $\boldsymbol{t}_{\mathsf{real}}$ *and* $\boldsymbol{t}_{\mathsf{ideal}}$ *be complex-valued random variables with averages* μ_{real} *and* μ_{ideal} *respectively. Let* q *be a positive integer such that* $\mathsf{Var}\,\boldsymbol{t}_{\mathsf{real}} \leq \sigma_{\mathsf{real}}^2/q$ *and* $\mathsf{Var}\,\boldsymbol{t}_{\mathsf{ideal}} \leq \sigma_{\mathsf{ideal}}^2/q$*. If*

$$q \geq \frac{2}{\varepsilon}\left(\frac{\sigma_{\mathsf{real}} + \sigma_{\mathsf{ideal}}}{|\mu_{\mathsf{real}} - \mu_{\mathsf{ideal}}|}\right)^2,$$

then $\boldsymbol{t}_{\mathsf{real}}$ *and* $\boldsymbol{t}_{\mathsf{ideal}}$ *can be distinguished with advantage* $1 - \varepsilon$*.*

Proof. Consider a hypothesis test with the half-plane below the dashed line in Figure 1.10 as its acceptance region. If $\boldsymbol{t}_{\mathsf{real}}$ is contained in the disk of radius τ_{real} around μ_{real}, then it is in the acceptance region. Likewise, if $\boldsymbol{t}_{\mathsf{ideal}}$ is contained in the disk of radius τ_{ideal} around μ_{ideal}, then it cannot be in the acceptance region. Since the advantage $1 - \varepsilon$ is equal to the difference between the success

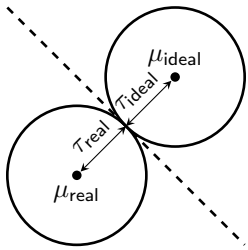


Figure 1.10: Separating $\boldsymbol{t}_{\mathsf{real}}$ and $\boldsymbol{t}_{\mathsf{ideal}}$ with a line in the complex plane.

probability and the false-positive rate, the parameter ε satisfies

$$\varepsilon \leq \Pr\left[|\boldsymbol{t}_{\mathsf{real}} - \mu_{\mathsf{real}}| \geq \tau_{\mathsf{real}}\right] + \Pr\left[|\boldsymbol{t}_{\mathsf{ideal}} - \mu_{\mathsf{ideal}}| \geq \tau_{\mathsf{ideal}}\right].$$

Applying Chebyshev's inequality to both terms yields

$$q \geq \frac{1}{\epsilon}\left(\frac{\sigma_{\mathsf{real}}^2}{\tau_{\mathsf{real}}^2} + \frac{\sigma_{\mathsf{ideal}}^2}{\tau_{\mathsf{ideal}}^2}\right).$$

The lower bound on q is minimized by setting $\tau_{\mathsf{real}} = (\sigma_{\mathsf{real}}/\sigma_{\mathsf{ideal}})\tau_{\mathsf{ideal}}$. Since $\tau_{\mathsf{real}} + \tau_{\mathsf{ideal}} = |\mu_{\mathsf{real}} - \mu_{\mathsf{ideal}}|$, this yields $\tau_{\mathsf{ideal}} = |\mu_{\mathsf{real}} - \mu_{\mathsf{ideal}}|/(1 + \sigma_{\mathsf{real}}/\sigma_{\mathsf{ideal}})$. Substituting this equality into the lower bound for q yields the result. □

Although Theorem 1.1 does not state it explicitly, its proof implies that the distinguisher can be implemented efficiently provided that $|\mu_{\mathsf{real}} - \mu_{\mathsf{ideal}}|$ and $\sigma_{\mathsf{real}}/\sigma_{\mathsf{ideal}}$ are known. If the test statistic is a consistent estimator, then the value of μ_{real} follows from the theoretical analysis. Estimates for μ_{ideal} and $\sigma_{\mathsf{real}}/\sigma_{\mathsf{ideal}}$ are usually easier to obtain.

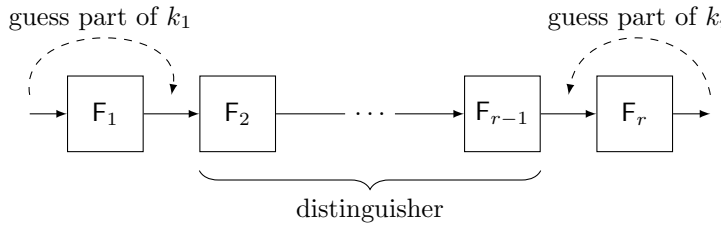


Figure 1.11: Key-recovery by prepending or appending rounds.

Distinguishers are often only the first step leading to an attack against the broader system. To limit the scope of this section, only key-recovery attacks are discussed here. There are two mechanisms by which key material can be extracted from a cryptanalytic property. The first approach assumes that the property is key-dependent. Little can be said about this in general, since the techniques are not generic. The other mechanism is illustrated in Figure 1.11.

As shown in Figure 1.11, one can guess key material from the first and/or last round and use a distinguisher for the middle rounds to check the validity of these guesses. In general, more than one round can be prepended or appended provided that the number of candidate keys is not too large. It is usually not necessary to guess all of the key material involved in these rounds, because most properties only depend on part of the state. To uniquely determine the correct guess, the distinguisher must have a low false-positive rate. The whole method relies on the assumption that a wrong guess breaks the middle-round property. For some properties, it is reasonable to assume that the behaviour for wrong guesses is comparable to that of the ideal primitive. This is called the *wrong-key-randomization hypothesis*.

To implement the key-recovery strategy from Figure 1.11, the test statistic has to be computed for all possible candidate keys. If there are n candidate keys and the distinguisher requires q samples, then the time complexity of a naive implementation of this idea is dominated by qn partial encryptions or decryptions. Chosen-plaintext attacks may require additional data if key material in the first round is guessed. The naive approach can be significantly improved when the test-statistic t is of the form $t = \sum_{i=1}^{q} f_k(z_i)$ for samples $z_1, \ldots, z_q$. For properties based on a rare event, it is often possible to discard

many samples without knowing the entire candidate key. Figure 1.12 illustrates another approach that will be referred to as *distillation*.

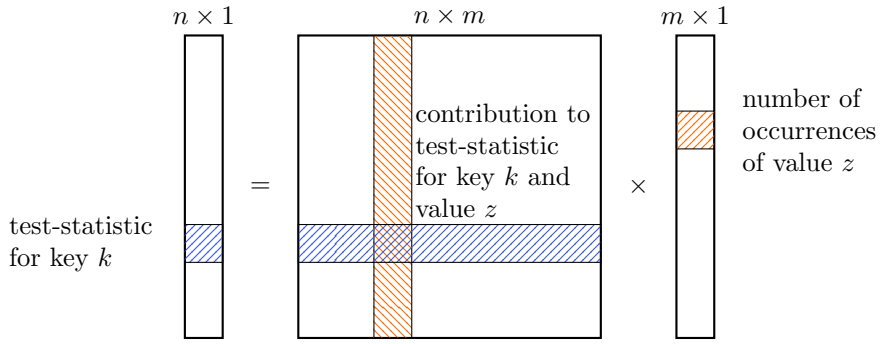


Figure 1.12: Key-recovery using distillation.

Distillation exploits the fact that the test statistic can often be computed from only part of the input and/or output of the middle rounds. Hence, it is worthwhile to separately count the number of samples with a given value for the relevant part. The results are stored in a vector of length m, as shown in the rightmost part of Figure 1.12. An $n \times m$ matrix containing the contribution $f_k(z)$ to the test statistic for every candidate key k and every value z is then constructed. The matrix-vector product yields the values of the test statistic for every key because

$$\sum_z \big|\{1 \le i \le q \mid z_i = z\}\big| \, f_k(z) = \sum_{i=1}^{q} f_k(z_i) = t \,.$$

The time complexity of this method is $\mathcal{O}(q + nm)$, which is often lower than $\mathcal{O}(qn)$. The memory-complexity is $\mathcal{O}(q + n + m)$ as opposed to $\mathcal{O}(q)$. Furthermore, as observed by Collard, Standaert and Quisquater [95], the matrix is circulant in many attacks. In this case, and with $n = m$, the time complexity reduces to $\mathcal{O}(q + n \log n)$ by using the fast Fourier transform (FFT) method to multiply with a circulant matrix.

1.4.2 Differential cryptanalysis

Differential cryptanalysis was proposed by Biham and Shamir at Crypto 1990 [58], and used to obtain attacks on reduced-round DES. They subsequently extended these results to the first attack on full-round DES [59]. Although their attack improves over exhaustive search, it requires a large number of chosen

plaintexts. This is in large part due to the fact that differential cryptanalysis was taken into account by the IBM team – assisted by the United States National Security Agency (NSA) – that designed DES. Indeed, although sources disagree about the relative involvement of IBM and NSA, the technique of differential cryptanalysis was known but classified in the 1970s [97, 170, 200].

Differential cryptanalysis is concerned with pairs of input and output differences, also known as *differentials*. The probability p of a differential (a, b) for a function $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is the number of solutions of the difference equation $\mathsf{F}(x + a) = \mathsf{F}(x) + b$ in x, divided by 2^n. Equivalently, with $\boldsymbol{x}$ uniform random on $\mathbb{F}_2^n$, the probability p satisfies

$$p = \Pr\left[\mathsf{F}(\boldsymbol{x} + a) = \mathsf{F}(\boldsymbol{x}) + b\right].$$

Differential cryptanalysis is a statistical attack in the sense that solutions (also called *right pairs*) can be found by sampling random values of x – although sometimes, such as in the analysis of hash functions, there are better options. If a differential has probability p, then $1/p$ samples will contain one right pair on average. This agrees with Theorem 1.1, which predicts a constant advantage for $q = \Omega(1/p)$. Indeed, if the test statistic is equal to the number of observed right pairs divided by q, then $\mu_{\mathsf{real}} = p$ and $\sigma^2_{\mathsf{real}} \approx p$ whereas $\mu_{\mathsf{ideal}} \approx 2^{-n}$ and $\sigma^2_{\mathsf{real}} \approx 2^{-n}$. More precise results can be obtained from the fact that the distribution of the test-statistic converges to a Poisson distribution as $q \to \infty$.

The key observation is that differentials can be propagated round-by-round. This provides a way to estimate the probability of a differential for a function F of the form $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$, where the functions F_i admit differentials with relatively high probability and are easier to analyze. In particular, the probability can be estimated based on *characteristics*. A characteristic is a sequence of intermediate input and output differences for each of the functions F_i. The probability of the characteristic $(a_1, \ldots, a_{r+1})$ is

$$\Pr\left[\bigwedge_{i=1}^{r} \mathsf{F}_i(\boldsymbol{x}_i + a_i) = \mathsf{F}_i(\boldsymbol{x}_i) + a_{i+1}\right],$$

with $\boldsymbol{x}_{i+1} = \mathsf{F}_i(\boldsymbol{x}_i)$ for $i = 1, \ldots, r$ and $\boldsymbol{x}_1$ uniform random on $\mathbb{F}_2^n$.

By the law of total probability, the probability of a differential is equal to the sum of the probabilities of all characteristics with matching input and output differences. Many analyses are based on the assumption that one or a few characteristics dominate the probability. However, estimating the probability of characteristics is still nontrivial: it amounts to counting the number of solutions to a system of several coupled difference equations. Lacking a better method, it has become common practice to multiply the one-round probabilities as if they correspond to independent events. Although this heuristic has been useful, it is clear that it cannot be correct in general.

Example 1.2 (Differential cryptanalysis of DES)**.** Figure 1.13 shows the overall structure of the characteristic used in Biham and Shamir's attack on DES [59]. The arrows are labeled by the differences on the values they carry. Only two rounds are shown because the input and output differences are the same; such a characteristic is called iterative. Careful analysis reveals two differences a that result in a zero-difference at the output of F with probability 35/8192. In fact, due to the structure of F, this result already uses the independence assumption mentioned above. Making further use of this assumption, the probability of the 13-round characteristic is approximately $(35/8192)^6 \approx 2^{-36}$.

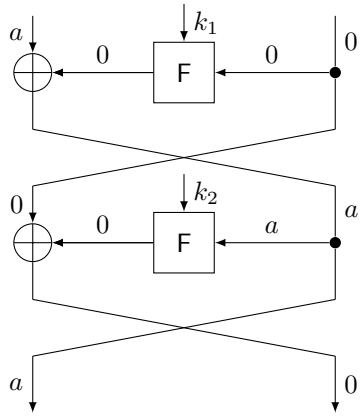


Figure 1.13: Iterative differential characteristic for DES.

The full-round attack prepends one round by choosing a structured set of inputs that is likely to contain a pair with the desired input difference. In addition, two rounds are appended. Each right pair suggests only a few *complete* candidate keys. Hence, candidates can be tested directly without storing a table. Biham and Shamir estimate the data complexity as 2^{47} chosen plaintexts [59]. ▷

The wide-trail strategy ensures that all differential characteristics have low probability, assuming the independence heuristic. For example, based on the differential branch number of MixColumns, it can be shown that any characteristic over four rounds of the AES must have a nonzero input difference in at least 25 S-boxes. Such S-boxes are called differentially active. Since each S-box has a maximum differential probability of 2^{-6}, this leads to a probability upper bound of 2^{-150}. In fact, it can be shown that the probability of every four-round differential is at most $(53 \cdot 2^{-34})^4 \approx 2^{-121}$, when averaged over independent and uniform random round keys [175].

1.4.3 Linear cryptanalysis

At Eurocrypt 1993, Matsui [215] introduced linear cryptanalysis as a new known-plaintext attack on DES. Linear cryptanalysis is based on probabilistic linear relations or *linear approximations*, a concept that was first used by Tardy-Corfdir and Gilbert in their attack on FEAL [261]. According to Coppersmith, linear cryptanalysis was not known to the designers of DES [97].

A linear approximation of a function $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ consists of a pair of bitvectors (u, v) in $\mathbb{F}_2^n \times \mathbb{F}_2^m$. The *masks* u and v determine linear Boolean functions $x \mapsto u^\mathsf{T} x = \sum_{i=1}^n u_i x_i$ and $x \mapsto v^\mathsf{T} x = \sum_{i=1}^m v_i x_i$. These functions define an equation $u^\mathsf{T} x = v^\mathsf{T} \mathsf{F}(x)$ in x. If F is a uniform random function, then the number of solutions is close to 2^{n-1} with high probability. The *bias* ε is equal to the number of solutions minus 2^{n-1}, and divided by 2^n. It turns out to be more convenient to work with the correlation $c = 2\varepsilon$ rather than the bias. That is, with $\boldsymbol{x}$ uniform random on $\mathbb{F}_2^n$, the correlation c satisfies

$$c = 2 \Pr\left[u^\mathsf{T} \boldsymbol{x} = v^\mathsf{T} \mathsf{F}(\boldsymbol{x})\right] - 1 \,.$$

Like differential cryptanalysis, linear cryptanalysis is a statistical attack: the correlation of an approximation can be estimated empirically by sampling random values of x. The test-statistic that estimates c by counting the number of solutions among q independent samples satisfies $\mu_{\mathsf{real}} = c$ and $\sigma^2_{\mathsf{real}} = (1 - c^2)/q$. For the ideal case, $\mu_{\mathsf{ideal}} \approx 0$ and $\sigma^2_{\mathsf{ideal}} \approx 1/q + 2^{-n}$. Hence, by Theorem 1.1, a constant advantage can be achieved when $q = \Omega(1/c^2)$. More accurate results can be obtained from the observation that the distribution of the test-statistic converges to a normal distribution as $q \to \infty$.

If $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$ with functions F_i that are relatively easy to analyze, then linear approximations can be analyzed round-by-round using *linear trails*. A linear trail is a sequence of compatible intermediate input and output masks for each of the functions F_i. The correlation of a linear trail is equal to the product of the correlations of the one-round approximations it encompasses. Tardy-Corfdir and Gilbert [261] and Matsui [215] combine the one-round approximations in a trail by adding up the corresponding equations. To compute the correlation of the combined approximations, Matsui introduced the following lemma.

Lemma 1.1 (Piling-up [215])**.** *For all independent random variables* $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_r$ *on* $\mathbb{F}_2$ *with* $c_i = 2 \Pr[\boldsymbol{z}_i = 0] - 1$*, it holds that* $2 \Pr[\sum_{i=1}^r \boldsymbol{z}_i = 0] - 1 = \prod_{i=1}^r c_i$*.*

Let $(u_1, \ldots, u_r)$ be a linear trail and let $\boldsymbol{x}_{i+1} = \mathsf{F}_i(\boldsymbol{x}_i)$ for $i = 1, \ldots, r$ with $\boldsymbol{x}_1$ uniform random on $\mathbb{F}_2^n$. Matsui applies Lemma 1.1 with $\boldsymbol{z}_i = u_i^\mathsf{T} \boldsymbol{x}_i + u_{i+1}^\mathsf{T} \mathsf{F}_i(\boldsymbol{x}_i)$ to estimate the correlation of the linear approximation (u_1, u_{r+1}). This is a heuristic argument, since Lemma 1.1 assumes that $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_r$ are independent –

which they are clearly not. Under this assumption, the correlation of the trail is an estimate for the correlation of the corresponding approximation.

It was shown by Daemen *et al.* [101] that the correlation of a linear approximation is equal to the sum of the correlations of all linear trails with matching input and output masks. This is conceptually similar to the relation between differentials and differential characteristics from Section 1.4.2 but, unlike for the probability of a characteristic, no heuristics are necessary to calculate the correlation of a trail. Many analyses, including Matsui's analysis of DES [214, 215], rely on the assumption that a single trail dominates the correlation.

Example 1.3 (Linear cryptanalysis of DES)**.** Figure 1.14 shows a linear trail for three rounds of DES [215, §5]. The arrows are labeled by the masks or the variables they carry. There exist masks u and v that define a linear approximation of F with correlation $\pm 5/8$. Hence, the correlation of the trail in Figure 1.14 is $\pm 25/64$. The sign depends on a linear combination of key bits.

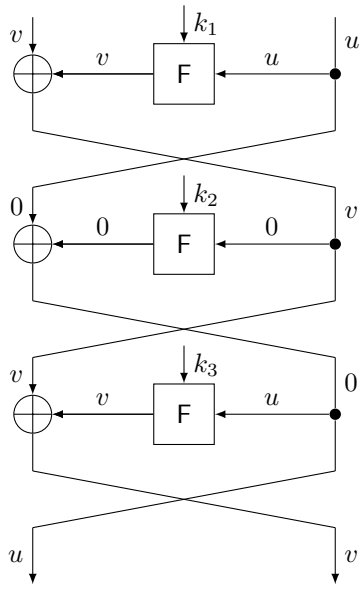


Figure 1.14: Linear trail for three rounds of DES.

Matsui [214] describes a 14-round trail with correlation approximately $\pm 2^{-20}$. The key-recovery strategy uses the distillation method described in Section 1.4.1. The data complexity is around 2^{43} for a success probability above 80%.

The linear attack has some advantages compared to the differential attack from Example 1.2. The data complexity is considerably lower, and the attack

only requires known plaintexts. A downside of the attack is that the success probability plummets when not enough data is available. If low success probabilities are acceptable, such as when the number of targets is large, then a differential attack is preferable because the probability of finding a right pair is proportional to the number of samples (when it is small). ▷

The wide-trail strategy leads to a security argument against linear cryptanalysis. For AES-like ciphers, the correlations of trails can be upper bounded in terms of the *linear branch number* of the linear layer. The linear branch number is defined similarly as the differential one, but with the transpose of the linear layer. Any linear trail over four rounds of the AES must have a nonzero output mask on at least 25 S-boxes. Such S-boxes are called linearly active. Since the maximum absolute correlation of any linear approximation over the S-box is 2^{-3}, one obtains an upper bound of 2^{-75} on the absolute correlation of any linear trail. In fact, it can be shown that the squared correlation of any linear approximation over four rounds of the AES is at most $(109\,953\,193 \cdot 2^{-54})^4 \approx 2^{-109}$ when averaged over independent and uniform random round keys [84, 175].

1.4.4 Integral cryptanalysis

Integral cryptanalysis originated in a dedicated attack on the block cipher Square [102], leading to the early name *Square attacks*. At FSE 2001, Knudsen and Wagner [184] systematized and extended these attacks and coined the term *integral attacks*. Nevertheless, until 2015, describing integral cryptanalysis alongside differential and linear attacks as one of the main families of techniques would have been far-fetched. This changed with the introduction of the *division property* by Todo [263, 264] and the follow-up work that ensued.

The attack on Square works by propagating a set of plaintexts with some constant cells and some *saturated* cells. A cell is saturated if all its possible values are realized, and every value occurs an equal number of times. One concludes that all ciphertexts sum to zero. A similar approach works for the AES, and in fact yields one of the most interesting reduced-round attacks.

Example 1.4 (Integral cryptanalysis of the AES.)**.** The notation of Knudsen and Wagner [102] will be used. Constant cells are labeled by 'C', saturated cells by 'A' and cells that sum to zero by 'S'. To indicate that some combination of cells is saturated, subscripts will be used. In particular, all saturated cells labeled by the same subscripts are jointly saturated.

Figure 1.15 illustrates the propagation of a set of 2^{32} plaintexts with constant off-diagonal elements and a saturated diagonal through four rounds of the AES. The functions $\mathsf{F}_1, \ldots, \mathsf{F}_4$ are the first four rounds, as depicted in Figure 1.8.

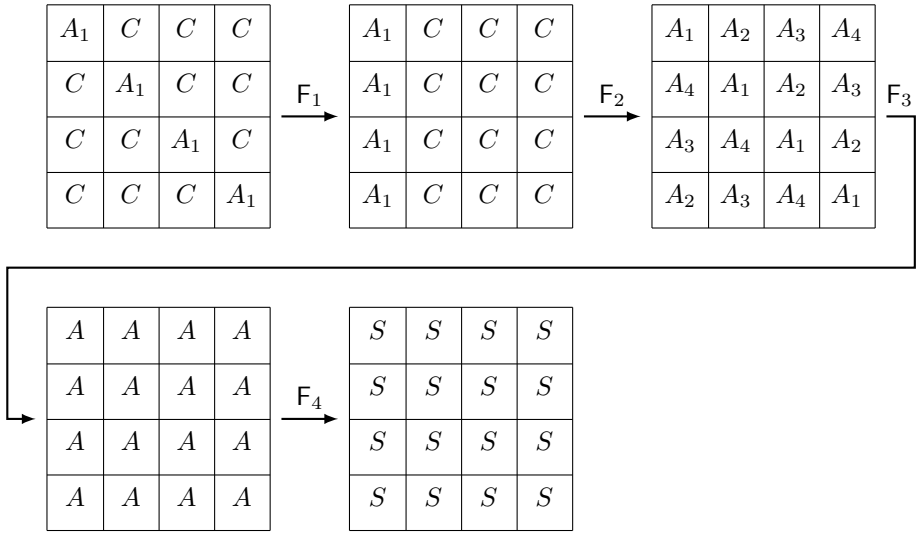


Figure 1.15: Integral property for four rounds of the AES.

After the last MixColumns step, none of the cells retains the saturation property. However, all bits of the state have the zero-sum property because every saturated set sums to zero and this property is preserved by linear maps.

The property from Figure 1.15 leads to a six-round key-recovery attack. The most efficient approach is based on the distillation framework described in Section 1.4.1, using the FFT-method [265]. ▷

Several years before the attack on Square, Knudsen [181] proposed another way to find zero-sum properties. Every Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be represented as a unique multivariate polynomial in the ring $\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 - x_1, \ldots, x_n^2 - x_n)$. This polynomial is called the *algebraic normal form* of f. If its degree is d or lower, then $\sum_{x \in V} f(x) = 0$ for all vector spaces V of dimension $d + 1$ and higher. Such sums were called higher-order derivatives by Lai [190], which motivated Knudsen to call this method higher-order differential cryptanalysis. Although integrals and higher-order differentials can both be used to deduce zero-sum properties, the underlying methods are different. Higher-order differential attacks traditionally focus on degree bounds, whereas integral attacks focus on the structure of the input set.

At Eurocrypt 2015, Todo [264] introduced a refinement of the zero-sum property. Specifically, a multiset $S \subseteq \mathbb{F}_2^n$ has the *conventional division property* of order k if all polynomials of degree strictly less than k sum to zero on S. If S is a nonempty

set, then the division property of order n corresponds to the saturated property. From a theoretical perspective, the division property partially reconciles the algebraic viewpoint of higher-order differentials with the structural viewpoint of integral attacks. However, it does not provide a complete unification because the saturated property is not a special case of the division property except for sets. From a practical perspective, the division property led to significant improvements to many integral attacks. For example, Todo [263] obtained the first full-round attack on the block cipher MISTY-1.

The structure of many block ciphers does not allow for a cell- or word-based analysis. This led to the desire to use the division property at the bit level. The original proposal of Todo and Morii [268], called the bit-based division property, was imperfect in the sense that it cannot explain all zero-sum properties. This gap was closed by subsequent theoretical work [79, 158, 166] and several nearly equivalent 'perfect' theories now exist; an overview is given by Hebborn, Leander and Udovenko [161]. From a contemporary point of view, a division property of a function $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is characterized by a pair (u, v) in $\mathbb{F}_2^n \times \mathbb{F}_2^m$. The *exponents* u and v correspond to monomial functions $x \mapsto x^u = \prod_{i=1}^n x_i^{u_i}$ and $x \mapsto x^v = \prod_{i=1}^m x^{v_i}$. For two such monomials, one can determine the coefficient of x^u in the algebraic normal form of the Boolean function $x \mapsto \mathsf{F}^v(x)$. To show that $\sum_{x \in S} \mathsf{F}^v(x) = 0$, it suffices to show that x^u does not occur in $\mathsf{F}^v(x)$ for all u such that $\sum_{x \in S} x^u = 1$. This connection was first made explicit by the parity set description of Boura and Canteaut [79].

If $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$, then the coefficient of x^u in $\mathsf{F}^v(x)$ can be determined using trails. A trail is a sequence $(u_1, \ldots, u_{r+1})$ of compatible intermediate input and output exponents such that $\mathsf{F}^{u_{i+1}}(x)$ contains the monomial x^{u_i} for $i = 1, \ldots, r$. Hao *et al.* [158] use the term *division trails*, whereas Hu *et al.* [166] use the term *monomial trails*[2]. In particular, x^{u_1} occurs in $\mathsf{F}^{u_{r+1}}(x)$ if and only if the number of trails between u_1 and u_{r+1} is odd. Zero-sum properties are often obtained by showing the absence of trails. This is easier than counting trails, but potentially misses some properties. The conventional division property is less precise, as it only keeps track of the Hamming weights of parts of the exponents.

[2]There is a subtle difference between both concepts that will be ignored until Chapter 5.

1.5 Goals

Successful cryptanalysis often builds on previous work, and over time this has led to the development of cryptanalytic theory. This theory is the subject of the first part of this thesis; its applications are examined in the second part.

1.5.1 Theory

It appears to be the case that new cryptanalytic techniques are only developed when the right target presents itself. Differential and linear cryptanalysis were developed in the wake of the publication of DES and FEAL. Integral cryptanalysis began as a dedicated attack on Square, and plenty of other techniques that follow the same pattern are discussed in later chapters of this thesis.

Although this 'bottom-up' approach to cryptanalysis has led to important advances, it also has its downsides – and there are good reasons to be critical of progress in academic cryptanalysis. Thirty years of research in differential cryptanalysis did not result in a general way to estimate the probability of differentials without relying on independence heuristics. The theory of linear cryptanalysis is fragmentary at best, making it difficult to advance beyond the analysis of linear trails. It took more than fifteen years to develop the potential of integral cryptanalysis, and even the division property does not fully consolidate the structural and algebraic approach to integral attacks. These examples illustrate three general problems in symmetric-key cryptanalysis:

(i) A lack of urgency to investigate assumptions leads to errors and missed opportunities. Heuristics can be useful, but they must be understood.

(ii) A lack of unification leads to duplication of effort.

(iii) A lack of perspective on the development of new techniques results in relatively few new proposals of general cryptanalytic methods.

Part I of this thesis hopes to contribute to the solution of these issues. To do so, an attempt will be made to approach cryptanalysis from the other end: rather than deducing theory from specific attacks, a general approach to symmetric-key cryptanalysis will be developed. This program intends to explain the extraordinary success of linear, differential and integral cryptanalysis from first principles. It aims to fill the gaps in existing results, to unify different techniques, and to suggest where to look for new methods.

Chapter 2 introduces and develops a proposal for such a general approach. It is used to reconstruct and extend the existing theory of linear, differential and

integral cryptanalysis in Chapters 3 to 5. For reasons that will be clarified in Chapter 2, it will be called the *geometric approach*.

Chapter 3 explores the consequences of the geometric approach in the context of linear cryptanalysis. The focus is on generalizations of linear cryptanalysis that have been proposed in the literature, in particular those that must be described by the higher-dimensional case of the geometric approach.

Chapter 4 is concerned with differential cryptanalysis, with emphasis on the problem of the independence heuristic. The one-dimensional case of the theory from Chapter 2 is applied.

Finally, Chapter 5 investigates the consequences of the one-dimensional case of the geometric approach for integral cryptanalysis. If the reader had any doubts about the inclusion of integral attacks alongside differential and linear cryptanalysis in Section 1.4, then Chapter 5 intends to dispel those.

1.5.2 Applications

Part II of this thesis serves a double purpose. On the one hand, it provides applications of the theoretical concepts introduced in Part I. On the other hand, it contributes to the cryptanalysis of concrete primitives.

Chapters 6 to 8 rely on Part I of this thesis. Chapter 6 discusses block cipher invariants from the point of view of Chapter 3 and contains attacks on round-reduced Midori-64 and MANTIS. Linear attacks on the tweakable block ciphers FEA-1, FEA-2 and the NIST standard FF3-1 are presented in Chapter 7. Finally, Chapter 8 reevaluates the differential cryptanalysis of Rectangle, KNOT and Speck using the methods from Chapter 4.

Chapters 9 and 10 present attacks that do not directly rely on the geometric approach, although in hindsight the results from Part I would have led to improvements in some cases. Chapter 9 presents attacks on generalized Feistel ciphers, with applications to the cryptanalysis of SM4. Further applications to GMiMC-crf and GMiMC-erf are given in Chapter 10, in addition to attacks on HadesMiMC and the Legendre PRF. These 'arithmetization-oriented primitives' are optimized for use in cryptographic protocols such as zero-knowledge proofs and multi-party computation.

Chapters 11 and 12 present applications of cryptanalysis outside of the traditional security analysis of primitives. Chapter 11 applies linear cryptanalysis to the analysis of countermeasures against side-channel attacks. Chapter 12 constructs block ciphers with intentional weaknesses or *backdoors*. In addition, an attack on an earlier such proposal, LowMC-M, is presented.

On the contribution of the author

Large parts of this thesis are based on published papers, some of which are joint work with other authors. For this reason, every chapter lists all relevant publications. Unless stated otherwise, all results presented in this thesis are due to the author alone.

I

Theory

2

Geometric approach to cryptanalysis

This chapter develops a general approach to the combinatorics of symmetric-key cryptanalysis. It is based on the observation that many combinatorial problems in cryptanalysis can be approached using the geometry and linear algebra of normed vector spaces.

The term 'geometric approach' was first used in the paper "A geometric approach to linear cryptanalysis" [40] (Asiacrypt 2021), which applied the techniques from this chapter to the particular case of linear cryptanalysis. Nevertheless, the results in this chapter have not appeared elsewhere in the same generality. Particular cases of the theory, such as linear cryptanalysis, are worked out in Chapters 3 to 5.

2.1 Introduction

As discussed in Section 1.4, the last three decades of research in symmetric-key cryptanalysis have been dominated by three major families of techniques: linear, differential and integral cryptanalysis. Part I of this thesis shows that these three techniques can be described in a uniform way. What is more, applying the same set of general principles results in new insights for each case. This chapter sets up the required definitions and derives these principles.

At their core, linear, differential and integral cryptanalysis involve a combinatorial problem. In linear and differential cryptanalysis, the cryptanalyst builds a distinguisher by comparing the observed number of solutions to certain equations to a theoretically computed value. Calculating the exact number of solutions is too difficult in most cases, so it is necessary to rely on approximations. This description may not seem to be applicable to integral cryptanalysis, where the cryptanalyst only determines the parity of the number of solutions. However, it will be shown in Chapter 5 that this can also be interpreted as a form of approximation that fits within the same framework. From this point of view, this chapter essentially develops a method for approximate counting.

The starting point of the geometric approach is a reformulation of cryptanalytic properties, such as linear approximations and differentials, in terms of pairs

of vector spaces. The input space corresponds to one or more functions that assign weights to the inputs of a primitive. The output space consists of linear functions that can be evaluated on the output weighting. Each such evaluation is the solution of a combinatorial problem of the type described above. This correspondence between cryptanalytic properties and pairs of vector spaces is explained in detail in Section 2.3.

The remainder of the chapter develops methods for approximate evaluation of cryptanalytic properties. The techniques are introduced gradually: Section 2.4 discusses the simpler case of properties defined by two one-dimensional vector spaces. This requires little more than expressing the results of Section 2.3 in an appropriately chosen basis. The principles for choosing this basis are the same for linear, differential and integral cryptanalysis and are also outlined in this section. The main result is a general theory of one-dimensional trails – linear trails, differential characteristics and division trails are examples that will be developed in Chapters 3, 4 and 5 respectively. Section 2.5 discusses the general case from a basis-free point of view. Apart from a generalized definition of trails, some results about perfect and zero-correlation approximations are given.

Section 2.6 provides a blueprint for specializing the theory to particular cases.

2.2 Linear algebra

This section elaborates on the mathematical setting for the theory that will be developed in Sections 2.3 to 2.5. Since no new results are presented, it may be safely skipped by readers who are familiar with the material.

Let V be a finite-dimensional vector space over a field k. The algebraic dual of V is introduced in Section 2.2.1, and its main properties are discussed to the extent that they will be used in the next sections. Section 2.2.2 introduces a metric structure on V and its dual. Finally, Section 2.2.3 reviews tensor product spaces and how they relate to the concepts from Sections 2.2.1 and 2.2.2.

2.2.1 Dual vector space

Recall that every vector space has a dual vector space, defined as in Definition 2.1. Like all results in this section, the following definition can be found in most linear algebra textbooks.

Definition 2.1 (Dual vector space)**.** Let V be a vector space over a field k. The dual space $V^\vee$ of V is the k-vector space of all linear functions $V \to k$. The elements of $V^\vee$ are called linear functionals.

It is not difficult to see that a linear combination of two linear functions is again a linear function, so that the set $V^\vee$ defined in Definition 2.1 is indeed a vector space. The following result shows that $\dim V = \dim V^\vee$ by explicitly constructing a basis for $V^\vee$.

Theorem 2.1 (Dual bases)**.** *Let V be a k-vector space with basis $\{b_1, \ldots, b_d\}$. If the linear functions $b^i : V \to k$ with i in $\{1, \ldots, d\}$ are defined by*

$$b^i(b_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases}$$

then $\{b^1, \ldots, b^d\}$ is a basis for the dual space $V^\vee$ of V.

The bases $\{b_1, \ldots, b_d\}$ and $\{b^1, \ldots, b^d\}$ in Theorem 2.1 are called dual bases.

Since $\dim V$ and $\dim V^\vee$ are equal, the vector spaces V and $V^\vee$ are isomorphic. Indeed, one can map a basis of V to a basis of $V^\vee$. The choice of isomorphism is arbitrary because different bases usually result in different isomorphisms, and will be avoided for now to avoid certain technical difficulties later on[1].

However, there is a 'canonical' isomorphism between V and $V^{\vee\vee}$ that can be specified without such an arbitrary choice of basis. It is given in Theorem 2.2.

Theorem 2.2. *Let V be a finite-dimensional vector space. For all v in V, define an 'evaluation map' $\mathsf{ev}_v : V^\vee \to k$ by $\mathsf{ev}_v(f) = f(v)$. The function $V \to V^{\vee\vee} : v \mapsto \mathsf{ev}_v$ is an isomorphism of vector spaces.*

Finally, recall that every subspace U of V has an annihilator. This is the subspace of linear functionals that vanish on U.

Definition 2.2 (Annihilator)**.** Let V be a finite-dimensional vector space. The annihilator of a subspace U of V is the subspace

$$U^0 = \left\{ v \in V^\vee \mid \forall u \in U : v(u) = 0 \right\}.$$

Applying Definition 2.2 to $V^\vee$ and using the canonical isomorphism between $V^{\vee\vee}$ and V, one likewise obtains the 'annihilator' of a subspace U of $V^\vee$ as

$$U^0 = \left\{ v \in V \mid \forall u \in U : u(v) = 0 \right\},$$

which is just the solution space of the system of equations defined by U.

[1] Choosing such an isomorphism is equivalent to choosing a bilinear form on V. However, such a form may be isotropic. For example, if k is a finite field with odd characteristic and $\dim V \geq 3$, then any bilinear form on V is isotropic [96, Theorem 6.23]. This would lead to artificial limitations in Section 2.5.

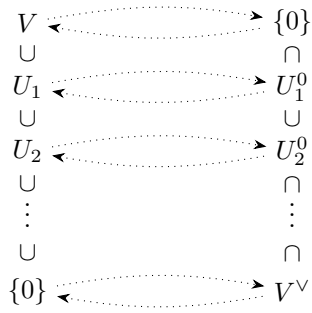


Figure 2.1: Antitone Galois connection between the subspaces of V and $V^\vee$.

It holds that $\dim U^0 = \dim V - \dim U$ and $U^{00} = U$. Taking the annihilator of a subspace is an inclusion-reversing operation, as illustrated in Figure 2.1. In other words, it is an antitone Galois connection [111, §7.23] between the lattices of subspaces of V and $V^\vee$.

2.2.2 Normed vector spaces

In Sections 2.3 to 2.5, suitable definitions of the magnitude of elements of k and the 'length' of vectors are needed. These concepts will be used to compare the quality of different cryptanalytic properties, and to measure the size of approximation errors.

Technically, V will be assumed to be a *normed* vector space. However, before norms can be introduced, the absolute value of elements of k must be defined. Additional background about absolute value functions can be found in textbooks such as [93, §8.2].

Definition 2.3 (Absolute value)**.** Let k be a field. An absolute value on k is a real-valued function $|\cdot| : k \to \mathbb{R}$ on k such that

(1) For all x in k, $|x| \geq 0$ with equality if and only if $x = 0$.

(2) The function $|\cdot|$ is multiplicative: for all x and y in k, $|xy| = |x|\,|y|$.

(3) The triangle-inequality holds: for all x and y in k, $|x + y| \leq |x| + |y|$.

Furthermore, if the strong triangle-inequality $|x+y| \leq \max\{|x|, |y|\}$ holds, then $|\cdot|$ is called a non-Archimidean or ultrametric absolute value. Otherwise, $|\cdot|$ is called Archimidean.

Definition 2.3 is inspired by the standard absolute value function on $\mathbb{Q}$ and $\mathbb{R}$. Another elementary example is the absolute value on $\mathbb{C}$ defined by $|a+\sqrt{-1}\,b| = \sqrt{a^2+b^2}$. Another example is given in Example 2.1. It will play a central role in Chapter 5.

Example 2.1. Let p be a prime. Every nonzero rational number x can be written as $x = p^e\,(a/b)$ with a and b integers indivisible by p. Let $|x|_p = p^{-e}$ and $|0|_p = 0$. For example, $|10|_2 = 1/2$. The function $x \mapsto |x|_p$ is a non-Archimidean absolute value on $\mathbb{Q}$. It is called the p-adic absolute value. ▷

Definition 2.3 provides a suitable notion of 'magnitude' for the elements of a field k. An absolute value also induces a metric: the distance between x and y in k is defined by $|x-y|$. Normed vector spaces over a field k with an absolute value can now be defined.

Definition 2.4 (Normed vector space)**.** Let V be a vector space over a field k with absolute value $|\cdot|$. A norm on V is a real-valued function $\|\cdot\| : V \to \mathbb{R}$ on V such that

(1) For all x in V, $\|x\| \geq 0$ with equality if and only if $x = 0$.

(2) For all x in V and λ in k, $\|\lambda\, x\| = |\lambda|\,\|x\|$.

(3) The triangle-inequality holds: for all x and y in V, $\|x+y\| \leq \|x\| + \|y\|$.

Furthermore, if the strong triangle-inequality $\|x+y\| \leq \max\{\|x\|, \|y\|\}$ holds, then $|\cdot|$ is called a non-Archimidean or ultrametric norm. Otherwise, $\|\cdot\|$ is called Archimidean. A vector space with a norm is called a normed vector space.

Several examples of norms on vector spaces are given in Example 2.2.

Example 2.2. Several norms can be defined on the vector spaces $\mathbb{Q}^n$, $\mathbb{R}^n$ and $\mathbb{C}^n$ with respect to the standard absolute value $|\cdot|$ on $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$. The p-norm of x is defined by

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

The p-norm with $p = 2$ is also called the Euclidean norm.

Let p be a prime and $|\cdot|_p$ the p-adic absolute value on $\mathbb{Q}$ from Example 2.1. A corresponding norm on $\mathbb{Q}^n$ can be defined as

$$\|x\| = \max_{1\leq i\leq n} |x_i|_p\,.$$

This ultrametric norm will be important in Chapter 5. ▷

The following theorem shows that the dual of a finite-dimensional normed vector space is again a normed vector space.

Theorem 2.3 (Dual norm)**.** *Let V be a finite-dimensional normed vector space with norm $\|\cdot\|$. The function $\|\cdot\|^\vee : V^\vee \to \mathbb{R}$ defined by*

$$\|u\|^\vee = \sup_{\substack{v \in V \\ \|v\| \leq 1}} |u(v)|\,,$$

for all u in $V^\vee$, is a norm on $V^\vee$.

The norm $\|\cdot\|^\vee$ from Theorem 2.3 is called the *dual norm* of $\|\cdot\|$. If k is complete with respect to $|\cdot|$, then the supremum in Theorem 2.3 may be replaced by a maximum. Using this norm, the isomorphism between V and $V^{\vee\vee}$ from Theorem 2.2 becomes an isometry provided that k is complete: $\|\mathsf{ev}_v\|^{\vee\vee} = \|v\|$.

Example 2.3. The dual norm of the p-norm from Example 2.2 is the $1/(1-1/p)$-norm with respect to the dual basis of the standard basis. In particular, the 2-norm is its own dual. The p-adic norm defined in Example 2.2 is also self-dual relative to the dual basis of the standard basis. ▷

Finally, the operator norm of a linear map $L : U \to V$ between normed vector spaces U and V is defined as

$$\|L\|_{\mathsf{op}} = \sup_{\substack{u \in U \\ \|u\|_U \leq 1}} \|Lu\|_V\,,$$

with $\|\cdot\|_U$ the norm on U and $\|\cdot\|_V$ the norm on V.

2.2.3 Tensor products

The tensor product of vector spaces can be defined in one of several ways. The most general definition is that the tensor product of k-vector spaces $V_1, \ldots, V_n$ is another k-vector space $V_1 \otimes \cdots \otimes V_n$ of dimension $\prod_{i=1}^n \dim V_i$ together with a multilinear map $\otimes : \prod_{i=1}^n V_i \to \bigotimes_{i=1}^n V_i$, which has the universal property that it uniquely linearizes arbitrary multilinear maps. Specifically, for any $T : \prod_{i=1}^n V_i \to W$ linear in each variable (multilinear), there exists a unique *linear* map $L : \bigotimes_{i=1}^n V_i \to W$ such that $T(v_1, \ldots, v_n) = L(v_1 \otimes \cdots \otimes v_n)$. This characterizes the tensor product up to unique isomorphism of vector spaces.

A more concrete but basis-dependent definition is as follows. Let $\mathcal{B}_i$ be a basis for V_i. The tensor product space $V_1 \otimes \cdots \otimes V_n$ can be informally defined as

$$V_1 \otimes \cdots \otimes V_n = \mathrm{Span}\left\{b_1 \otimes b_2 \otimes \cdots \otimes b_n \;\middle|\; b_i \in \mathcal{B}_i \text{ for } i = 1, \ldots, n\right\},$$

where $b_1 \otimes b_2 \otimes \cdots \otimes b_n$ are formal basis vectors. This can be formalized as a quotient of the free vector space. Furthermore, the tensor product of vectors $v_1, \ldots, v_n$ with $v_i = \sum_{b \in \mathcal{B}_i} c_b \, b$ is defined by bilinearity:

$$v_1 \otimes \cdots \otimes v_n = \sum_{b_1 \in \mathcal{B}_1} \cdots \sum_{b_n \in \mathcal{B}_n} \left(\prod_{i=1}^n c_{b_i} \right) b_1 \otimes \cdots \otimes b_n \, .$$

Since the dual spaces $V_1^\vee, \ldots, V_n^\vee$ are themselves k-vector spaces, the above also defines the tensor product space $V_1^\vee \otimes \cdots \otimes V_n^\vee$. Throughout this thesis, this space will be identified with $(V_1 \otimes \cdots \otimes V_n)^\vee$ using the canonical isomorphism that maps $f_1 \otimes \cdots \otimes f_n$ to $(x_1 \otimes \cdots \otimes x_n) \mapsto f_1(x_1) \cdots f_n(x_n)$.

More generally, the set of linear operators between two k-vector spaces is itself a vector space over k. Hence, the tensor product of linear operators is well-defined. Throughout this thesis, the tensor product $L_1 \otimes \cdots \otimes L_n$ of linear maps $L_i : V_i \to U_i$ will be identified with the linear map

$$\bigotimes_{i=1}^n V_i \to \bigotimes_{i=1}^n U_i$$

$$v_1 \otimes \cdots \otimes v_n \mapsto (L_1 v_1) \otimes \cdots \otimes (L_n v_n) \, .$$

In general, norms on the spaces $V_1, \ldots, V_n$ do not extend to $V_1 \otimes \cdots \otimes V_n$ in a canonical way. However, all of the norms that will be used in this thesis satisfy $\|v_1 \otimes \cdots \otimes v_n\| = \|v_1\| \cdots \|v_n\|$.

The following definition will be important in Chapter 3.

Definition 2.5 (Tensor rank)**.** Let $V_1, \ldots, V_n$ be k-vector spaces. The rank of a vector v in $\bigotimes_{i=1}^n V_i$ is the least integer $r \geq 0$ such that

$$v = \sum_{i=1}^r \lambda_i \, v_1^{(i)} \otimes \cdots \otimes v_n^{(i)} \, ,$$

where $v_1^{(i)} \in V_1, \ldots, v_n^{(i)} \in V_n$ and $\lambda_1, \ldots, \lambda_r \in k$.

Example 2.4. Let $V = \mathbb{R}^2 \otimes \mathbb{R}^2$. The vector $(1,0) \otimes (1,0)$ in $\mathbb{R}^2 \otimes \mathbb{R}^2$ has tensor rank one. Furthermore, it is easy to check that $(1,0) \otimes (1,0) + (0,1) \otimes (0,1)$ has rank two. However, $(1,0) \otimes (1,0) + (1,0) \otimes (0,1) + (0,1) \otimes (1,0) + (0,1) \otimes (0,1)$ has rank one because it is equal to $(1,1) \otimes (1,1)$. ▷

2.3 Cryptanalytic properties

This section introduces the class of combinatorial problems that Part I of this thesis intends to address. The goal is to approximately evaluate cryptanalytic

properties. Section 2.3.1 formalizes cryptanalytic properties as a pair of vector spaces. The effect of applying a function on these subspaces is discussed in Section 2.3.2. Finally, a general definition of correlation is given in Section 2.3.3.

2.3.1 Properties

Let X and Y be finite sets and $\mathsf{F} : X \to Y$ a function. In concrete cases of the theory, the sets X and Y will be related to the state space of a cryptographic primitive. For example, X could be the set of all possible inputs, or the set of all possible input pairs. The function F will be the primitive itself or function derived from it, such as an extension that works on pairs.

In the combinatorial problems that we consider, each element of X is assigned a weight. It will be assumed that these weights are numbers, *i.e.* elements of some field k. Mathematically, such an assignment of weights is described by a function from X to k. The set of all functions from X to k is denoted by k^X. The sum of functions f and g in k^X is the function $f + g$ defined by $(f + g)(x) = f(x) + g(x)$. Similarly, a function f can be multiplied by a scalar λ to obtain a function λf defined by $(\lambda f)(x) = \lambda f(x)$. Hence, k^X is a vector space over the field k. Alternatively, one can think of k^X as the free k-vector space over the set X.

Applying a function $\mathsf{F} : X \to Y$ to the state transforms the assignment of weights on X to a corresponding assignment on Y. In the simplest case, when $X = Y$ and F is a permutation, it leads to a rearrangement of the weights that were assigned to elements of X. Section 2.3.2 below describes the effect of general functions $\mathsf{F} : X \to Y$. For now, it is sufficient to say that the result is characterized by some function $T^{\mathsf{F}} : k^X \to k^Y$.

It is rarely necessary to keep track of all the weights assigned to elements of Y. Instead, it is sufficient to know the evaluation of a function $k^Y \to k$ on the vector of output weights. In this thesis only linear functions, *i.e.* elements of the dual space $(k^Y)^\vee$ of k^Y, are considered. This assumption may seem stringent at first, but it is more than sufficient to describe all of the techniques mentioned in Section 1.4. This is not a coincidence, since linearity is implied by several rudimentary properties such as being able to evaluate properties by summing values obtained from individual input-output pairs.

Summarizing the above, the problem reduces to evaluating an expression of the form $v(T^{\mathsf{F}}u)$, for u in k^X and v in $(k^Y)^\vee$. Cryptanalysis often involves multiple such expressions. That is, one is interested in the value of $v(T^{\mathsf{F}}u)$ for all u in U and v in V, where $U \subseteq k^X$ and $V \subseteq (k^Y)^\vee$ are subsets. Due to linearity, U and V can be assumed to be subspaces. This leads to the following definition.

Definition 2.6 (Cryptanalytic property)**.** A cryptanalytic property for a function $\mathsf{F} : X \to Y$ is a pair (U, V) with U a subspace of k^X and V a subspace of $(k^Y)^\vee$. The evaluation of a property at u in U and v in V is equal to $v(T^{\mathsf{F}}u)$.

The problems addressed in Chapters 2 to 5 of this thesis can now be described as follows: given a property (U, V) for a function F, estimate its evaluations $v(T^{\mathsf{F}}u)$ at arbitrary u in U and v in V. The meaning of 'estimate' is with respect to a metric structure on k. For this purpose, it is assumed that the field k comes with an absolute value function $|\cdot| : k \to \mathbb{R}$.

To end this section, a basic example of a cryptanalytic property in the sense of Definition 2.6 is given. More significant examples will be discussed in Chapters 3 to 5.

Example 2.5. Consider a distinguisher that relies on encrypting elements of a set A and counting the number of outputs that are elements of a set B. The sets A and B together determine a cryptanalytic property that can be brought into the form of Definition 2.6 as follows.

Let $U = \mathrm{Span}\{\mathbb{1}_A\}$, where $\mathbb{1}_A : X \to k$ is the indicator function of a subset A of X. In particular, $\mathbb{1}_A$ is defined by $\mathbb{1}_A(x) = 1$ if $x \in A$ and zero elsewhere. Furthermore, let $V = \mathrm{Span}\{\mathbb{S}_B\}$ with $\mathbb{S}_B : k^Y \to k$ the 'summation' functional defined by $\mathbb{S}_B(f) = \sum_{x \in B} f(x)$.

If F is a permutation, then $T^{\mathsf{F}}\mathbb{1}_A = \mathbb{1}_{\mathsf{F}(A)}$. Hence, the evaluation of the cryptanalytic property (U, V) at $\mathbb{1}_A$ and $\mathbb{S}_B$ is equal to

$$\mathbb{S}_B(T^{\mathsf{F}}\mathbb{1}_A) = \sum_{x \in A} \mathbb{1}_B(\mathsf{F}(x)) = |\{x \in A \mid \mathsf{F}(x) \in B\}|\,. \tag{2.1}$$

Even if F is not a permutation, the above equality remains valid. This will be shown in Section 2.3.2. ▷

2.3.2 Propagation

This section describes how an assignment of weights to the state changes when a function is applied. This will be referred to as *propagation*, in analogy to the propagation of masks and differences in linear and differential cryptanalysis. In fact, it will be shown in Chapters 3 to 5 that the familiar rules for propagation are simple consequences of the results in this section.

As mentioned in Section 2.3.1, the effect of applying a function F to the state can be described by a function $T^{\mathsf{F}} : k^X \to k^Y$. This function is defined formally

in Definition 2.7. Below, for all x in X, the function $\delta_x : X \to k$ is defined by

$$\delta_x(y) = \begin{cases} 1 & \text{if } y = x\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

The functions δ_x with x in X are linearly independent. Since $f = \sum_{x \in X} f(x)\,\delta_x$ for any f in k^X, the set $\{\delta_x \mid x \in X\}$ is a basis for k^X. It will be called the *standard basis* of k^X.

Definition 2.7 (Pushforward operator)**.** Let $\mathsf{F} : X \to Y$ be a function. The pushforward operator along F is the linear map $T^{\mathsf{F}} : k^X \to k^Y$ defined by

$$T^{\mathsf{F}}\,\delta_x = \delta_{\mathsf{F}(x)}\,,$$

for all x in X.

As mentioned in Section 2.3.2, if F is a permutation, then applying T^{F} amounts to a rearrangement of weights. If F is not a permutation, then Definition 2.7 implies that the weight of y in Y is obtained by adding the weights of all the preimages of y under F. That is, for any f in k^X, it holds that

$$\big(T^{\mathsf{F}} f\big)(y) = \sum_{x \in \mathsf{F}^{-1}(y)} f(x) = \sum_{x \in X} \delta_y(\mathsf{F}(x)) f(x)\,. \tag{2.2}$$

Since T^{F} is a linear operator, it can be represented as a matrix. For convenience, the matrix representation of T^{F} with respect to the standard bases of k^X and k^Y will also be denoted by T^{F}. By (2.2), the coordinates of T^{F} are equal to

$$T^{\mathsf{F}}_{y,x} = \delta_y(\mathsf{F}(x))\,.$$

Since the standard bases of k^X and k^Y are indexed by elements of X and Y, using the same convention to denote the coordinates of T^{F} avoids arbitrary choices. This result implies (2.1) in Example 2.5 for arbitrary functions F, since $(T^{\mathsf{F}} \mathbb{1}_A)(y) = \sum_{x \in A} \delta_y(\mathsf{F}(x))$ and hence $\mathbb{S}_B(T^{\mathsf{F}} \mathbb{1}_A) = \sum_{y \in B} \sum_{x \in A} \delta_y(\mathsf{F}(x))$.

Example 2.6. Let $\mathsf{F} : \mathbb{F}_2^2 \to \mathbb{F}_2^2$ be the function defined by $\mathsf{F}(x) = (x_1, x_1 x_2)$. The matrix T^{F} is given by:

$$T^{\mathsf{F}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where the standard basis of $k^{\mathbb{F}_2^2}$ is ordered as $\delta_{(0,0)}, \delta_{(0,1)}, \delta_{(1,0)}, \delta_{(1,1)}$. ▷

Definition 2.7 and the discussion following it describe propagation in the forward direction. However, it is sometimes useful to start from an element of $(k^Y)^\vee$ and propagate it backwards through a function F to obtain an element of $(k^X)^\vee$. The natural way to do this is given in Definition 2.8. In this definition, the functionals $\delta^y : k^Y \to k$ are defined by $\delta^y(f) = f(y)$. The set $\{\delta^y \mid y \in Y\}$ is the standard basis of $(k^Y)^\vee$. It is the dual basis of the standard basis of k^Y.

Definition 2.8 (Pullback operator)**.** Let $\mathsf{F} : X \to Y$ be a function. The pullback operator along F is the linear map $T^{\mathsf{F}^\vee} : (k^Y)^\vee \to (k^X)^\vee$ defined by

$$T^{\mathsf{F}^\vee}\, \delta^y = \delta^y \circ T^{\mathsf{F}}\,,$$

for all y in Y.

An alternative formulation of Definition 2.8 is that the pullback operator $T^{\mathsf{F}^\vee}$ is the adjoint of the pushforward operator T^{F}. In other words, forward and backward propagation are dual to each other. Indeed, Definition 2.8 implies that all v in $(k^Y)^\vee$ satisfy $T^{\mathsf{F}^\vee} v = v \circ T^{\mathsf{F}}$. Combining Definitions 2.7 and 2.8 shows that for all u in k^X and v in $(k^Y)^\vee$,

$$\big(T^{\mathsf{F}^\vee} v\big)(u) = v\big(T^{\mathsf{F}} u\big)\,.$$

It also follows that the matrix representation of $T^{\mathsf{F}^\vee}$ with respect to the standard bases of $(k^X)^\vee$ and $(k^Y)^\vee$ is the transpose of the matrix T^{F}. Specifically,

$$T^{\mathsf{F}^\vee}_{x,y} = \big(T^{\mathsf{F}^\vee} \delta^y\big)(\delta_x) = \delta_y(\mathsf{F}(x)) = T^{\mathsf{F}}_{y,x}.$$

The following example constructs $T^{\mathsf{F}^\vee}$ for the function F from Example 2.6.

Example 2.7. For the function $\mathsf{F} : \mathbb{F}_2^2 \to \mathbb{F}_2^2$ defined by $\mathsf{F}(x) = (x_1, x_1x_2)$ from Example 2.6, the matrix corresponding to the pullback operator is

$$T^{\mathsf{F}^\vee} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where the standard basis of $(k^{\mathbb{F}_2^2})^\vee$ is ordered as $\delta^{(0,0)}, \delta^{(0,1)}, \delta^{(1,0)}, \delta^{(1,1)}$. It is the transpose of the matrix given in Example 2.6. ▷

Theorem 2.4 states the main properties of pushforward and pullback operators. They are all simple to state and prove. However, as shown in Sections 2.4 and 2.5 and especially Chapters 3 to 5, their consequences are profound.

Note that property (1) holds up to the canonical isomorphism between k^X and $\bigotimes_{i=1}^n k^{X_i}$ for $X = \prod_{i=1}^n X_i$. This isomorphism maps the standard basis vector $\delta_x = \delta_{(x_1,\ldots,x_n)}$ to the tensor $\bigotimes_{i=1}^n \delta_{x_i}$.

Theorem 2.4 (Properties of pushforward and pullback operators.)**.** *Let* $\mathsf{F} : X \to Y$ *be a function. The pushforward operator* T^{F} *and pullback operator* $T^{\mathsf{F}^\vee}$ *satisfy the following properties:*

(1) If $\mathsf{F}(x) = (\mathsf{F}_1(x_1), \ldots, \mathsf{F}_n(x_n))$*, then* $T^{\mathsf{F}} = \bigotimes_{i=1}^n T^{\mathsf{F}_i}$ *and* $T^{\mathsf{F}^\vee} = \bigotimes_{i=1}^n T^{\mathsf{F}_i^\vee}$.

(2) If $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$*, then* $T^{\mathsf{F}} = T^{\mathsf{F}_r} \cdots T^{\mathsf{F}_1}$ *and* $T^{\mathsf{F}^\vee} = T^{\mathsf{F}_1^\vee} \cdots T^{\mathsf{F}_r^\vee}$.

The same properties apply to the matrix-representations of T^{F} *and* $T^{\mathsf{F}^\vee}$*, with the tensor product* $\otimes$ *corresponding to the Kronecker product of matrices and the composition of linear maps to matrix multiplication.*

Proof. Property (1) follows from

$$T^{\mathsf{F}}\delta_x = \delta_{\mathsf{F}(x)} = \bigotimes_{i=1}^n \delta_{\mathsf{F}_i(x_i)} = \bigotimes_{i=1}^n T^{\mathsf{F}_i}\delta_{x_i} = \left(\bigotimes_{i=1}^n T^{\mathsf{F}_i}\right) \bigotimes_{i=1}^n \delta_{x_i}\,,$$

for all x in X. The proof of the corresponding property for $T^{\mathsf{F}^\vee}$ is similar, or alternatively follows from standard results about adjoint maps.

For property (2), it is sufficient to see that for all x in X,

$$T^{\mathsf{F}}\delta_x = \delta_{\mathsf{F}(x)} = \delta_{(\mathsf{F}_r \circ \cdots \circ \mathsf{F}_1)(x)} = T^{\mathsf{F}_r}\delta_{(\mathsf{F}_{r-1} \circ \cdots \circ \mathsf{F}_1)(x)} = \ldots = T^{\mathsf{F}_r} \cdots T^{\mathsf{F}_1}\delta_x\,.$$

Again, the proof of the corresponding property for $T^{\mathsf{F}^\vee}$ is similar, or can be deduced using standard results about adjoint maps. □

Example 2.8. Let $\mathsf{G} : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ be defined by $\mathsf{G}(x) = (x_1, x_1x_2, x_3)$. Since G is essentially the same as $(\mathsf{F}, \mathsf{id})$ with F defined in Example 2.6 and $\mathsf{id}(x_3) = x_3$, Theorem 2.4 (1) yields

$$T^{\mathsf{G}} = T^{\mathsf{F}} \otimes T^{\mathsf{id}} = \left[\begin{array}{cc:cc:cc:cc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

where the standard basis of $k^{\mathbb{F}_2^3}$ is ordered as $\delta_{(0,0,0)}, \delta_{(0,0,1)}, \ldots, \delta_{(1,1,1)}$. In addition, one can verify that $T^{\mathsf{G}}T^{\mathsf{G}} = T^{\mathsf{G}\circ\mathsf{G}} = T^{\mathsf{G}}$ as expected from Theorem 2.4 (2). ▷

2.3.3 Correlation

Before turning to methods to approximate evaluations of cryptanalytic properties, a final comment on Definition 2.6 is in order. In order to compare different properties, it is sometimes useful to define the 'principal correlation' of a property. This terminology is due to the link with absolute correlations in linear cryptanalysis, which will be explained in Chapter 3.

Definition 2.9 (Principal correlation)**.** Let k be a field with absolute value $|\cdot|$ and let $\|\cdot\|_X$ and $\|\cdot\|_Y$ be norms on k^X and k^Y respectively. The principal correlation of a cryptanalytic property (U, V) for a function $\mathsf{F} : X \to Y$ with $U \subseteq k^X$ and $V \subseteq (k^Y)^\vee$ is equal to

$$\sup_{\substack{v \in V \\ \|v\|_Y^\vee \leq 1}} \sup_{\substack{u \in U \\ \|u\|_X \leq 1}} |v(T^{\mathsf{F}}\, u)|\,,$$

where $\|\cdot\|_Y^\vee$ denotes the dual norm of $\|\cdot\|_Y$.

The definition of the dual norm implies that the correlation is a value between zero and $\|T^{\mathsf{F}}\|_{\mathsf{op}}$. The principal correlation is sometimes a good measure of the quality of a property, but it is not the case that only properties with high correlation are useful. At the extreme end, properties with principal correlation equal to zero ('zero-correlation') have many applications.

It is worth emphasizing that the main purpose of the theory in Part I of this thesis is not to identify cryptanalytic properties with high principal correlation. Instead, the theory addresses the more basic problem of accurately evaluating properties.

2.4 One-dimensional theory

This section develops a simplified version of the more general theory that will be introduced in Section 2.5. The simplification is due to the fact that only 'one-dimensional' properties, *i.e.* with $\dim U = \dim V = 1$ in Definition 2.6, are considered. Restricting to this case has the advantage that most results can be obtained simply by expressing the results from Section 2.3.2 in an appropriately chosen basis. The downside of this approach is that it does not work well in the multidimensional case. Nevertheless, the one-dimensional case is important enough by itself to warrant a separate discussion.

2.4.1 Change-of-basis

Let $\mathcal{B} = \{b_1, b_2, \ldots, b_{|X|}\}$ be a basis for k^X. Recall that a change-of-basis transformation from the standard basis of k^X to $\mathcal{B}$ is an invertible linear map from $k^{|X|}$ to itself, which maps the standard basis coordinates of b_i to the i^{th} standard basis vector of $k^{|X|}$. In order to allow arbitrary labelings of the basis vectors, the following variation on this definition will be used in this thesis.

Definition 2.10 (Change-of-basis)**.** Let $\mathcal{B} = \{b_\beta \mid \beta \in B\}$ be a basis for k^X labeled by a set B. The change-of-basis transformation from the standard basis of k^X to the basis $\mathcal{B}$ is the linear map $P_{\mathcal{B}} : k^X \to k^B$ defined by $P_{\mathcal{B}}\, b_\beta = \delta_\beta$ for all β in B. Furthermore, the dual change-of-basis transformation is the linear map $P_{\mathcal{B}}^{-\vee} : (k^X)^\vee \to (k^B)^\vee$.

The dual change-of-basis transformation defined in Definition 2.10 maps the dual basis of $\mathcal{B}$ to the standard basis of $(k^B)^\vee$. Recall that the dual basis of $\mathcal{B}$ is the unique basis $\mathcal{B}^\vee = \{b^\beta \mid \beta \in B\}$ such that $b^\alpha(b_\beta) = \delta_\alpha(\beta)$ for all α and β in B. If $P_{\mathcal{B}}$ is the change-of-basis transformation for $\mathcal{B}$ as in Definition 2.10, then the linear map $P_{\mathcal{B}}^{-\vee} : (k^X)^\vee \to (k^B)^\vee$ satisfies

$$\big(P_{\mathcal{B}}^{-\vee} b^\alpha\big)(\delta_\beta) = \big(b^\alpha \circ P_{\mathcal{B}}^{-1}\big)(\delta_\beta) = b^\alpha(b_\beta)\,,$$

for all α and β in B. By the uniqueness of dual bases, it follows that $P_{\mathcal{B}}^{-\vee} b^\alpha = \delta^\alpha$. Hence, $P_{\mathcal{B}}^{-\vee}$ can be interpreted as a change-of-basis transformation from the standard basis of $(k^X)^\vee$ to the basis $\mathcal{B}^\vee$.

Example 2.9. Let $\mathcal{B} = \{b_1, b_2\} \subset k^{\mathbb{F}_2}$ with $b_1 = \delta_0 + \delta_1$ and $b_2 = \delta_1$. This is a basis for $k^{\mathbb{F}_2}$, with change-of-basis transformation given by

$$P_{\mathcal{B}} \quad = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix},$$

where the matrix-representation is with respect to the standard bases of k^B and $k^{\mathbb{F}_2}$. Indeed, the first standard basis vector δ_{b_1} of k^B is equal to $P_{\mathcal{B}}(\delta_0 + \delta_1)$ and $\delta_{b_2} = P_{\mathcal{B}}\delta_1$. The dual basis of $\mathcal{B}$ is given by $\mathcal{B}^\vee = \{b^1, b^2\}$ with $b^1 = \delta^0$ and $b^2 = \delta^1 - \delta^0$. The corresponding dual change-of-basis transformation is given by

$$P_{\mathcal{B}}^{-\vee} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

again with respect to the standard bases of $(k^B)^\vee$ and $(k^{\mathbb{F}_2})^\vee$. ▷

From a strictly theoretical point of view, Definition 2.10 (and change-of-basis in general) is redundant. Practically, however, a good choice of basis can simplify the matrix representation of the pushforward and pullback operators T^{F} and $T^{\mathsf{F}^\vee}$. Nevertheless, it will be clarified in Section 2.5 that the same results can be obtained without choosing a basis.

2.4.2 Propagation

This section takes up the same topic as Section 2.3.2, namely the propagation of state functions through a function $\mathsf{F} : X \to Y$. Recall that this is described by a pushforward operator $T^{\mathsf{F}} : k^X \to k^Y$, or dually by the corresponding pullback operator $T^{\mathsf{F}^\vee}$. Change-of-basis transformations can also be applied to these operators.

Definition 2.11 (Relative pushforward and pullback)**.** Let $\mathsf{F} : X \to Y$ be a function and $\mathcal{X}$ and $\mathcal{Y}$ bases for k^X and k^Y respectively. The pushforward operator of F relative to $\mathcal{X}$ and $\mathcal{Y}$ is the linear transformation $B^{\mathsf{F}} = P_{\mathcal{Y}} T^{\mathsf{F}} P_{\mathcal{X}}^{-1}$. The pullback operator relative to $\mathcal{X}$ and $\mathcal{Y}$ is the dual map $B^{\mathsf{F}^\vee}$.

Example 2.10. From the basis $\mathcal{B} = \{\delta_0 + \delta_1, \delta_1\}$ for $k^{\mathbb{F}_2}$ from Example 2.9, one can construct a basis $\mathcal{X} = \mathcal{Y} = \{\delta_{(0,0)} + \delta_{(0,1)} + \delta_{(1,0)} + \delta_{(1,1)}, \delta_{(0,1)} + \delta_{(1,1)}, \delta_{(1,0)} + \delta_{(1,1)}, \delta_{(1,1)}\}$ for $k^{\mathbb{F}_2^2}$ by taking tensor products. The pushforward operator of the function $\mathsf{F} : x \mapsto (x_1, x_1 x_2)$ from Example 2.6 relative to $\mathcal{X}$ and $\mathcal{Y}$ is then given by the matrix

$$B^{\mathsf{F}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 1 & 0 & 0 \\ -2 & -1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{bmatrix}.$$

The matrix representation of the relative pullback operator of F is the transpose of the above matrix. ▷

The relative pushforward operator B^{F} has the same properties as T^{F}. In particular, the following variation on Theorem 2.4 holds in the relative setting.

Theorem 2.5 (cf. Theorem 2.4)**.** *Let $\mathsf{F} : X \to Y$ be a function. The pushforward operator B^{F} and pullback operator $B^{\mathsf{F}^\vee}$ relative to bases $\mathcal{X}$ and $\mathcal{Y}$ satisfy the following properties:*

(1) Let $\mathcal{X}_1, \ldots, \mathcal{X}_n$ and $\mathcal{Y}_1, \ldots, \mathcal{Y}_n$ be bases such that $P_{\mathcal{X}} = \bigotimes_{i=1}^n P_{\mathcal{X}_i}$ and $P_{\mathcal{Y}} = \bigotimes_{i=1}^n P_{\mathcal{Y}_i}$. If $\mathsf{F}(x) = (\mathsf{F}_1(x_1), \ldots, \mathsf{F}_n(x_n))$, then

$$B^{\mathsf{F}} = \bigotimes_{i=1}^n B^{\mathsf{F}_i} \text{ and } B^{\mathsf{F}^\vee} = \bigotimes_{i=1}^n B^{\mathsf{F}_i^\vee},$$

where B^{F_i} *and* $B^{\mathsf{F}_i^\vee}$ *are relative to* $\mathcal{X}_i$ *and* $\mathcal{Y}_i$.

(2) Let $\mathcal{X}_1 = \mathcal{X}, \mathcal{X}_2, \ldots, \mathcal{X}_{r+1} = \mathcal{Y}$ *be bases. If* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$, *then*

$$B^{\mathsf{F}} = B^{\mathsf{F}_r} \cdots B^{\mathsf{F}_2}\, B^{\mathsf{F}_1} \text{ and } B^{\mathsf{F}^\vee} = B^{\mathsf{F}_1^\vee}\, B^{\mathsf{F}_2^\vee} \cdots B^{\mathsf{F}_r^\vee},$$

where B^{F_i} *and* $B^{\mathsf{F}_i^\vee}$ *are relative to* $\mathcal{X}_i$ *and* $\mathcal{X}_{i+1}$.

The same properties apply to the matrix-representations of B^{F} *and* $B^{\mathsf{F}^\vee}$, *with the tensor product* $\otimes$ *corresponding to the Kronecker product of matrices and the composition of linear maps to matrix multiplication.*

Proof. Both properties are direct consequences of the corresponding properties in Theorem 2.5. For (1), it follows from $T^{\mathsf{F}} = \bigotimes_{i=1}^{n} T^{\mathsf{F}_i}$ that

$$P_{\mathcal{Y}} T^{\mathsf{F}} P_{\mathcal{X}}^{-1} = \left(\bigotimes_{i=1}^{n} P_{\mathcal{Y}_i}\right)\left(\bigotimes_{i=1}^{n} T^{\mathsf{F}_i}\right)\left(\bigotimes_{i=1}^{n} P_{\mathcal{X}_i}^{-1}\right) = \bigotimes_{i=1}^{n} P_{\mathcal{Y}_i} T^{\mathsf{F}_i} P_{\mathcal{X}_i}^{-1}.$$

The result then follows by Definition 2.11. Property (1) follows from the equality $T^{\mathsf{F}} = T^{\mathsf{F}_r} \cdots T^{\mathsf{F}_2} T^{\mathsf{F}_1}$, since

$$P_{\mathcal{Y}}\, T^{\mathsf{F}}\, P_{\mathcal{X}}^{-1} = (P_{\mathcal{X}_{r+1}}\, T^{\mathsf{F}_r}\, P_{\mathcal{X}_r}^{-1}) \cdots (P_{\mathcal{X}_3}\, T^{\mathsf{F}_2}\, P_{\mathcal{X}_2}^{-1})(P_{\mathcal{X}_2}\, T^{\mathsf{F}_1}\, P_{\mathcal{X}_1}^{-1}).$$

Applying Definition 2.11 yields the result. □

Example 2.11. Consider the function $\mathsf{G} : (x_1, x_2, x_3) \mapsto (x_1, x_1 x_2, x_3)$ defined in Example 2.8. Construct a basis for $k^{\mathbb{F}_2^3}$ by tensoring the basis $\mathcal{B}$ from Example 2.9 three times. The relative pushforward matrix of G is given by

$$B^{\mathsf{G}} = B^{\mathsf{F}} \otimes B^{\mathsf{id}} = \begin{bmatrix} 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

It can be checked that $B^{\mathsf{G}} B^{\mathsf{G}} = B^{\mathsf{G} \circ \mathsf{G}} = B^{\mathsf{G}}$. ▷

2.4.3 Approximations and trails

This section considers cryptanalytic properties of a function $\mathsf{F} : X_1 \to X_{r+1}$ that can be decomposed as $\mathsf{F} = \mathsf{F}_r \circ \circ \cdots \mathsf{F}_2 \circ \mathsf{F}_1$ where $\mathsf{F}_i : X_i \to X_{i+1}$. For

$i = 1, \ldots, r$, fix a basis $\mathcal{B}_i = \{b_{\beta_i} \mid \beta_i \in B_i\}$ of k^{X_i}. For notational convenience, it is assumed that all basis vectors have the same norm.

The cryptanalytic properties we consider are of the form (U, V) with $U = \mathrm{Span}\{b_{\beta_1}\}$ and $V = \mathrm{Span}\{b^{\beta_{r+1}}\}$ with b_{β_1} in $\mathcal{B}_1$ and $b^{\beta_{r+1}}$ the dual basis vector of $b_{\beta_{r+1}}$ in $\mathcal{B}_{r+1}$. Recall from Section 2.3 that one is interested in evaluating $b^{\beta_{r+1}}(T^{\mathsf{F}}\, b_{\beta_1})$. By changing bases, one obtains

$$b^{\beta_{r+1}}(T^{\mathsf{F}} b_{\beta_1}) = \big(P^{\vee}_{\mathcal{B}_{r+1}}\, \delta^{\beta_{r+1}}\big) T^{\mathsf{F}} \big(P^{-1}_{\mathcal{B}_1}\, \delta_{\beta_1}\big) = \delta^{\beta_{r+1}} \big(P_{\mathcal{B}_{r+1}} T^{\mathsf{F}} P^{-1}_{\mathcal{B}_1}\big) \delta_{\beta_1} = B^{\mathsf{F}}_{\beta_{r+1},\beta_1}\,,$$

where B^{F} is relative to $\mathcal{B}_1$ and $\mathcal{B}_{r+1}$. Hence, evaluating the property (U, V) is equivalent to computing a coordinate of the standard-basis matrix-representation of B^{F}.

The pair (β_1, β_{r+1}) is an example of what will be called a forward approximation of F in Section 2.5. The name refers to the idea that we try to 'approximate' $T^{\mathsf{F}} b_{\beta_1}$ by a scalar multiple of $b_{\beta_{r+1}}$. To be consistent with the evaluation of the property, the multiplier should equal $B^{\mathsf{F}}_{\beta_{r+1},\beta_1}$. This quantity is called the correlation[2] of the approximation. Its absolute value equals the principal correlation of (U, V).

Although the difference between the approximation and the cryptanalytic property may seem minimal, it is useful to keep these concepts separate. For the general case discussed in Section 2.5, the differences are more apparent.

The main result about approximations is that their correlation can be computed as the sum of the correlations of *trails*. A trail is a tuple of $r + 1$ basis functions that defines a chain of compatible approximations for the intermediate functions F_i. For instance, the trail $(\beta_1, \beta_2, \ldots, \beta_{r+1})$ defines the approximations (β_1, β_2), $(\beta_2, \beta_3), \ldots, (\beta_r, \beta_{r+1})$.

Theorem 2.6 (Sum of one-dimensional trails)**.** *Let* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$. *The matrix representation of* B^{F} *has standard-basis coordinates*

$$B^{\mathsf{F}}_{\beta_{r+1},\beta_1} = \sum_{\beta_2,\ldots,\beta_r} \prod_{i=1}^{r} B^{\mathsf{F}_i}_{\beta_{i+1},\beta_i}\,,$$

where $\beta_1 \in B_1$, $\beta_{r+1} \in B_{r+1}$ *and the sum is over all* $(\beta_2, \ldots, \beta_r)$ *in* $\prod_{i=2}^{r} B_i$.

In practice, such as for the bases chosen in Chapters 3 to 5 and for common primitives, Theorem 2.6 is not used directly because the number of trails is too large. Instead, the sum is restricted to a set of 'dominant' trails to estimate the correlation.

[2] Assuming that all basis vectors have the same length, in particular $\|b_{\beta_1}\| = \|b_{\beta_{r+1}}\|$.

Corollary 2.1 (Dominant trail approximation)**.** *Let* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$*. For all subsets* Λ *of the set* Ω *of all trails from* β_1 *in* B_1 *to* β_{r+1} *in* B_{r+1}*,*

$$\left| B^{\mathsf{F}}_{\beta_{r+1},\beta_1} - \sum_{\beta \in \Lambda} \prod_{i=1}^{r} B^{\mathsf{F}_i}_{\beta_{i+1},\beta_i} \right| \leq \left| \sum_{\beta \in \Omega \setminus \Lambda} \prod_{i=1}^{r} B^{\mathsf{F}_i}_{\beta_{i+1},\beta_i} \right| .$$

In practice, a good enough error bound is often difficult to obtain. Hence, it is common to rely on the assumption that if $|\prod_{i=1}^{r} B^{\mathsf{F}_i}_{\beta_{i+1},\beta_i}|$ is much smaller for β in $\Omega \setminus \Lambda$ than for β in Λ, then the error will be small too. This will be called the dominant trail assumption. This hypothesis is false in general, but nevertheless useful. However, if $|\cdot|$ is an ultrametric absolute value, then the dominant trail assumption is actually a theorem. This will be important in Chapter 5.

2.4.4 Group and monoid actions

By performing a change of basis, it is theoretically possible to simplify the matrix-representation of the pushforward and pullback operators. However, the actual choice of basis was not discussed in Sections 2.4.1 to 2.4.3. Ideally, many of the matrices B^{F_i} should be diagonal, so that the propagation of basis vectors simplifies to scalar multiplication. From the point of view of Theorem 2.6, this also keeps the number of trails with a nonzero correlation small.

In general, it is not possible to find bases that simultaneously diagonalize all or even most of the pushforward operators T^{F_i}. Hence, this section is limited to specific classes of functions. In particular, only functions coming from a group or monoid action are considered.

Let M be a monoid acting on a set X. The action of m in M on x in X will be denoted by $m \cdot x$. For every m in M, one can define a function $\mathsf{F}_m(x) = m \cdot x$. Using Definition 2.7, this extends the action of M on X to an action on k^X by setting $m \cdot f = T^{\mathsf{F}_m} f$ for all f in k^X. With some abuse of notation, the notation $T^m = T^{\mathsf{F}_m}$ will be adopted. These extended actions are closely related to representation theory. For completeness, the definition of a representation is recalled below.

Definition 2.12 (Representation)**.** Let k be a field. A representation of a monoid M is a homomorphism $M \to \mathsf{End}(V)$, where V is a k-vector space. Equivalently, a representation is a k-vector space V together with an M-action on V.

Definition 2.13 (Subrepresentation)**.** Let V be a representation of a monoid M. A subrepresentation of V is a subspace U of V that is left-invariant under

the M-action. That is, $m \cdot U \subseteq U$ for all m in M. A representation V is called irreducible if its only subrepresentations are $\{0\}$ and V.

In our case, $V = k^X$ and the homomorphism is given by $m \mapsto T^m$. A central question in representation theory is whether or not a representation V can be decomposed as a direct sum $\bigoplus_{i=1}^{n} V_i$ of irreducible subrepresentations $V_1, \ldots, V_n$. Concretely, does there exist a basis for k^X such that the matrices B^m are all block-diagonal *with blocks of minimal sizes* $\dim V_i$, $i = 1, \ldots, n$?

An important result in representation theory says that this is possible whenever M is an inverse monoid, up to some constraints on the characteristic of k. This is a monoid M such that for every x in M, there exists a unique y in M with $xyx = x$.

If all the irreducible representations have dimension one, *i.e.* $\dim V_i = 1$ for $i = 1, \ldots, n$, then the block-diagonalization reduces to a complete diagonalization. It turns out that this is possible if M is commutative and inverse.

A particularly important case is that of a monoid M acting on itself by multiplication, *i.e.* $X = M$ and $m \cdot x = mx$. By Definition 2.12, the irreducible representations of a commutative inverse monoid M are homomorphisms $M \to k$. These functions are called characters.

Definition 2.14. A character of a commutative inverse monoid M is a homomorphism of monoids $\chi : M \to k$. That is, $\chi(1) = 1$ and $\chi(xy) = \chi(x)\chi(y)$ for all x and y in M.

Theorem 2.7. *The characters of a finite commutative inverse monoid M form a finite commutative inverse monoid under pointwise multiplication. This monoid is called the dual monoid and denoted by $\widehat{M}$.*

The term dual monoid is due to the canonical isomorphism $M \to \widehat{\widehat{M}} : x \mapsto \mathsf{ev}_x$ with $\mathsf{ev}_x(\chi) = \chi(x)$. For groups and with $k = \mathbb{C}$, the duality between M and $\widehat{M}$ is known as Pontryagin duality and M and $\widehat{M}$ are isomorphic.

Let b_χ with χ in $\widehat{M}$ be the basis vectors corresponding to the irreducible representations $V_\chi = \mathrm{Span}\{b_\chi\}$ in the decomposition $k^M = \bigoplus_{\chi \in \widehat{M}} V_\chi$. By definition, the basis functions b_χ satisfy

$$T^m b_\chi = \chi(m)\, b_\chi \,.$$

In particular the diagonal of B^m relative to this basis is given by $B^{\mathsf{F}}_{\chi,\chi} = \chi(m)$.

In fact, the dual basis of $\{b_\chi \mid \chi \in \widehat{M}\}$ can be constructed explicitly from the characters of M. The construction is given in Theorem 2.8.

Theorem 2.8. *For each character χ of M, let $b^\chi = \sum_{x \in M} \chi(x)\, \delta^x$. The vectors b^χ form a basis for $(k^M)^\vee$ such that for all m in M, the vector b^χ is an eigenvector of $T^{m^\vee}$ with eigenvalue $\chi(m)$. Furthermore, $b^\chi(b_\psi) = \delta_\chi(\psi)$ for all characters χ and ψ.*

Proof. The function b^χ is an eigenvector of $T^{m^\vee}$ since for all y in M,

$$\big(T^{m^\vee}\, b^\chi\big)(\delta_y) = \sum_{x \in M} \chi(x)\delta^x(\delta_{my}) = \chi(my) = \chi(m)\, b^\chi(\delta_y)\,.$$

Combining the equalities $T^{m^\vee}\, b^\chi = \chi(m) b^\chi$ and $T^m b_\psi = \psi(m) b_\psi$ yields

$$\chi(m) b^\chi(b_\psi) = \big(T^{m^\vee}\, b^\chi\big)(b_\psi) = b^\chi\big(T^m b_\psi\big) = \psi(m) b^\chi(b_\psi)\,.$$

This implies that $\{b_\chi \mid \chi \in \widehat{M}\}$ and $\{b^\chi \mid \chi \in \widehat{M}\}$ are dual bases. That is, $b^\chi(b_\psi) = \delta_\chi(\psi)$. □

If χ in $\widehat{M}$ is an invertible element, then one can verify that $b_\chi = \chi^{-1}/|M|$. Explicit formulas for b_χ when χ is not invertible are more complicated and are discussed in Section 5.3.

2.5 Multidimensional theory

In the previous section, one-dimensional cryptanalytic properties where discussed. This section extends these results to the general case. In order to do this, basis-free definitions of approximations and trails are introduced.

2.5.1 Approximations

Let $\mathsf{F} : X \to Y$ be a function and let U be the input space of some cryptanalytic property. The idea of an approximation is to project the vectors $T^{\mathsf{F}}\, u$ with u in U on another space V. In practice, the space V will be low-dimensional. The projection should be done in such a way that evaluations of the cryptanalytic property are preserved. In other words, from the point of view of the property, no approximation errors are made.

For the next definitions, we introduce the following notation. Let U be a subspace of k^X. The inclusion map on U is the map $\iota_U : U \to k^X$ defined by $\iota_U(x) = x$. Likewise, for a subspace V of $(k^X)^\vee$, it holds that $\iota_V(x) = x$. Finally, a projection on a subspace W of k^X or $(k^X)^\vee$ is a linear map π_W onto W such that $\pi_W^2 = \pi_W$. A projection is uniquely determined by its kernel.

Definition 2.15 (Forward approximation)**.** A forward approximation of a function $\mathsf{F} : X \to Y$ is a pair of subspaces (U, V) of k^X and k^Y respectively, together with an algebraic complement V^{c} of V. The approximation map of (U, V) is the linear operator $\langle V, U\rangle_{\mathsf{F}} = \pi_V \, T^{\mathsf{F}} \, \iota_U : U \to V$, with π_V the projection on V with kernel V^{c}.

Although the choice of the complement V^{c} is an essential part of the approximation, it is often convenient to refer to approximations by the pair (U, V) alone. In such cases, the complement will be clear from the context. For example, the notation $\langle V, U\rangle_{\mathsf{F}}$ does not include V^{c} although the map depends on it. This does not lead to confusion because, in this thesis, at most one complement V^{c} will be considered for any given V.

Dually, one can define backward approximations using $T^{\mathsf{F}^\vee}$. The idea is similar, but the definition starts from the output space of the property.

Definition 2.16 (Backward approximation)**.** A backward approximation of a function $\mathsf{F} : X \to Y$ is a pair of subspaces (V, U) of $(k^Y)^\vee$ and $(k^X)^\vee$ respectively, together with an algebraic complement U^{c} of U. The approximation map of (V, U) is the linear operator $\langle U, V\rangle_{\mathsf{F}} = \pi_U \, T^{\mathsf{F}^\vee} \, \iota_V : V \to U$, with π_U the projection on U with kernel U^{c}.

A forward approximation (U, V) with complement V^{c} preserves evaluations of the cryptanalytic property $(U, (V^{\mathsf{c}})^0)$. That is, for all u in U and v in $(V^{\mathsf{c}})^0$,

$$v\big(\langle V, U\rangle_{\mathsf{F}} \, u\big) = (v \circ \pi_V)\big(T^{\mathsf{F}} u\big) = v\big(T^{\mathsf{F}} u\big)\,,$$

where the second equality is due to $v \circ \pi_V = v$ for v in $(V^{\mathsf{c}})^0$. Hence, the approximation map can be used to evaluate the cryptanalytic property $(U, (V^{\mathsf{c}})^0)$. Conversely, if one can evaluate the property, then the approximation map can be computed. Similarly, a backward approximation (V, U) with complement U^{c} is related to the cryptanalytic property $((U^{\mathsf{c}})^0, V)$. The following result is a simple consequence of these relations.

Theorem 2.9. *Let (U, V) be a forward approximation of $\mathsf{F} : X \to Y$ with complement V^{c}. The principal correlation of $(U, (V^{\mathsf{c}})^0)$ is equal to $\|\langle V, U\rangle_{\mathsf{F}}\|_{op}$, with $\|\cdot\|_{op}$ the operator norm induced by the norms on k^X and k^Y.*

Proof. The operator norm $\|\langle V, U\rangle_{\mathsf{F}}\|_{\mathsf{op}}$ satisfies

$$\|\langle V, U\rangle_{\mathsf{F}}\|_{\mathsf{op}} = \sup_{\substack{u \in U \\ \|u\| \le 1}} \|\langle V, U\rangle_{\mathsf{F}} \, u\| = \sup_{\substack{u \in U \\ \|u\| \le 1}} \|\mathsf{ev}_{\langle V, U\rangle_{\mathsf{F}} \, u}\|^{\vee\vee}\,,$$

where $\mathsf{ev}_x : (k^Y)^\vee \to k$ is the evaluation map at x. By definition, the double-dual norm of $\mathsf{ev}_{\langle V,U\rangle_\mathsf{F}\, u}$ is equal to

$$\left\|\mathsf{ev}_{\langle V,U\rangle_\mathsf{F}\, u}\right\|^{\vee\vee} = \sup_{\substack{v\in V^\vee \\ \|v\|^\vee \le 1}} \left|\mathsf{ev}_{\langle V,U\rangle_\mathsf{F}\, u}(v)\right| = \sup_{\substack{v\in (V^\mathsf{c})^0 \\ \|v\|^\vee \le 1}} \left|v(T^\mathsf{F} u)\right|,$$

where the second equality follows from the fact that $\pi_V^\vee : V^\vee \to (k^Y)^\vee$ has range $(V^\mathsf{c})^0$. The result follows by taking the supremum with respect to u. □

Example 2.12. Let $\mathcal{X} = \{x_\alpha \mid \alpha \in A\}$ and $\mathcal{Y} = \{y_\beta \mid \beta \in B\}$ be bases for k^X and k^Y respectively, and choose x_α in $\mathcal{X}$ and y_β in $\mathcal{Y}$. If $U = \operatorname{Span}\{x_\alpha\}$ and $V = \operatorname{Span}\{y_\beta\}$ with complement $V^\mathsf{c} = \operatorname{Span}\mathcal{Y} \setminus \{y_\beta\}$, then the approximation map of (U, V) satisfies

$$\langle V, U\rangle_\mathsf{F}(\lambda x_\alpha) = \lambda\, y^\beta(T^\mathsf{F} x_\alpha) = \lambda\, B^\mathsf{F}_{\beta,\alpha},$$

where y^β is the dual basis vector of y_β and B^F is relative to $\mathcal{X}$ and $\mathcal{Y}$. ▷

2.5.2 Trails

Throughout this section, fix functions $\mathsf{F} : X_1 \to X_{r+1}$ and $\mathsf{F}_i : X_i \to X_{i+1}$ for $i = 1, \dots, r$ such that $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$. As in the one-dimensional case, trails provide a method to estimate the map of an approximation of F by gluing together the maps of approximations of the functions F_i. Equivalently, due to the link between approximations and properties discussed in Section 2.5.1, the evaluation of a property for F is estimated by evaluating sequences of properties for the functions F_i.

Definition 2.17 (Forward trail)**.** A forward trail for F is a sequence $(U_1, U_2, \dots, U_{r+1})$ such that (U_i, U_{i+1}) is a forward approximation of F_i.

Like approximations, trails can be defined either in the forward or backward direction.

Definition 2.18 (Backward trail)**.** A backward trail for F is a sequence $(U_{r+1}, \dots, U_2, U_1)$ such that (U_{i+1}, U_i) is a backward approximation of F_i.

Informally, the following result shows that the map of an approximation (U_1, U_{r+1}) of F is equal to the sum of the maps of all trails between U_1 and U_{r+1}. The map of a trail is the composition of the maps of its constituent approximations. If all vector spaces are one-dimensional, then this result is equivalent to Theorem 2.6.

Theorem 2.10 (Sum of forward trails)**.** *For $i = 1, \ldots, r+1$, let Ω_i be a set of subspaces of k^{X_i} such that $k^{X_i} = \bigoplus_{U \in \Omega_i} U$. Fix the complement of U_i in Ω_i for any approximation as $U_i^{\mathsf{c}} = \bigoplus_{U \in \Omega_i \setminus \{U_i\}} U$. For every forward approximation (U_1, U_{r+1}) of F with U_1 in Ω_1 and U_{r+1} in Ω_{r+1},*

$$\langle U_{r+1}, U_1 \rangle_{\mathsf{F}} = \sum_{U_2, \ldots, U_r} \langle U_{r+1}, U_r \rangle_{\mathsf{F}_r} \cdots \langle U_3, U_2 \rangle_{\mathsf{F}_2} \langle U_2, U_1 \rangle_{\mathsf{F}_1} ,$$

where the sum is over all $(U_2, \ldots, U_r)$ in $\prod_{i=2}^{r} \Omega_i$.

Proof. By Definition 2.15, $\langle U_{r+1}, U_i \rangle_{\mathsf{F}_r \circ \cdots \circ \mathsf{F}_i} = \pi_{U_{r+1}} T^{\mathsf{F}_r \circ \cdots \circ \mathsf{F}_i} \iota_{U_i}$. Furthermore, by the definition of Ω_{i+1}, the map $\sum_{U \in \Omega_{i+1}} \pi_U$ is the identity. Hence,

$$\langle U_{r+1}, U_i \rangle_{\mathsf{F}_r \circ \cdots \circ \mathsf{F}_i} = \sum_{U_{i+1} \in \Omega_{i+1}} \langle U_{r+1}, U_{i+1} \rangle_{\mathsf{F}_r \circ \cdots \circ \mathsf{F}_{i+1}} \langle U_{i+1}, U_i \rangle_{\mathsf{F}_i} .$$

The result follows by repeatedly applying this equality for $i = 1, \ldots, r-1$. □

Theorem 2.10 can equivalently be formulated in terms of backward approximations and trails. The proof is analogous.

Theorem 2.11 (Sum of backward trails)**.** *For $i = 1, \ldots, r+1$, let Ω_i be a set of subspaces of $(k^{X_i})^\vee$ such that $(k^{X_i})^\vee = \bigoplus_{U \in \Omega_i} U$. Fix the complement of U_i in Ω_i for any approximation as $U_i^{\mathsf{c}} = \bigoplus_{U \in \Omega_i \setminus \{U_i\}} U$. For every backward approximation (U_{r+1}, U_1) of F with U_1 in Ω_1 and U_{r+1} in Ω_{r+1},*

$$\langle U_1, U_{r+1} \rangle_{\mathsf{F}} = \sum_{U_2, \ldots, U_r} \langle U_1, U_2 \rangle_{\mathsf{F}_1} \langle U_2, U_3 \rangle_{\mathsf{F}_2} \cdots \langle U_r, U_{r+1} \rangle_{\mathsf{F}_r} ,$$

where the sum is over all $(U_2, \ldots, U_r)$ in $\prod_{i=2}^{r} \Omega_i$.

Theorems 2.10 and 2.11 are usually not directly useable in practice because the number of trails is either too large, or because some of the trails involve approximations that are too complicated. Instead, the following corollary is used. A similar result holds for backward trails.

Corollary 2.2 (Dominant trail approximation)**.** *For $i = 1, \ldots, r+1$, let Ω_i be a set of subspaces of k^{X_i} such that $k^{X_i} = \bigoplus_{U \in \Omega_i} U$. Fix the complement of U_i in Ω_i for any approximation as $U_i^{\mathsf{c}} = \bigoplus_{U \in \Omega_i \setminus \{U_i\}} U$.*

For every approximation (U_1, U_{r+1}) of F with U_1 in Ω_1 and U_{r+1} in Ω_{r+1}, let Ω be the set of all forward trails $(U_1, U_2, \ldots, U_{r+1})$ between U_1 and U_{r+1} with

U_i in Ω_i for $i = 2, \ldots, r$. For all subsets Λ of Ω,

$$\left\| \langle U_{r+1}, U_1 \rangle_{\mathsf{F}} - \sum_{U \in \Lambda} \langle U_{r+1}, U_r \rangle_{\mathsf{F}_r} \cdots \langle U_3, U_2 \rangle_{\mathsf{F}_2} \langle U_2, U_1 \rangle_{\mathsf{F}_1} \right\|$$

$$\leq \left\| \sum_{U \in \Omega \setminus \Lambda} \langle U_{r+1}, U_r \rangle_{\mathsf{F}_r} \cdots \langle U_2, U_1 \rangle_{\mathsf{F}_1} \right\|.$$

Proof. The result follows from Theorem 2.10. □

Corollary 2.2 is neither surprising nor particularly difficult. Nevertheless, its applications are extensive. It enables the analysis of linear, differential and integral properties (Corollary 2.1), and their higher-dimensional generalizations.

2.5.3 Perfect and zero-correlation approximations

Two special cases of Definitions 2.15 and 2.16 are important enough to deserve a separate discussion. The first of these are perfect approximations.

Definition 2.19 (Perfect approximation)**.** Let $\mathsf{F} : X \to Y$. A forward approximation (U, V) of F is perfect if and only if $T^{\mathsf{F}} U \subseteq V$. A backward approximation (V, U) of F is perfect if and only if $T^{\mathsf{F}^\vee} V \subseteq U$.

For trails consisting only of perfect approximations, the sum in Theorem 2.10 contains a single term. Hence, such a trail yields the exact approximation map. Iterative perfect approximations are called invariants.

Definition 2.20 (Invariant)**.** Let $\mathsf{F} : X \to X$. A forward invariant of F is a subspace V of k^X such that (V, V) is a perfect forward approximation of F. A backward invariant of F is a subspace V of $(k^X)^\vee$ such that (V, V) is a perfect backward approximation.

The following result is not surprising, but it is worth noting because it will be useful in Chapters 3 and 6. An analogous result holds for backward invariants.

Theorem 2.12. *Let V be a forward invariant of a permutation $\mathsf{F} : X \to X$. If the field k is algebraically closed and of characteristic zero, then V has a basis consisting of eigenvectors of T^{F}.*

Proof. The map T^{F} is diagonalizable over any algebraically closed field of characteristic zero. Indeed, since F^n is the identity function for some $n \geq 1$, the

minimal polynomial of T^{F} divides $x^n - 1$. This polynomial has distinct roots over an algebraically closed field of characteristic not dividing n.

If V is an invariant and F a permutation, then $T^{\mathsf{F}}V = V$. It follows that the minimal polynomial of the restriction $T^{\mathsf{F}}|_V : V \to V$ divides the minimal polynomial of T^{F}. Hence, $T^{\mathsf{F}}|_V$ is diagonalizable. □

Recall from Section 2.3.3 that a zero-correlation property (U, V) has principal correlation equal to zero. Equivalently, $T^{\mathsf{F}}U \subseteq V^0$.

Definition 2.21 (Zero-correlation approximation)**.** Let $\mathsf{F} : X \to Y$. A forward approximation (U, V) of F with complement V^{c} is zero-correlation if and only if $T^{\mathsf{F}}U \subseteq V^{\mathsf{c}}$. A backward approximation (V, U) of F with complement U^{c} is zero-correlation if and only if $T^{\mathsf{F}^\vee}V \subseteq U^{\mathsf{c}}$.

A zero-correlation approximation (U, V) satisfies $\langle V, U\rangle_{\mathsf{F}} = 0$. In general, zero-correlation approximations can be found by showing that all trails in the expansion in Theorem 2.10 or Theorem 2.11 contain an approximation with a trivial map. One can often use a miss-in-the-middle approach to simplify this process. Let $\mathsf{F} = \mathsf{F}_2 \circ \mathsf{F}_1$ be a function, (U, W_1) a forward approximation of F_1 and (V, W_2) a backward approximation of F_2. The miss-in-the middle principle states that if $W_1 \subseteq W_2^0$ or equivalently $W_1^0 \supseteq W_2$, then (U, V) is a zero-correlation property.

Finally, perfect and zero-correlation approximations are related as follows.

Theorem 2.13. *A forward approximation (U, V) with complement V^{c} is zero-correlation if and only if (U, V^{c}) is perfect with complement V. A backward approximation (V, U) with complement U^{c} is zero-correlation if and only if (V, U^{c}) is perfect with complement U.*

Proof. Consider the forward case, the backward case is similar. A forward approximation (U, V) of F is zero-correlation if and only if $T^{\mathsf{F}}U \subseteq V^{\mathsf{c}}$. That is, if and only (U, V^{c}) is perfect. □

The statement and proof of Theorem 2.13 are deceptively simple, but the result generalizes non-trivial connections between established cryptanalytic techniques. This will be discussed in Chapter 3.

2.6 Specializing the theory

In the next three chapters, the theory developed above will be specialized by instantiating three parameters. The first parameter is the relation between the function $\mathsf{F} : X \to Y$ and the cryptographic primitive under analysis. A second aspect is the choice of the field k, its absolute value function, and the metric structure of k^X. Finally, in each case a monoid action on X (and Y) will be specified, leading to a preferred choice of basis as discussed in Section 2.4.4.

It turns out that appropriate choices of these parameters yield the core techniques of modern cryptanalysis: linear cryptanalysis (Chapter 3), differential cryptanalysis (Chapter 4) and integral cryptanalysis (Chapter 5). In fact, in each case, a rich generalization of existing techniques is obtained. An overview is shown in Table 2.1.

Table 2.1: Three applications of the geometric approach to cryptanalysis.

	Linear Chapter 3	Differential Chapter 4	Integral Chapter 5
X	Group G, $+$ Commutative	Group $G \oplus G$, $+$ Commutative	Monoid M, $\cdot$ Commutative inverse
F	Primitive	Primitive for pairs	Primitive
k	$\mathbb{C}$	$\mathbb{C}$	$\mathbb{C}_p$
$\lvert\cdot\rvert$	$\lvert a+bi\rvert = \sqrt{a^2+b^2}$	$\lvert a+bi\rvert = \sqrt{a^2+b^2}$	$\lvert\alpha\rvert = \lvert\alpha\rvert_p$
Norm	Euclidean	Euclidean	$\lVert f\rVert = \max_{x\in X} \lvert f(x)\rvert$
Action	Translation $x \mapsto x - t$	Translation $(x, y) \mapsto (x - t, y - t)$	Coordinate scaling $x \mapsto t \cdot x$
Basis	Fourier	Quasidifferential†	Ultrametric†
Matrix	Correlation matrix C^{F}	Quasidifferential transition matrix D^{F}	Ultrametric transition matrix A^{F}

† These terms will be defined in Chapters 5 and 8.

2.6.1 Linear cryptanalysis

The case of linear cryptanalysis is the most straightforward. The function F is the same as the primitive, so that X and Y correspond to the input and output space of the primitive. Typically, $X = \mathbb{F}_2^n$ and $Y = \mathbb{F}_2^m$, but with an eye to the applications in Part II, it is useful to develop the theory for arbitrary finite

commutative groups. The field k is chosen as the complex numbers, with the modulus function as the absolute value. Mathematically, this naturally leads one to consider the Euclidean norm on k^X. As will be discussed in Chapter 3, this choice also has a cryptanalytic motivation.

The one-dimensional theory from Section 2.4 leads to linear trails and approximations. These concepts follow automatically by choosing the basis that diagonalizes the translation action $x \mapsto x - t$ of group elements t. Performing the corresponding change-of-basis on T^{F} gives the correlation matrix C^{F}. These matrices were introduced from a different point of view by Daemen *et al.* [101] shortly after the discovery of linear cryptanalysis.

Since the one-dimensional theory of linearly cryptanalysis is relatively well-understood by means of correlation matrices, Chapter 3 focuses on extensions such as multiple linear cryptanalysis, invariants and nonlinear approximations. A full description of these variants of linear cryptanalysis relies on the basis-free theory from Section 2.5.

2.6.2 Differential cryptanalysis

Differential cryptanalysis does not deal with properties of individual inputs, but of input pairs. In particular, the input and output spaces are of the form $X = G \oplus G$ and $Y = H \oplus H$ with with G and H finite commutative groups – typically vector spaces over $\mathbb{F}_2$. The function F is then equal to $\mathsf{F}((x, y)) = (\mathsf{G}(x), \mathsf{G}(y))$, with G the actual primitive. As in linear cryptanalysis, $k = \mathbb{C}$ and k^X is equipped with the Euclidean norm.

Contrary to linear cryptanalysis, even the one-dimensional theory is new in the case of differential cryptanalysis. In Chapter 4, a suitable basis that diagonalizes the translation action $(x, y) \mapsto (x-t, y-t)$ is introduced. This leads to the notion of quasidifferential trails. The dominant theory of differential cryptanalysis is obtained when all results are averaged over independent and uniform random round keys.

Consequently, Chapter 4 primarily develops the one-dimensional theory of quasidifferential trails.

2.6.3 Integral cryptanalysis

Integral cryptanalysis considers parts of the ciphertext that are saturated or sum to zero. Evaluating a zero-sum property implies that one solves a combinatorial problem modulo two, *i.e.* one should expect to use the theory from this chapter

with $k = \mathbb{F}_2$. This turns out to be possible, and even quite successful: it naturally leads to the parity set description of the division property by Canteaut and Boura [79], and to division trails.

Since $\mathbb{F}_2$ comes with the discrete topology, some of the concepts introduced in this chapter become trivial or less natural – for example, the correlation of a trail is either zero or one. For this reason, Chapter 5 proposes a broad extension of integral cryptanalysis that makes the theory more complete and highlights the similarity with linear and differential cryptanalysis. This theory is constructed by choosing k to be an algebraically closed extension of the field of p-adic numbers $\mathbb{Q}_p$. As explained in Chapter 5, there is also a cryptanalytic motivation for the proposed extension.

As in linear cryptanalysis, the function F is the primitive itself with X and Y the input and output space respectively. It will be assumed that X and Y are commutative inverse monoids. Typical examples are the monoids $\mathbb{F}_q^n$ with coordinate-wise multiplication. The monoid action $x \mapsto t \cdot x$ obtained by multiplying with a constant leads to a preferred basis. The space k^X is equipped with a natural ultrametric norm. This non-Archimidean metric structure is an essential difference with both linear and differential cryptanalysis.

Chapter 5 focuses on the one-dimensional case. The existing theories of division trails and parity sets can be understood as approximations of the 2-adic theory for $X = \mathbb{F}_2^n$ and $Y = \mathbb{F}_2^m$, by dropping all trails with absolute correlation below $1/2$.

3

Linear cryptanalysis

This chapter applies the geometric approach from Chapter 2 to linear cryptanalysis. As the one-dimensional theory of linear trails was already known prior to this work, the multidimensional theory is the main focus of this chapter. Most of the results in this chapter are obtained by translating previously proposed extensions of linear cryptanalysis into the geometric framework, and subsequently exploring the implications.

Large parts of this chapter are based on the paper "A geometric approach to linear cryptanalysis" [40] from Asiacrypt 2021. The first ideas for this paper came from my master's thesis "Linear cryptanalysis in the weak-key model" [38], which was supervised by Vincent Rijmen. I also thank Gregor Leander and Christof Beierle for interesting discussions about this work at Ruhr-University Bochum. In addition to the results of [40], this chapter includes several unpublished results.

3.1 Introduction

Linear approximations over multiple rounds of a cipher are typically obtained by combining several one-round approximations. Matsui [215] initially accomplished this by assuming the independence of the one-round approximations, so that the 'piling-up lemma' could be applied. Shortly after, the theoretical advances of Nyberg [224] and Daemen *et al.* [101] led to additional insight into this heuristic approach. In Sections 3.2 and 3.3 of this chapter, it is shown that the one-dimensional theory from Section 2.4 reproduces the correlation matrix approach of Daemen. As the choice of basis is determined by the group action corresponding to key or constant additions, the importance of linear cryptanalysis is not surprising from this viewpoint.

The success of linear cryptanalysis has led to the development of a myriad of extensions and variants of linear approximations. Kaliski and Robshaw [173] suggested using multiple linear approximations. Hermelin, Cho and Nyberg [162] proposed the related multidimensional linear attack. Both extensions are widely used. Generalizations of linear cryptanalysis to groups other than $\mathbb{F}_2^n$ were

proposed by Granboulan, Levieil and Piret [148] and Baignères, Stern and Vaudenay [17]. The use of nonlinear approximations is another natural extension, and has been attempted by Knudsen and Robshaw [183], Harpes, Kramer and Massey [159] with I/O sums, Harpes and Massey [160] with partitioning attacks and by Beierle, Canteaut and Leander [26]. Section 3.4 of this chapter serves as a dictionary between all of these cryptanalytic properties and their representation as pairs of vector spaces following Definition 2.6. An overview is shown in the third column of Table 3.1.

The analysis of cryptanalytic properties for iterated primitives depends on forward and backward approximations. Although approximations and cryptanalytic properties are different in general, there is a one-to-one correspondence between them in the case of linear cryptanalysis and throughout this chapter. Section 3.5 extends Sections 2.5.1 and 2.5.3 with results about approximations that are specific to this context. For example *principal correlations* are introduced in Definition 2.9 as a natural multidimensional extension of the correlation of a linear approximation.

Table 3.1: Cryptanalytic properties (U, V) for a function F with U and V vector spaces of dimension d.

	Zero-correlation $C^{\mathsf{F}}U \subseteq V^0$	Thm. 3.4 ⇄ Perfect $C^{\mathsf{F}}U \subseteq V$	Section 3.6.2 → General $\langle V, U\rangle_{\mathsf{F}}$
$d = 1$	Linear zero-correlation [73] Nonlinear zero-correlation §3.8	Invariant subspaces [196] Nonlinear invariants [266] Eigenvectors of C^{F} [37]	Linear cryptanalysis [215] Abelian groups [17] I/O sums [159] Beierle *et al.* [26] Rank-one (Section 3.7)
$d \geq 1$	Multidim. zero-correlation [72]	Saturation attacks [184] General invariants (Definition 2.20, §3.5.1)	Multiple linear [63, 173] Multidim. linear [162] Partitioning [160] Projection, χ^2 [16, 274, 280]

Several lightweight block ciphers have been found vulnerable to weak key attacks based on invariant subspaces [196] and nonlinear invariants [266]. These attacks have led to renewed interest in linear cryptanalysis and its generalizations. Indeed, nonlinear invariants provide one of the most compelling examples of nonlinearity in cryptanalysis, with applications including the analysis of SCREAM, iSCREAM, Midori and MANTIS [37,266]. Theorem 2.12 in Section 2.5.3 leads to describing invariant subspaces and nonlinear invariants as eigenvectors

of pushforward operators. Applications of this approach to the cryptanalysis of Midori-64 and MANTIS are discussed in Chapter 6.

In a different direction, Bogdanov and Rijmen [73] introduced zero-correlation linear cryptanalysis to exploit unbiased linear approximations. Zero-correlation linear properties are examples of Definition 2.21. Several extensions are listed in the first column of Table 3.1. At Asiacrypt 2012, Bogdanov, Leander, Nyberg and Wang [72] established a link between multidimensional linear zero-correlation approximations and integral distinguishers with the saturation property [184]. Corollary 3.4 in Section 3.5.2, a straightforward consequence of Theorem 2.13, generalizes this result.

Section 3.5.3 shows how the principal correlations of approximations relate to the data complexity of optimal distinguishers. This extends earlier results by Baignères, Junod and Vaudenay [16].

All of the variants of linear cryptanalysis in the third column of Table 3.1 rely on heuristic methods to glue together several approximations over multiple rounds of a cipher. These methods will be collectively referred to as the *piling-up principle*. This principle has traditionally been justified using independence or Markov chain assumptions [16, 280], similar to the initial approach in ordinary linear cryptanalysis. However, such assumptions are hard to reconcile with the key-dependence of approximations and the increased importance of cryptographic permutations. In fact, key-dependence is one of the fundamental difficulties of nonlinear cryptanalysis. Alternatively, the correlation matrix framework of Daemen *et al.* [101] is more suitable for the fixed-key setting. However, it only applies to ordinary linear cryptanalysis. In Section 3.6, it is shown that a general piling-up principle can be deduced from the dominant trail approximation from Corollary 2.1, thereby avoiding the independence heuristic.

Abdelraheem, Ågren, Beelen and Leander [1] found links between invariant subspaces and linear cryptanalysis. Beierle, Canteaut and Leander [26] extended these links to some classes of nonlinear invariants. In Section 3.6.2, the characterization of invariants as eigenvectors of correlation matrices is combined with the general piling-up principle to simplify and extend these results.

The aforementioned results established a strong link between nonlinear invariants and linear cryptanalysis, but a true statistical generalization of the nonlinear invariant attack was left open in previous work. Section 3.7 introduces rank-one approximations to analyze cell-oriented ciphers. A tool to find optimal rank-one trails is introduced, and its application to searching for invariants is discussed. Perhaps surprisingly, the tool is based on numerical optimization on a Riemannian manifold. This is enabled by introducing new types of approximations, resulting in a smooth search space. Rank-one approximations

are used in Section 3.8 to resolve a problem introduced by Beierle *et al.* [26], which is representative of other concrete problems.

3.2 Mathematical setting

Let $\mathsf{F} : G \to H$ be a cryptographic primitive with G and H finite commutative groups. This could be generalized to arbitrary sets G and H together with suitable group actions, but this case will not be considered in this thesis. The properties considered in linear cryptanalysis involve counting over the integers or more generally the rational numbers. Distances are measured using the ordinary absolute value function $|\cdot|$. Since $\mathbb{Q}$ is not complete with respect to $|\cdot|$, it is mathematically more convenient to work over the field of real numbers or its algebraic closure $\mathbb{C}$.

It was already mentioned in Section 2.6.1 that the vector spaces $\mathbb{C}^G$ and $\mathbb{C}^H$ can be equipped with the Euclidean norm $\|\cdot\|_2$. In fact, to be precise, a scaled variant of this norm is used in this chapter. Section 3.2.1 reviews the theory of inner products, which are closely related to the Euclidean norm. Section 3.2.2 motivates the choice of the Euclidean norm.

Finally, Section 3.2.3 discusses the choice of the group actions on $\mathbb{C}^G$ and $\mathbb{C}^H$.

3.2.1 Inner product spaces

This section reviews the theory of inner products. Since no new results are presented, readers who are familiar with linear algebra in inner product spaces may skip this section. Additional information can be found in standard references such as Halmos' textbook on finite-dimensional vector spaces [157].

Definition 3.1 (Inner product space)**.** Let V be a vector space over $\mathbb{C}$. An inner product on V is a function $V \times V \to \mathbb{C}$, denoted by $\langle \cdot, \cdot \rangle$, such that

(1) For all x, y and z in V and λ and μ in $\mathbb{C}$, $\langle x, \lambda y + \mu z \rangle = \lambda \langle x, y \rangle + \mu \langle x, z \rangle$.

(2) It is antisymmetric: $\overline{\langle x, y \rangle} = \langle y, x \rangle$ for all x and y in V.

(3) For all x in V, $\langle x, x \rangle \geq 0$ with equality if and only if $x = 0$.

A vector space with an inner product is called an inner product space.

Example 3.1. The map $(x, y) \mapsto \sum_{i=1}^{n} \overline{x_i} y_i$ is an inner product on $\mathbb{C}^n$. ▷

The next result shows that every inner product space is normed. Throughout this chapter, norms on inner product spaces are always defined as in Theorem 3.1.

Theorem 3.1. *If V is a vector space with inner product $\langle \cdot, \cdot \rangle$, then $x \mapsto \|x\| = \sqrt{\langle x, x \rangle}$ is a norm on V.*

The norm on an inner product space satisfies the Cauchy-Schwarz inequality.

Theorem 3.2 (Cauchy-Schwarz inequality)**.** *Let V be a vector space with inner product $\langle \cdot, \cdot \rangle$ and corresponding norm $\| \cdot \|$. For all x and y in V, it holds that $|\langle x, y \rangle| \leq \|x\| \|y\|$ with equality for $y = x/\|x\|$.*

Inner products are closely related to dual spaces. Specifically, as shown by Theorem 3.3, every inner product defines a one-to-one correspondence between a vector space and its dual. Recall that an anti-isomorphism of vector spaces over $\mathbb{C}$ is an invertible map f such that $f(\lambda x + \mu y) = \overline{\lambda} f(x) + \overline{\mu} f(y)$ for all vectors x and y and scalars λ and μ. A map is called isometric if it preserves norms. Theorem 3.3 can be deduced from the Cauchy-Schwarz inequality (Theorem 3.2).

Theorem 3.3. *Let V be a finite-dimensional vector space with inner product $\langle \cdot, \cdot \rangle$. For any x in V, define x^* in $V^\vee$ by $x^*(y) = \langle x, y \rangle$ for all y in V. The map $x \mapsto x^*$ is an isometric anti-isomorphism.*

It was mentioned in Example 2.3 that the Euclidean norm $\| \cdot \|_2$ has the remarkable property that it is self-dual. Using Theorem 3.3, this can be explained by the fact that $\|x\|_2 = \sqrt{\langle x, x \rangle}$ with $\langle x, y \rangle = \sum_{i=1}^n \overline{x_i} y_i$ the standard inner product on $\mathbb{C}^n$.

Recall that a linear map $L : U \to V$ between finite-dimensional vector spaces U and V has an adjoint $L^\vee : V^\vee \to U^\vee$ defined by $L^\vee f = f \circ L$ for all f in $V^\vee$. If $\langle \cdot, \cdot \rangle_U$ and $\langle \cdot, \cdot \rangle_V$ are inner products on U and V respectively, then by Theorem 3.3 there exists a linear map $L^\dagger : V \to U$ such that $(L^\dagger v)^* = L^\vee v^*$ for all v in V. In terms of inner products, $L^\dagger$ satisfies

$$\langle L^\dagger v, u \rangle_U = \langle v, Lu \rangle_V \,,$$

for all u in U and v in V. The map $L^\dagger$ is also called the adjoint of L.

Example 3.2. Let $\mathsf{F} : X \to Y$ be a function between sets X and Y. If $\mathbb{C}^X$ and $\mathbb{C}^Y$ are inner product spaces, then the adjoint of T^{F} is the linear map represented by the transpose of the matrix T^{F}. That is, up to the anti-isomorphism from Theorem 3.3, $T^{\mathsf{F}^\dagger}$ is equivalent to the pullback operator $T^{\mathsf{F}^\vee}$. ▷

Inner products come with a geometric interpretation. Two vectors whose inner product is zero are said to be orthogonal. A basis consisting of mutually

orthogonal vectors with norm one is called an orthonormal basis. The orthogonal complement of a subspace V of an inner product space W is the vector space $V^\perp$ of all vectors orthogonal to V:

$$V^\perp = \left\{ w \in W \mid \langle v, w \rangle = 0 \text{ for all } v \text{ in } V \right\}.$$

Orthogonal complements are also algebraic complements. That is, $W = V \oplus V^\perp$. Hence, one can define a projection $\pi_V : W \to V$ with kernel $V^\perp$. For any w in W, $\pi_V(w)$ is the *orthogonal projection* of w on V.

More generally, the modulus of the inner product between two normalized vectors can be interpreted as the cosine of the smallest angle enclosed by them – although for non-real vectors, several definitions of angles are plausible.

The concept of angles between vectors can be generalized to subspaces of an inner product space W. For this purpose, it is convenient to extend the inner product notation $\langle \cdot, \cdot \rangle$ to subspaces. For subspaces U and V, define the linear map $\langle V, U \rangle : U \to V$ by $\langle V, U \rangle = \pi_V \, \iota_U$, where $\iota_U : U \to W$ is the inclusion map and $\pi_V : W \to V$ is the orthogonal projection on V. Note that $\langle V, U \rangle = \langle U, V \rangle^\dagger$ since projection and inclusion are adjoint.

Example 3.3. Let U and V be one-dimensional subspaces of W spanned by unit-norm vectors u and v respectively. By definition, $\iota_U(\lambda u) = \lambda u$ and $\pi_V(x) = v\langle v, x \rangle$. Consequently, $\langle V, U \rangle : U \to V$ is the map $\lambda u \mapsto \langle v, u \rangle \lambda v$. The matrix representation of this map is thus simply the 1×1 matrix containing the inner product $\langle v, u \rangle$. ▷

The transformation $\langle V, U \rangle$ comes with a geometric interpretation, which will be important in Sections 3.5 and 3.6. Due to standard properties of orthogonal projection, $\langle V, U \rangle$ maps any u in U to the nearest vector v in V. In addition, no other vector in V of the same length makes a smaller angle to u than v. This suggests that $\langle V, U \rangle$ encodes all information about the 'angles' between U and V. This claim can be made precise using the notion of principal angles between subspaces, which is due to Jordan [171]. The characterization below follows Björck and Golub [66].

Definition 3.2 (Principal angles)**.** Let U and V be finite-dimensional subspaces of a vector space with inner product $\langle \cdot, \cdot \rangle$ and corresponding norm $\| \cdot \|$, and let $d = \min\{\dim U, \dim V\}$. The principal angles $0 \leq \theta_1 \leq \ldots \leq \theta_d \leq \pi/2$ between U and V are recursively defined by (for $i = 1, 2, \ldots, d$)

$$\cos \theta_i = \frac{\langle u_i, v_i \rangle}{\|u_i\| \, \|v_i\|} = \max_{\substack{u \in U_i \setminus \{0\} \\ v \in V_i \setminus \{0\}}} \frac{|\langle u, v \rangle|}{\|u\| \, \|v\|},$$

where u_i in U_i and v_i in V_i are nonzero vectors for which the maximum in the right-hand side is achieved with $\langle u_i, v_i \rangle$ a non-negative real number, $U_i = U \cap \mathrm{Span}\{u_1, \ldots, u_{i-1}\}^\perp$ and $V_i = V \cap \mathrm{Span}\{v_1, \ldots, v_{i-1}\}^\perp$.

The cosines of the principal angles are precisely the singular values of $\langle V, U \rangle$, and the singular vectors are the directions along which these angles are to be measured. This follows directly from the variational characterization of singular values. Further details may be found in [66]. For completeness, the spectral characterization of singular vectors is given in Definition 3.3. Note that this definition relies on the observation that $L^\dagger L$ is positive semi-definite and self-adjoint. Hence, by the spectral theorem, its eigenvalues are non-negative real numbers.

Definition 3.3 (Singular value decomposition)**.** Let $L : U \to V$ be a linear map between inner product spaces U and V and let $\sigma_1^2 \geq \ldots \geq \sigma_{\dim U}^2$ be the eigenvalues of $L^\dagger L$. A singular value decomposition of L consists of orthonormal bases $\{u_1, \ldots, u_{\dim U}\}$ and $\{v_1, \ldots, v_{\dim V}\}$ for U and V respectively, such that for all x in U

$$L(x) = \sum_{i=1}^{d} \sigma_i \langle u_i, x \rangle \, v_i \,,$$

with $\langle \cdot, \cdot \rangle$ the inner product on U and $d = \min\{\dim U, \dim V\}$. The values $\sigma_1, \ldots, \sigma_d$ are called the singular values of L and the vectors $v_1, \ldots, v_d$ and $u_1, \ldots, u_d$ are called left and right singular vectors respectively.

The Frobenius norm of a linear map $L : U \to V$ with singular values $\sigma_1, \ldots, \sigma_d$ is defined by

$$\|L\|_{\mathsf{fr}} = \sqrt{\sum_{i=1}^{d} \sigma_i^2} \,.$$

Due to Definition 3.3, the squared Frobenius norm also equals the sum of the squared absolute values of the coordinates of any matrix representing L relative to orthogonal bases for U and V. Furthermore, one can shown that $\|L\|_{\mathsf{fr}} = \sqrt{\langle L, L \rangle_{\mathsf{fr}}}$ where $\langle L, M \rangle_{\mathsf{fr}} = \mathrm{Tr}(L^\dagger M)$ is the Frobenius inner product between linear maps L and M.

3.2.2 Motivation for the Euclidean norm

Throughout this chapter, the following norm on $\mathbb{C}^G$ is used:

$$\|u\|_G = \sqrt{|G|} \, \|u\|_2 = \sqrt{|G| \sum_{x \in G} |u(x)|^2} \,.$$

The $\|\cdot\|_G$-norm is induced by the inner product $\langle v, u\rangle_G = |G| \sum_{x\in G} \overline{v(x)}u(x)$. Hence, by Theorem 3.3, the map $v \mapsto \langle v, \cdot\rangle_G$ is an isometric anti-isomorphism from $\mathbb{C}^G$ to $(\mathbb{C}^G)^\vee$.

The correspondence between $\mathbb{C}^G$ and $(\mathbb{C}^G)^\vee$ leads to a number of simplifications and brings out additional geometric aspects of the theory. In particular, forward and backward approximations (Definitions 2.15 and 2.16) are simplified by choosing orthogonal complementary spaces. This will be clarified in Section 3.5.

The existence of an inner product makes the (scaled) Euclidean norm a natural choice from a mathematical point of view. However, it can also be motivated on cryptanalytic grounds. More specifically, the norm $\|\cdot\|_G$ has a statistical motivation.

Recall from Definition 2.6 that the evaluation of a cryptanalytic property (U, V) for $\mathsf{F} : G \to H$ at u in U and $\langle v, \cdot\rangle_H$ in V is equal to

$$\langle v, T^{\mathsf{F}} u\rangle_H = |H| \sum_{x\in G} u(x)\overline{v(\mathsf{F}(x))}\,.$$

Let $(x_1, y_1), \ldots, (x_q, y_q)$ be q plaintext-ciphertext pairs in $G \times H$. An unbiased known-plaintext estimator of $\langle v, T^{\mathsf{F}} u\rangle_H$ is given by

$$t = \frac{|G|\,|H|}{q} \sum_{i=1}^{q} u(x_i)\overline{v(y_i)}\,.$$

Let $\boldsymbol{t}_{\mathsf{ideal}}$ be the random variable obtained from the estimator t when the plaintext-ciphertext pairs (x_i, y_i) are independent and uniform random on $H \times G$. The label 'ideal' refers to the fact that $\boldsymbol{t}_{\mathsf{ideal}}$ is often a good model for the estimator when the data is obtained from the ideal primitive[1].

If $\mathsf{E}\,\boldsymbol{t}_{\mathsf{ideal}} = 0$, then the variance of $\boldsymbol{t}_{\mathsf{ideal}}$ satisfies

$$\mathsf{Var}\,\boldsymbol{t}_{\mathsf{ideal}} = \frac{|G|^2\,|H|^2}{q}\mathsf{E}|u(\boldsymbol{x})|^2\,|v(\boldsymbol{y})|^2 = \|u\|_G^2\,\|v\|_H^2/q\,.$$

A fair comparison of the quality of different choices of u and v should keep the variance of $\boldsymbol{t}_{\mathsf{ideal}}$ constant. Indeed, Theorem 1.1 shows that the data complexity of many attacks depends on the variance of $\boldsymbol{t}_{\mathsf{ideal}}$. This is a strong motivation for the $\|\cdot\|_G$- and $\|\cdot\|_H$-norms.

However, the data complexity also depends on the variance of the test-statistic for the real primitive. For linear cryptanalysis and many of its variants, the real-case variance is close to the variance of $\boldsymbol{t}_{\mathsf{ideal}}$. Section 3.5.3 contains a more detailed discussion of the data complexity for cryptanalytic properties satisfying this condition.

[1] This is a simplification of reality even if the ideal primitive is a uniform random function.

3.2.3 Group action

As discussed in Section 2.4.4, the specification of a group action on $\mathbb{C}^G$ leads to a preferred basis. Since G is a group, any element t of G acts on G by $x \mapsto x - t$ and hence on $\mathbb{C}^G$ by

$$\big(T^t f\big)(x) = f(x + t)\,,$$

where the positive sign is due to the fact that T^t describes forward propagation.

The group action $x \mapsto x - t$ is a natural choice from a cryptanalytic viewpoint because most primitives involve both key and constant additions. Hence, it is useful to choose a basis that simplifies these operations as much as possible. This can be achieved by diagnalizing the operators T^t, which has the additional benefit of keeping the number of trails small. The resulting preferred basis will be obtained in Section 3.3.1, following the general principles from Section 2.4.4.

3.3 One-dimensional theory

In this section, the one-dimensional theory from Section 2.4 is applied to the setting that was described in Section 3.2. Specifically, Section 3.3.1 shows that the preferred basis resulting from the group action defined in Section 3.2.3 is the Fourier basis. Expressing pushforward operators in terms of the Fourier basis leads to correlation matrices. In Section 3.3.2, it is shown how the known properties of correlation matrices are immediate consequences of this fact. The resulting theory of one-dimensional approximations and trails is that of classical linear cryptanalysis (extended to arbitrary groups), and is briefly reviewed in Section 3.3.3.

3.3.1 Fourier basis

Following Section 2.4.4, the matrices T^t corresponding to the action of t in G are simultaneously diagonalized in the basis of group characters. All characters in this chapter are assumed to be complex-valued. That is, they are group homomorphisms from G to $\mathbb{C}^\times$.

Recall from Theorem 2.7 that the characters of G form a commutative group $\widehat{G}$ under pointwise multiplication. The group $\widehat{G}$ is the Pontryagin dual of G.

Example 3.4. The dual of the additive group $\mathbb{F}_2$ is $\widehat{\mathbb{F}}_2 = \{x \mapsto 1, x \mapsto (-1)^x\}$. Indeed, these are the only two group homomorphisms from $\mathbb{F}_2$ to $\mathbb{C}^\times$. ▷

A few standard properties of the dual group are given in Theorem 3.4 below. The third property also holds for monoids, but the first two are specific to groups. Property (2) shows that the basis of characters is orthogonal.

Theorem 3.4 (Properties of dual groups [262])**.** *If G is a finite commutative group with dual $\widehat{G}$, then:*

(1) The dual group $\widehat{G}$ is isomorphic to G.

(2) For all χ and ψ in $\widehat{G}$, it holds that $\langle \chi/|G|, \psi/|G| \rangle_G = \delta_\chi(\psi)$.

(3) If $G = G_1 \oplus G_2$ with $\oplus$ the internal direct sum, then $\widehat{G} = \widehat{G}_1 \oplus \widehat{G}_2$.

By Theorem 3.4 (1), $\widehat{G}$ can be identified with G. In general, this identification is not unique. However, as discussed in Section 2.4.4, there is a *functorial* isomorphism between the double dual of G and G itself, which identifies g in G with the evaluation map $\chi \mapsto \chi(g)$ in the dual of $\widehat{G}$. In order to avoid arbitrary choices, isomorphisms between G and G will be avoided throughout Part I. This makes no difference in specific calculations, but it is theoretically more elegant.

Example 3.5. Since the additive group $\mathbb{F}_2^n$ is the direct sum of n copies of $\mathbb{F}_2$, it follows from Theorem 3.4 (3) that the dual group is essentially the direct sum of n copies of $\widehat{\mathbb{F}}_2$. Specifically, $\widehat{\mathbb{F}}_2^n = \{x \mapsto \prod_{i=1}^n (-1)^{u_i x_i} = (-1)^{u^\intercal x} \mid u \in \mathbb{F}_2^n\}$. Note that identifying $\widehat{\mathbb{F}}_2^n$ and $\mathbb{F}_2^n$ requires choosing a basis for $\mathbb{F}_2^n$. ▷

Similar to the annihilator of a subspace, the annihilator of a subgroup is defined as follows. Despite the similarities to Definition 2.2, one should keep in mind that group characters are homomorphisms to the multiplicative group $\mathbb{C}^\times$, whereas linear functionals are homomorphisms to the additive group $\mathbb{C}$.

Definition 3.4 (Annihilator)**.** Let G be a finite commutative group. The annihilator of a subset H of G is the subgroup

$$H^1 = \left\{ \chi \in \widehat{G} \mid \forall x \in H : \chi(x) = 1 \right\}.$$

Taking the annihilator is an antitone Galois connection between the lattices of subgroups of G and $\widehat{G}$. Furthermore, it holds that $|H^1| = |G|/|H|$.

By Definition 2.10, the change-of-basis transformation $\mathscr{F}_G : \mathbb{C}^G \to \mathbb{C}^{\widehat{G}}$ to the basis $\{\chi/|G| \mid \chi \in \widehat{G}\}$ is defined by $\mathscr{F}_G \, \chi/|G| = \delta_\chi$ for all characters χ. The factor $1/|G|$ is due to the fact that the correct basis actually consists of the functions $\chi/|G|$, as explained in the last paragraph of Section 2.4.4. The transformation $\mathscr{F}_G$ is also known as the Fourier transformation on G. Due to the orthogonality of group characters, *i.e.* Theorem 3.4 (2), the following definition is equivalent.

Definition 3.5 (Fourier transformation [262])**.** The Fourier transformation is the map $\mathscr{F}_G : \mathbb{C}^G \to \mathbb{C}^{\widehat{G}}$ defined by

$$\left(\mathscr{F}_G f\right)(\chi) = \langle \chi, f \rangle = \sum_{x \in G} \overline{\chi(x)} f(x)\,,$$

for all χ in $\widehat{G}$.

It is worth mentioning that $\mathscr{F}_G$ is an isomorphism of algebras which swaps the pointwise product and convolution. This is by construction, since the set of convolution operators is generated by translations.

The vector space $\mathbb{C}^{\widehat{G}}$ can be equipped with the standard inner product

$$\langle f, g \rangle = \sum_{\chi \in \widehat{G}} \overline{f(\chi)}\, g(\chi)\,.$$

Due to the orthogonality of characters, the inner product between u and v in $\mathbb{C}^G$ coincides with the above inner product of their Fourier transformations:

$$\left\langle \mathscr{F}_G\, u, \mathscr{F}_G\, v \right\rangle = \sum_{\chi \in \widehat{G}} \sum_{x,y \in G} \chi(x-y)\, \overline{u(x)} v(y) = \langle u, v \rangle_G\,.$$

In other words, $\mathscr{F}_G$ is a unitary map relative to the inner products $\langle \cdot, \cdot \rangle_G$ and $\langle \cdot, \cdot \rangle$. That is, $\mathscr{F}_G^{-1} = \mathscr{F}_G^\dagger$ with $\mathscr{F}_G^\dagger$ the adjoint of $\mathscr{F}_G$.

To end this section, consider the case $G = \bigoplus_{i=1}^n G_i$. As mentioned above, one has $\mathbb{C}^G = \bigotimes_{i=1}^n \mathbb{C}^{G_i}$ up to canonical isomorphism. By Theorem 3.4 (3), the dual group satisfies $\widehat{G} = \bigoplus_{i=1}^n \widehat{G}_i$. Hence, one also has $\mathbb{C}^{\widehat{G}} = \bigotimes_{i=1}^n \mathbb{C}^{\widehat{G}_i}$. Consequently, the Fourier transformation on $\mathscr{F}_G$ is given by $\bigotimes_{i=1}^n \mathscr{F}_{G_i}$. Equivalently, the matrix representation of $\mathscr{F}_G$ in the standard basis is the Kronecker product of the matrix representations of $\mathscr{F}_{G_1}, \ldots, \mathscr{F}_{G_n}$ in the standard basis.

3.3.2 Correlation matrices

This section discusses the implications of expressing the pushforward operator of a function $\mathsf{F} : G \to H$ relative to the Fourier bases of $\mathbb{C}^G$ and $\mathbb{C}^H$, following Definition 2.11. The matrix representation of the resulting operator will be called the *correlation matrix* of F.

Correlation matrices (for $G = \mathbb{F}_2^n$ and $H = \mathbb{F}_2^m$) were introduced by Daemen *et al.* [101] shortly after the discovery of linear cryptanalysis. They provide a natural description of linear cryptanalysis.

Definition 3.6 (Correlation matrix)**.** Let $\mathsf{F} : G \to H$ be a function between finite commutative groups G and H. Define $C^{\mathsf{F}} : \mathbb{C}^{\widehat{G}} \to \mathbb{C}^{\widehat{H}}$ as the pushforward operator of F relative to the Fourier basis. That is, $C^{\mathsf{F}} = \mathscr{F}_H T^{\mathsf{F}} \mathscr{F}_G^{-1}$, with $\mathscr{F}_H$ and $\mathscr{F}_G$ the Fourier transformation on $\mathbb{C}^H$ and $\mathbb{C}^G$ respectively.

The correlation matrix of F is the matrix representation of C^{F} with respect to the standard bases of $\mathbb{C}^{\widehat{G}}$ and $\mathbb{C}^{\widehat{H}}$.

As usual, the notation C^{F} refers to both the linear operator and its standard matrix representation. The coordinates of C^{F} are given by

$$C^{\mathsf{F}}_{\chi,\psi} = \langle \delta_\chi, C^{\mathsf{F}} \delta_\psi \rangle = \langle \chi/|H|, T^{\mathsf{F}} \psi/|G| \rangle_H = \frac{1}{|G|} \sum_{x \in G} \overline{\chi(\mathsf{F}(x))} \psi(x).$$

For $G = \mathbb{F}_2^n$ and $H = \mathbb{F}_2^m$, and after identifying these groups with their dual, the expression above coincides with the original definition of correlation matrices by Daemen.

The following two theorems list the main properties of correlation matrices that will be used throughout this thesis. Corollary 3.1 is an immediate consequence of Theorem 2.5.

Corollary 3.1 (Properties of correlation matrices)**.** *The correlation matrix C^{F} of $\mathsf{F} : G \to H$ has the following properties:*

(1) If $\mathsf{F}(x_1, \ldots, x_n) = (\mathsf{F}_1(x_1), \ldots, \mathsf{F}_n(x_n))$, then $C^{\mathsf{F}} = \bigotimes_{i=1}^n C^{\mathsf{F}_i}$.

(2) If $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$, then $C^{\mathsf{F}} = C^{\mathsf{F}_r} \cdots C^{\mathsf{F}_2} C^{\mathsf{F}_1}$.

Theorem 3.5 (Properties of correlation matrices)**.** *The correlation matrix C^{F} of $\mathsf{F} : G \to H$ has the following properties:*

(1) If F is a bijection, then C^{F} is a unitary matrix.

(2) If F is a group homomorphism, then $C^{\mathsf{F}}_{\chi,\psi} = \delta_{\chi \circ \mathsf{F}}(\psi)$.

(3) If $G = H$ and $\mathsf{F}(x) = x - t$ for some constant t in G, then C^{F} is a diagonal matrix with $C^{\mathsf{F}}_{\chi,\chi} = \chi(t)$.

Proof. As discussed in Chapter 2, if F is a permutation, then T^{F} is a permutation matrix and thus unitary. Furthermore $\mathscr{F}_G^{-1}$ and $\mathscr{F}_H$ are unitary with respect to appropriate inner products. Property (1) follows since the composition of unitary maps is unitary and $C^{\mathsf{F}} = \mathscr{F}_H T^{\mathsf{F}} \mathscr{F}_G^{-1}$.

For (2), note that if F is a group homomorphism, then so is $\chi \circ \mathsf{F} : G \to \mathbb{C}^\times$. Hence, by the orthogonality of group characters, $C^{\mathsf{F}}_{\chi,\psi} = \delta_{\chi \circ \mathsf{F}}(\psi)$. As discussed

in Section 3.3.1, property (3) holds by construction of the Fourier transformation. Indeed, note that $T^{\mathsf{F}} = T^t$. □

3.3.3 Approximations and trails

An ordinary linear approximation corresponds to a property (U, V) with $U = \mathrm{Span}\{\chi\}$ and $V = \mathrm{Span}\{\langle\psi, \cdot\rangle_H\}$ for χ in $\widehat{G}$ and ψ in $\widehat{H}$. Following the general principles from Section 2.4.1, the evaluation of this property at $\chi/|G|$ and $\langle\psi/|H|, \cdot\rangle_H$ is equal to $C^{\mathsf{F}}_{\psi,\chi}$.

Corollary 3.2 (Sum of linear trails, *cf.* Theorem 2.6)**.** *If* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$*, then the correlation* $C^{\mathsf{F}}_{\chi_{r+1},\chi_1}$ *is equal to*

$$C^{\mathsf{F}}_{\chi_{r+1},\chi_1} = \sum_{\chi_2,\ldots,\chi_r} \prod_{i=1}^{r} C^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} ,$$

where the sum ranges over all intermediate group characters.

Corollary 3.2 is the main theoretical result of ordinary linear cryptanalysis. It states that the correlation of a linear approximation is equal to the sum of the correlations of all trails within the approximation. For $G = \mathbb{F}_2^n$ and $H = \mathbb{F}_2^m$, Corollary 3.2 was first obtained by Daemen [101, §6.1]. Truncating the sum to a subset of trails leads to the principle of dominant trails, *i.e.* Corollary 2.1 for the Fourier basis.

Corollary 3.3 (Dominant trail approximation *cf.* Corollary 2.1)**.** *Let* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$*. For all subsets* Λ *of the set* Ω *of all trails from* χ_1 *to* χ_{r+1}*,*

$$\left| C^{\mathsf{F}}_{\chi_{r+1},\chi_1} - \sum_{\chi\in\Lambda} \prod_{i=1}^{r} C^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} \right| \leq \left| \sum_{\chi\in\Omega\setminus\Lambda} \prod_{i=1}^{r} C^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} \right| ,$$

with $\chi = (\chi_2, \ldots, \chi_r)$*.*

Tardy-Corfdir and Gilbert [261] and Matsui [215] implicitly relied on Corollary 3.3 with Λ a singleton. That is, early work in linear cryptanalysis relied on a single trail $(\chi_1, \ldots, \chi_{r+1})$ with the largest absolute correlation and the estimate

$$C^{\mathsf{F}}_{\chi_{r+1},\chi_1} \approx \prod_{i=1}^{r} C^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} .$$

This approximation is known as the piling-up heuristic.

At Eurocrypt 1994, Nyberg [224] presented a result similar to but weaker than Corollary 3.2 for $G = \mathbb{F}_2^n$ and $H = \mathbb{F}_2^m$. If $\mathsf{F}_k = \mathsf{R}_r \circ \cdots \circ \mathsf{R}_1$ is a key-dependent function with k in G^r and $\mathsf{R}_i(x) = \mathsf{F}_i(x) + k_i$ for $i = 1, \ldots, r$, then

$$\mathsf{E}_{\boldsymbol{k}} |C^{\mathsf{F}_{\boldsymbol{k}}}_{\chi_{r+1},\chi_1}|^2 = \sum_{\chi_2,\ldots,\chi_r} \prod_{i=1}^{r} |C^{\mathsf{F}_i}_{\chi_{i+1},\chi_i}|^2 , \tag{3.1}$$

where the round keys $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_r$ are independent and uniform random. This can be shown by substituting $C^{\mathsf{R}_i}_{\chi,\psi} = \chi(k) C^{\mathsf{F}_i}_{\chi,\psi}$ into Corollary 3.2 and using the fact that the variance of uncorrelated random variables is additive. For general finite commutative groups, (3.1) was first obtained by Baignères, Stern and Vaudenay [17].

It is important to mention that, in general, (3.1) is *not* a reliable way to estimate the squared correlation of a linear approximation. In many cases, the fixed-key squared correlation differs significantly from the average squared correlation. Hence, indiscriminate usage of the average as a substitute can lead to incorrect results.

3.4 Cryptanalytic properties

As the one-dimensional theory of linear cryptanalysis is relatively well-understood, the remainder of this chapter focuses on the multidimensional theory.

The one-dimensional theory from Section 3.3 describes classical linear cryptanalysis and its extension to arbitrary groups, but it does not apply to many important variants such as multiple and multidimensional linear cryptanalysis, χ^2-distinguishers, invariant subspaces and nonlinear invariants, ...

The discussion below is structured as a dictionary between conventional descriptions of cryptanalytic properties and the corresponding pairs of subspaces. Once these subspaces are known, the general machinery from Section 2.5 can be applied. As a side-effect, this point of view often clarifies the relations between different properties.

A short summary for $G = \mathbb{F}_2^n$ is given in Table 3.2. The table includes both the subspaces of $\mathbb{C}^G$ and their Fourier transforms, which are subspaces of $\mathbb{C}^{\widehat{G}}$. Importantly, there are other useful subspaces which do not correspond to any of the constructions discussed below. One example will be discussed in Section 3.7.

Table 3.2: Commonly used cryptanalytic properties and their corresponding subspaces. The characters of $\mathbb{F}_2^n$ are denoted by $\chi_u(x) = (-1)^{u^\mathsf{T} x}$, where $u \in \mathbb{F}_2^n$.

Property	Basis for subspace $V \subseteq \mathbb{C}^{\mathbb{F}_2^n}$	Basis for subspace $\mathscr{F}_{\mathbb{F}_2^n} V \subseteq \mathbb{C}^{\widehat{\mathbb{F}_2^n}}$	Applications
Affine space $a + U \subseteq \mathbb{F}_2^n$	$\{\mathbb{1}_{a+U}\}$	$\{\chi_a \, \mathbb{1}_{U^\perp}\}$	Invariant subspaces
Affine spaces $a_1 + U_1, \ldots \subseteq \mathbb{F}_2^n$	$\{\mathbb{1}_{a_1+U_1}, \ldots\}$	$\{\chi_{a_1} \, \mathbb{1}_{U_1^\perp}, \ldots\}$	Saturation attack
Probability dist. $p : \mathbb{F}_2^n \to [0,1]$	$\{p\}$	$\{\mathscr{F}_{\mathbb{F}_2^n} p\}$	Statistical saturation
Linear *Mask* $u \in \mathbb{F}_2^n$	$\{\chi_u\}$	$\{\delta_{\chi_u}\}$	Linear cryptanalysis
Multidim. linear *Subspace* $U \subseteq \mathbb{F}_2^n$	$\{\chi_u \mid u \in U\}$	$\{\delta_{\chi_u} \mid u \in U\}$	Multidimensional linear cryptanalysis
Multiple linear *Subset* $U \subseteq \mathbb{F}_2^n$	$\{\chi_u \mid u \in U\}$	$\{\delta_{\chi_u} \mid u \in U\}$	Multiple linear cryptanalysis
Nonlinear *Fun.* $\mathsf{P} : \mathbb{F}_2^n \to \mathbb{F}_2$	$\{(-1)^\mathsf{P}\}$	$\{\mathscr{F}_{\mathbb{F}_2^n}[(-1)^\mathsf{P}]\}$	Nonlinear invariants I/O sums
Projection *Fun.* $\mathsf{P} : \mathbb{F}_2^n \to X$	$\{\delta_x \circ \mathsf{P} \mid x \in X\}$	$\{\mathscr{F}_{\mathbb{F}_2^n}(\delta_x \circ \mathsf{P}) \mid x \in X\}$	Partitioning attacks χ^2 distinguishers

3.4.1 Indicator functions

Several cryptanalytic properties correspond to subspaces spanned by one or more indicator functions. The general principle behind such properties was already briefly discussed in Example 2.5.

A first example are properties spanned by indicators of affine subspaces of $\mathbb{F}_2^n$, such as the invariant subspace attack of Leander *et al.* [196]. Specifically, the property is of the form $(U, U^*) = (\mathrm{Span}\{\mathbb{1}_A\}, \mathrm{Span}\{\langle \mathbb{1}_A, \cdot \rangle_{\mathbb{F}_2^n}\})$ with A an affine subspace. The approximation (U, U) is an invariant in the sense of Definition 2.20. This generalizes to arbitrary groups.

Integral properties of the 'saturated' type provide another example. A saturated property expresses that the marginal distribution of a part of the output is uniform. In this case, the corresponding vector spaces are spanned by the indicator functions of all sets which are saturated on certain bits. They are typically higher-dimensional as they express several possible sets in which the

state could be contained. The actual sets, especially at the output side, tend to be defined implicitly. Example 3.10 on page 80 works out the vector spaces in detail.

Not many variants of linear cryptanalysis are directly based on non-uniform probability distributions. The statistical saturation attack of Collard and Standaert [94], in its original form, can be considered an example. In this attack, one estimates the key-dependent probability distribution of the state of a block cipher when some of the plaintext bits are constant and the others are uniform random. However, depending on how the estimated distribution is used, it may be more appropriate to approach this attack using the projection functions discussed below.

3.4.2 Projection functions

Let $\mathsf{P} : G \to Z$ be a function between a finite commutative group G and a finite set Z, with Z typically much smaller than G. Such functions play an important role in Wagner's framework of 'commutative diagram cryptanalysis', where they are called *projections* [280]. Baignères *et al.* [16] analyze the statistical properties of distinguishers based on balanced projections, such as χ^2-attacks [274], partitioning cryptanalysis [160] and multidimensional linear attacks [162].

Properties defined by projection functions form an important subclass of those defined by indicator functions. Although any property described by projection functions can also be described by indicator functions, many properties are more naturally described using the former approach.

Viewed as a subspace of $(\mathbb{C}^G)^\vee$, a projection function gives access to the evaluation of P on the state. Equivalently, it allows observing the inner product with any linear combination of the functions $\delta_z \circ \mathsf{P}$, where $\{\delta_z \mid z \in Z\}$ is the standard basis of $\mathbb{C}^Z$. More generally, any function on Z can be 'pulled back' to G along the projection function P. This leads to Definition 3.7 below.

Definition 3.7 (Pullback space)**.** Let $\mathsf{P} : G \to Z$ be a function. The pullback space (of $\mathbb{C}^Z$ to $\mathbb{C}^G$) along P is the vector space defined by

$$\operatorname{im} T^{\mathsf{P}^\dagger} = \left\{ f \circ \mathsf{P} \mid f \in \mathbb{C}^Z \right\}.$$

Similarly, the Fourier transformation $\mathscr{F}_G \operatorname{im} T^{\mathsf{P}^\dagger}$ will be called the pullback (of $\mathbb{C}^Z$ to $\mathbb{C}^{\widehat{G}}$) along P. If Z is a commutative group, then $\mathscr{F}_G \operatorname{im} T^{\mathsf{P}^\dagger} = \operatorname{im} C^{\mathsf{P}^\dagger}$.

Let V_P be the vector space corresponding to the projection property defined by P, *i.e.* the pullback along P. It was already mentioned above that $\{\delta_z \circ \mathsf{P} \mid z \in Z\}$

is a basis for V_{P}. However, if Z is a commutative group, then it is often more convenient to use the basis of functions $\chi \circ \mathsf{P}$ where $\chi \in \widehat{Z}$. This choice behaves particularly well for homomorphisms $\mathsf{P} : G \to Z$ when working with the Fourier transformation of V_{P}, since $\mathscr{F}_G(\chi \circ \mathsf{P})/|G| = \delta_{\chi \circ \mathsf{P}}$.

The following example describes the vector space corresponding to a Boolean projection function in more detail. Such properties are closely related to classical linear cryptanalysis, and more generally to the I/O-sums of Harpes *et al.* [159] and the nonlinear approximations considered by Beierle *et al.* [26]. However, as discussed in Section 3.4.3, there is a subtle difference.

Example 3.6. Let $\mathsf{P} : \mathbb{F}_2^n \to \mathbb{F}_2$ be a Boolean function. Denote the characters of $\mathbb{F}_2^n$ by $\chi_u(x) = (-1)^{u^{\mathsf{T}}x}$. The pullback space V_{P} along P is equal to

$$V_{\mathsf{P}} = \operatorname{Span}\{\delta_0 \circ \mathsf{P}, \delta_1 \circ \mathsf{P}\} = \operatorname{Span}\{\mathbb{1}, (-1)^{\mathsf{P}}\},$$

with $\mathbb{1} = \chi_0$ the trivial character of $\mathbb{F}_2^n$. Hence, the Fourier transformation of V_{P} is given by

$$\mathscr{F}_G\, V_{\mathsf{P}} = \operatorname{Span}\{\delta_{\mathbb{1}}, \mathscr{F}_G[(-1)^{\mathsf{P}}]\}.$$

The function $\mathscr{F}_G[(-1)^{\mathsf{P}}]$ is often called the Walsh-Hadamard transformation of P. If P is a linear function, then $\mathsf{P}(x) = u^{\mathsf{T}}x$ for some $u \in \mathbb{F}_2^n$. Hence, $(-1)^{\mathsf{P}} = \chi_u$ and consequently $\mathscr{F}_G\, V = \operatorname{Span}\{\delta_{\mathbb{1}}, \delta_{\chi_u}\}$. ▷

3.4.3 Subspaces of pullbacks

Example 3.6 generalizes to other finite commutative groups. Let Z be a finite commutative group and $\mathsf{P} : G \to Z$ a homomorphism. Since $\chi \circ \mathsf{P} \in \widehat{G}$ for any character χ of Z, the pullback along P is spanned by the functions $\chi \circ \mathsf{P}$ with χ in $\widehat{Z}$. Hence, $\dim V = |Z|$. However, the dimension could be reduced by one for permutations. This leads to the generalization of linear cryptanalysis proposed by Granboulan *et al.* [148, §3].

However, it is also reasonable to consider only one of the functions $\chi \circ \mathsf{P}$. This results in one-dimensional subspaces and is closer to the spirit of ordinary linear cryptanalysis. This leads to the generalization of ordinary linear cryptanalysis to other groups from Section 2.4, which was (partially) developed by Baignères *et al.* [17].

The difference between *multiple* and *multidimensional* linear cryptanalysis is of the same nature. For multiple linear properties, one uses subspaces spanned by one or more group characters. In multidimensional linear cryptanalysis, these characters form a subgroup and consequently the subspace is the pullback along a homomorphism to some subgroup. In particular, for this reason, the

statistical analysis of Baignères *et al.* [16] applies to multidimensional but not to multiple linear cryptanalysis.

3.5 Approximations

The inner-product structure of the vector spaces $\mathbb{C}^G$ and $\mathbb{C}^H$ leads to a simplification of the definition of forward and backward approximations. Indeed, by Definition 2.15, an approximation is a pair of subspaces (U, V) of $\mathbb{C}^G$ and $\mathbb{C}^H$ together with an algebraic complement V^{c} of V. In general, the choice of the complement V^{c} is arbitrary. However, in inner product spaces, the choice $V^{\mathsf{c}} = V^{\perp}$ is natural. Hence, one obtains the following simplified version of Definition 2.15.

Definition 3.8 (Approximation)**.** Let G and H be finite commutative groups. An approximation of a function $\mathsf{F} : G \to H$ is a pair (U, V) of subspaces U of $\mathbb{C}^G$ and V of $\mathbb{C}^H$.

The approximation map of (U, V) is the linear transformation $\langle V, U\rangle_{\mathsf{F}} : U \to V$ defined by $\langle V, U\rangle_{\mathsf{F}} = \pi_V \, T^{\mathsf{F}} \, \iota_U$, with ι_U the inclusion map and π_V the orthogonal projection on V.

Definition 3.8 refers to subspaces of $\mathbb{C}^G$ and $\mathbb{C}^H$. An equivalent definition could be given for subspaces of $\mathbb{C}^{\widehat{G}}$ and $\mathbb{C}^{\widehat{H}}$, taking into account that T^{F} should be replaced by C^{F} in the definition of the approximation map.

If (U, V) is an approximation in the sense of Definition 3.8, then (U, V) is a forward approximation with complementary space $V^{\mathsf{c}} = V^{\perp}$ in the sense of Definition 2.15. Due to Theorem 3.3, a separate definition for backward approximations is not necessary.

Throughout this chapter, the complementary space for all approximations will be chosen as the orthogonal complement, as in Definition 3.8. With this assumption, every approximation (U, V) uniquely corresponds to the cryptanalytic property (U, V^*) with V^* the subspace obtained by applying the anti-isomorphism from Theorem 3.3 to V. Indeed, recall that an approximation (U, V) with complement V^{c} preserves evaluations at u in U and v in $(V^{\mathsf{c}})^0$. The result then follows from $(V^{\perp})^0 = V^*$.

Given orthonormal bases $u_1, u_2, \ldots$ and $v_1, v_2, \ldots$ for U and V respectively, the coordinates of the matrix representing the approximation map are given by the inner products $\langle v_i, T^{\mathsf{F}} u_i\rangle$.

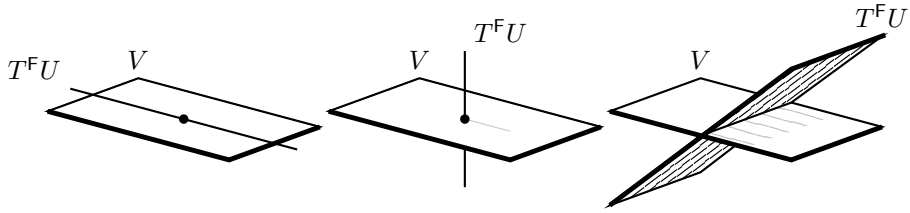
Example 3.7. Consider a linear approximation of a function $\mathsf{F} : G \to H$. As listed in Table 3.2, linear properties correspond to one-dimensional spaces

$U = \mathrm{Span}\{\psi\}$ and $V = \mathrm{Span}\{\chi\}$ with characters ψ in $\widehat{G}$ and ψ in $\widehat{H}$. The inclusion map is defined by $\iota_U(x) = x$ and the orthogonal projection by $\pi_V(x) = \langle \chi, x \rangle_H \, \chi/|H|^2$. Hence, $\langle V, U \rangle_{\mathsf{F}}$ is given by

$$\lambda \frac{\psi}{|G|} \mapsto \left\langle \chi/|H|, T^{\mathsf{F}}\psi/|G| \right\rangle_H \lambda \frac{\chi}{|H|} = C^{\mathsf{F}}_{\chi,\psi} \, \lambda \frac{\chi}{|H|} \,.$$

That is, in the Fourier basis, $\langle V, U \rangle_{\mathsf{F}}$ corresponds to multiplication by $C^{\mathsf{F}}_{\chi,\psi}$. ▷

As illustrated in Figure 3.1, two geometrically intuitive edge cases of Definition 3.8 can be identified: parallel or orthogonal spaces V and $T^{\mathsf{F}}U$. Approximations in the former category will be called 'perfect'. This includes the important case of invariants. The latter category is a broad generalization of zero-correlation linear approximations. In the remaining cases, the vector spaces V and $T^{\mathsf{F}}U$ are neither completely parallel nor fully orthogonal. All three cases are discussed in detail in Sections 3.5.1 to 3.5.3.



(a) Perfect. (b) Zero-correlation. (c) General case.

Figure 3.1: Geometric interpretation of Definition 2.15.

The following definition extends Definition 2.9 with additional principal correlations. Note that $\|\langle V, U \rangle_{\mathsf{F}}\|_{\mathsf{op}}$ is equal to the largest singular value of $\langle V, U \rangle_{\mathsf{F}}$, such that the largest principal correlation defined in Definition 3.9 indeed equals *the* principal correlation from Definition 2.9.

Definition 3.9 (Principal correlations)**.** Let (U, V) be an approximation of a function $\mathsf{F} : G \to H$ between finite commutative groups G and H. Let $d = \min\{\dim U, \dim V\}$. The principal correlations of the approximation (U, V) are the d largest singular values of the approximation map $\langle V, U \rangle_{\mathsf{F}}$.

The geometric interpretation of the principal correlations is clarified by the following result, which relates them to the principal angles between the subspaces $T^{\mathsf{F}}U$ and V.

Theorem 3.6. *Let (U, V) be an approximation of a function $\mathsf{F} : G \to H$ between finite commutative groups G and H. Let $d = \min\{\dim U, \dim V\}$. If* F

is balanced, then the principal correlations of (U, V) *are equal to the cosines of the* d *smallest principal angles between the subspaces* $T^{\mathsf{F}}U$ *and* V*.*

Proof. If F is balanced, then a direct calculation shows that $T^{\mathsf{F}^\dagger}T^{\mathsf{F}}$ is a nonzero multiple of the identity map. That is, T^{F} preserves the inner product up to multiplication by a constant.

If T^{F} preserves the inner product up to multiplication by a nonzero constant, then $\langle u_{i+1}, u_i\rangle_G = 0$ implies $\langle T^{\mathsf{F}}u_{i+1}, T^{\mathsf{F}}u_i\rangle_H = 0$. Hence, the result follows from the fact that the variational characterization of singular values is equivalent to the definition of principal angles (Definition 3.2). □

3.5.1 Perfect approximations and invariants

Following Definition 2.19 in Section 2.5.3, an approximation (U, V) of F is called perfect if and only if $T^{\mathsf{F}}U \subseteq V$. A perfect approximation of the form (V, V) was called an invariant in Definition 2.20.

Integral properties of the saturated type are perfect, but Chapter 5 provides a more complete approach to integral cryptanalysis. However, invariants are of particular interest as they include the invariant subspaces of Leander *et al.* [196] and the nonlinear invariants of Todo *et al.* [266].

By Theorem 2.12 and because $\mathbb{C}$ is algebraically closed, every invariant V has a basis consisting of eigenvectors of T^{F}. Equivalently, relative to the Fourier basis, any invariant is spanned by eigenvectors of C^{F}. This characterization of invariant subspaces and nonlinear invariants was the basis of the paper "Block cipher invariants as eigenvectors of correlation matrices" from Asiacrypt 2018 [37]. This point of view will be applied in Chapter 6.

Example 3.8 (Invariant subspaces)**.** If S is an invariant subset for F, then $T^{\mathsf{F}}\,\mathbb{1}_S = \mathbb{1}_S$ or equivalently $C^{\mathsf{F}}\,\widehat{\mathbb{1}}_S = \widehat{\mathbb{1}}_S$ with $\widehat{\mathbb{1}}_S = \mathscr{F}_G\mathbb{1}_S$. Invariant subspaces are obtained by taking S affine. ▷

Example 3.9 (Nonlinear invariants)**.** A nonlinear invariant for $\mathsf{F} : G \to G$ is a function $\mathsf{P} : G \to Z$ such that $\mathsf{P}(\mathsf{F}(x)) = \mathsf{P}(x) + c$ for some constant c in Z with Z a finite commutative group. Following Section 3.4 and Example 3.6 in particular, the pullback space V_{P} along P is equal to $V_{\mathsf{P}} = \mathrm{Span}\{\delta_z \circ \mathsf{P} \mid z \in Z\}$. The approximation $(V_{\mathsf{P}}, V_{\mathsf{P}})$ is an invariant for F since $T^{\mathsf{F}}\,V_{\mathsf{P}} = V_{\mathsf{P}}$.

As discussed above, any invariant is spanned by eigenvectors of T^{F}. A basis of eigenvectors is easy to obtain: since $\mathsf{P}\circ\mathsf{F}$ and P differ by a constant, it is natural to consider the basis $\{\chi\circ\mathsf{P} \mid \chi \in \widehat{Z}\}$ of V_{P}. It holds that $T^{\mathsf{F}}(\chi\circ\mathsf{P}) = \chi(c)\,(\chi\circ\mathsf{P})$. Equivalently, $\mathscr{F}_G\,(\chi \circ \mathsf{P})$ is an eigenvector of C^{F} with eigenvalue $\chi(c)$.

For $G = \mathbb{F}_2^n$ and $Z = \mathbb{F}_2$, the last result is equivalent to [37, Corollary 1], which states that the Walsh-Hadamard transform of a nonlinear invariant is an eigenvector of C^{F}. ▷

To compute the invariants of a permutation F, one could resort to standard numerical algorithms to compute the eigenvectors of T^{F} or C^{F}. This is not a particularly efficient approach: the computational cost is $\mathcal{O}(|G|^3)$, which is of the same order as the ANF-based algorithm proposed by Todo *et al.* [266] to find nonlinear invariants.

In fact, due to the structure of the matrices T^{F} and C^{F}, their eigendecomposition can be computed using at most $\widetilde{\mathcal{O}}(|G|^2)$ operations. The following algorithm generalizes the cycle structure approach that is mentioned by Todo *et al.* [266] as "potentially applicable". For each cycle $(x_0, \ldots, x_{l-1})$ of F, and for every $0 \leq k < l$, one obtains an eigenvector[2] $v = \sum_{i=0}^{l-1} \zeta^{-i}\, \delta_{x_i}$ with corresponding eigenvalue ζ, where $\zeta = e^{2\pi k\sqrt{-1}/l}$. Indeed,

$$T^{\mathsf{F}} v = \sum_{i=0}^{l-1} \zeta^{-i}\, T^{\mathsf{F}} \delta_{x_i} = \sum_{i=0}^{l-1} \zeta^{-(i-1)} \delta_{x_i} = \zeta v\,.$$

As the sum of all cycle lengths is $|G|$, this method yields a complete eigenvector basis. The time complexity is $\sum_i l_i^2 = \mathcal{O}(|G|^2)$, with l_i the length of the i^{th} cycle. The eigenvectors of C^{F} can be computed by taking $|G|$ Fourier transformations.

Even the improved algorithm above is impractical for most realistic state sizes. Furthermore, although this method outputs a complete eigenvector basis, one is usually interested in eigenvectors with some additional structure. For instance, as shown by the following result, the requirement that an invariant is preserved by many key-additions directly restricts its structure.

Theorem 3.7. *If v is an eigenvector of C^k for all k in a subset K of G, then $\operatorname{supp} v \subseteq \chi\, K^1$ for some χ in $\widehat{G}$. Furthermore, v is an eigenvector of C^k with corresponding eigenvalue $\chi(k)$ for all k in the subgroup generated by K.*

Proof. If v is an eigenvector of C^k with eigenvalue $\lambda(k)$ for all k in K, then $\lambda(k)\, v(\psi) = \psi(k) v(\psi)$ for all ψ in $\widehat{G}$ and all k in K. This implies that there exists a character χ of G such that $\lambda(k) = \chi(k)$ for all k in K. Indeed, it suffices to take any χ in the support of v. The identity $(\psi/\chi)(K) = 1$ for all ψ in $\operatorname{supp} v$ then implies that $\operatorname{supp} v \subseteq \chi\, K^1$. This proves the first part of the result.

For the second part, let k be an element of the subgroup $\langle K \rangle$ generated by K. By the above, every character in $\operatorname{supp} v$ is of the form $\chi\psi$ with ψ in K^1. Since

[2] It is not hard to see that it is linearly independent from previously computed eigenvectors.

$\langle K \rangle^1 = K^1$, it follows that

$$(\chi\psi)(k)\, v(\chi\psi) = \chi(k) v(\chi\psi)\,.$$

That is, $C^k v = \chi(k)\, v$. This proves the second part of the result. □

Examples of invariants will be given in Section 3.7 and Chapter 6.

3.5.2 Zero-correlation approximations

Zero-correlation linear approximations were introduced by Bogdanov and Rijmen [73]. They are linear approximations $(\mathrm{Span}\{\psi\}, \mathrm{Span}\{\chi\})$ such that $C^{\mathsf{F}}_{\chi,\psi} = 0$. That is, χ is orthogonal to $T^{\mathsf{F}}\psi$. This corresponds to the geometric situation sketched in Figure 3.1b, and is captured by Definition 2.21. In particular (U, V) is a zero-correlation approximation if and only if $V \perp T^{\mathsf{F}} U$.

Recall that Theorem 2.13 showed that zero-correlation and perfect approximations are closely related, despite being opposite extremes. For completeness, Corollary 3.4 restates this result for approximations with orthogonal complements. This result is also clear from a geometrical point of view, see for instance Figures 3.1a and 3.1b.

Corollary 3.4 (*cf.* Theorem 2.13)**.** *If (U, V) is a zero-correlation approximation, then $(U, V^\perp)$ is a perfect approximation and conversely.*

Corollary 3.4 is deceptively simple, but the result is powerful. Indeed, it generalizes the well-known correspondence between multidimensional linear zero-correlation approximations and saturation properties in integral cryptanalysis, first noted by Bogdanov *et al.* at Asiacrypt 2012 [72][3] and discussed further by Sun *et al.* [258].

Example 3.10. Let (U, V) be an integral property of saturation type for $\mathsf{F} : G \to H$. That is, $U = \mathrm{Span}\{\mathbb{1}_x \mid x \in G/A\}$ for a subgroup A of G. The output space V is typically implicitly defined by a projection homomorphism $\mathsf{P} : H \to B$ to a subgroup B of H. That is, $V = \mathrm{Span}\{v \in \mathbb{C}^H \mid \exists \lambda \in \mathbb{C} : T^{\mathsf{P}} v = \lambda \mathbb{1}\}$.

Since every χ in A^1 is constant on the cosets of the subgroup A, it holds that $U \supseteq \mathrm{Span}\, A^1$. In fact, comparing dimensions shows that this is an equality. Hence, $U = \mathrm{Span}\, A^1$. Furthermore, $V = \ker T^{\mathsf{P}} + \mathrm{Span}\{\mathbb{1}\}$ because $T^{\mathsf{P}}\mathbb{1} = |H|/|B|\,\mathbb{1}$. It is not difficult to see that $\ker T^{\mathsf{P}} = \mathrm{Span}\, \widehat{H} \setminus (\ker P)^1$. It follows that $V^\perp = \mathrm{Span}\, (\ker P)^1 \setminus \{\mathbb{1}\}$.

[3] For the case of multidimensional zero-correlation approximations with 'coupled masks', apply Corollary 3.4 to the function $x \mapsto (x, \mathsf{F}(x))$ to obtain their result.

By Corollary 3.4, (U, V) is a perfect approximation if and only if $(U, V^\perp)$ is a zero-correlation approximation. In particular, for all ψ in A^1 and $\chi \neq \mathbb{1}$ in $(\ker P)^1$, it holds that $C^{\mathsf{F}}_{\chi,\psi} = 0$. The converse also holds. This extends the main results of [72] to arbitrary commutative groups. ▷

3.5.3 General approximations

The behavior of most approximations is in-between the extremal cases of zero-correlation and perfect approximations. As described in Section 2.5.2, such approximations are typically obtained using the principle of dominant trails. This means that usually only an estimate of the approximation map is available. Section 3.6 discusses how such an estimate can be obtained using trails. This section focuses on a different issue, namely that of judging the quality of approximations. The main observation is that the principal correlations are central to this issue.

It follows from Example 3.7 that the unique principal correlation of an ordinary linear approximation equals the absolute value of its correlation. For fixed advantage, the data complexity of a linear distinguisher is inversely proportional to the square of the correlation. Hence, the quality of a linear approximation is arguably determined by its unique principal correlation.

The preceding paragraph can be generalized as follows. Let (U, V) be an approximation of a function $\mathsf{F} : G \to H$. In many cases, the principal correlations determine the optimal data complexity of known-plaintext distinguishers based on estimates of the evaluations $\langle v_i, T^{\mathsf{F}} u_i \rangle_H$ for $r \leq \max\{\dim U, \dim V\}$ pairs (u_i, v_i) in $U \times V$.

As in Section 3.2.2, the estimators of $\langle v_i, T^{\mathsf{F}} u_i \rangle_H$ for $i = 1, \ldots, r$ are given by

$$t(u_i, v_i) = \frac{|G||H|}{q} \sum_{j=1}^{q} \overline{u_i(x_j)} v_i(y_j) \,,$$

with $(x_1, y_1), \ldots, (x_q, y_q)$ the given plaintext-ciphertext pairs. Let $\boldsymbol{t}_{\mathsf{real}}(u_i, v_i)$ denote the i^{th} estimator when the pairs are obtained from the real cipher and $\boldsymbol{t}_{\mathsf{ideal}}(u_i, v_i)$ the i^{th} estimator for uniform random pairs. It will be assumed that the mean of $\boldsymbol{t}_{\mathsf{ideal}}(u_i, v_i)$ is zero.

For independent plaintext-ciphertext pairs, the joint distribution of the estimators tends to normal as $q \to \infty$. This follows from the multivariate central limit theorem, although one should bear in mind that convergence need not be fast. If in addition the variances of $\boldsymbol{t}_{\mathsf{real}}(u_i, v_i)$ and $\boldsymbol{t}_{\mathsf{ideal}}(u_i, v_i)$ are equal, then the acceptance region of an optimal distinguisher is defined by a separating

hyperplane. That is, a linear combination of the estimators is used as the test-statistic of a simple hypothesis test. This is known as linear discriminant analysis in the statistics literature. The data complexity is determined by the distance between the averages of the real and ideal test-statistics, measured in units of standard deviation. Hence, the optimal choice of the separating hyperplane maximizes this distance for constant variance:

$$\begin{aligned} \Delta = \max_{c_1,\dots,c_r\in\mathbb{C}} \max_{\substack{u_1,\dots,u_r\in U\\ v_1,\dots,v_r\in V}} \left| \mathsf{E} \sum_{i=1}^{r} \overline{c_i}\, \boldsymbol{t}_{\mathsf{real}}(u_i,v_i) \right| \\ \text{subject to} \quad \mathsf{Var} \sum_{i=1}^{r} \overline{c_i}\, \boldsymbol{t}_{\mathsf{ideal}}(u_i,v_i) = 1/q\,, \end{aligned} \tag{3.2}$$

where $(c_1,\dots,c_r)$ is a vector orthogonal to the separating hyperplane. Since the estimators are unbiased, $\mathsf{E}\sum_{i=1}^{r} \overline{c_i}\, \boldsymbol{t}_{\mathsf{real}}(u_i,v_i) = \sum_{i=1}^{r} \overline{c_i}\langle v_i, T^{\mathsf{F}} u_i\rangle_H$. For the variance, a trite calculation shows that

$$\mathsf{Var} \sum_{i=1}^{r} \overline{c_i}\, \boldsymbol{t}_{\mathsf{ideal}}(u_i,v_i) = \frac{1}{q} \sum_{1\le i,j\le r} \overline{c_j \langle v_i, v_j\rangle_H}\; c_i \langle u_i, u_j\rangle_G\,.$$

To simplify the solution of (3.2), the Frobenius inner product from Section 3.2.1 will be used. Recall that the Frobenius inner product between linear operators $A : \mathbb{C}^G \to \mathbb{C}^H$ and $B : \mathbb{C}^G \to \mathbb{C}^H$ is $\langle A, B\rangle_{\mathsf{fr}} = \operatorname{Tr}(A^\dagger B)$. The Frobenius norm of A is $\|A\|_{\mathsf{fr}} = \sqrt{\langle A, A\rangle_{\mathsf{fr}}}$. Using this notation, one can see that

$$\sum_{i=1}^{r} \overline{c_i} \langle v_i, T^{\mathsf{F}} u_i\rangle_H = \sum_{i=1}^{r} \overline{c_i} \operatorname{Tr}\left(T^{\mathsf{F}} u_i \langle v_i, \cdot\rangle_H\right) = \left\langle \textstyle\sum_{i=1}^{r} c_i\, v_i \langle u_i, \cdot\rangle_G,\ \langle V, U\rangle_{\mathsf{F}} \right\rangle_{\mathsf{fr}}.$$

Similarly, the variance can be rewritten as

$$\sum_{1\le i,j\le r} \overline{c_j \langle v_i, v_j\rangle_H}\; c_i \langle u_i, u_j\rangle_G = \left\| \textstyle\sum_{i=1}^{r} c_i v_i \langle u_i, \cdot\rangle_G \right\|_{\mathsf{fr}}^2.$$

Setting $X = \sum_{i=1}^{r} c_i v_i \langle u_i, \cdot\rangle_G$, the optimization problem (3.2) is equivalent to

$$\begin{aligned} \Delta = \max_{X\,:\,U\to V} \left| \left\langle \langle V, U\rangle_{\mathsf{F}},\ X \right\rangle_{\mathsf{fr}} \right| \\ \text{subject to} \quad \|X\|_{\mathsf{fr}} = 1 \text{ and } \operatorname{rank} X \le r\,. \end{aligned} \tag{3.3}$$

The solution to (3.3) is given by Theorem 3.8, which can be interpreted as a refinement of the Cauchy-Schwarz inequality (Theorem 3.2). It can also be rewritten as a variant of the Eckart-Young theorem [132] involving relative errors.

Theorem 3.8. *Let U and V be finite-dimensional inner product spaces, and let $\langle \cdot, \cdot \rangle_{\mathsf{fr}}$ be the induced Frobenius inner product. For every linear map $L : U \to V$ with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_d$, and for every linear map $X : U \to V$ with* $\operatorname{rank} X \leq r \leq d$,

$$|\langle L, X \rangle_{\mathsf{fr}}| \leq \|X\|_{\mathsf{fr}} \sqrt{\sum_{i=1}^{r} \sigma_i^2}\,.$$

Furthermore, equality is achieved for $X = \sum_{i=1}^{r} (\sigma_i / \sigma)\, u_i \langle v_i, \cdot \rangle$ where u_i and v_i are left and right singular vectors corresponding to σ_i and $\sigma = \sqrt{\sum_{i=1}^{r} \sigma_i^2}$.

Due to Theorem 3.8, we have $\Delta = \sqrt{\sum_{i=1}^{r} \sigma_i^2}$ where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$ are the first r principal correlations of (U, V). Assuming normality, using $q = \alpha / \sum_{i=1}^{r} \sigma_i^2$ data results in success probability $P_{\mathsf{S}} = \Phi(\Phi^{-1}(P_{\mathsf{F}}) - \sqrt{\alpha})$ for false-positive rate P_{F}.

Choosing $r = d$ results in the lowest data complexity, *i.e.* around $1/\|\langle U, V \rangle_{\mathsf{F}}\|_{\mathsf{fr}}^2$ because the squared Frobenius norm is equal to the sum of the squared principal correlations. The quantity $\|\langle U, V \rangle_{\mathsf{F}}\|_{\mathsf{fr}}^2$ regularly pops up in previous work about statistical aspects of linear cryptanalysis. Two examples are given below.

Example 3.11 (Capacity)**.** Let $\mathsf{F} : G \to H$ be a function and $S \subset \widehat{G}$, $T \subset \widehat{H}$ subsets of group characters. Relative to the Fourier basis, the approximation map of the multiple linear approximation $(\operatorname{Span} T, \operatorname{Span} S)$ is represented by a submatrix of C^{F} with coordinates $C^{\mathsf{F}}_{\chi,\psi}$ for ψ in S and χ in T. Hence, the sum of the squared principal correlations is equal to

$$\|\langle V, U \rangle_{\mathsf{F}}\|_{\mathsf{fr}}^2 = \sum_{\substack{\chi \in T \\ \psi \in S}} \left| C^{\mathsf{F}}_{\chi,\psi} \right|^2 .$$

This quantity is known as the fixed-key capacity, and it is inversely proportional to the data complexity of an optimal distinguisher. Despite this result, the squared principal correlations are *not* equal to the correlations of the individual linear approximations. ▷

The Frobenius norm of the approximation matrix is also related to the squared Euclidean imbalance, which was shown to be inversely proportional to the data complexity of optimal distinguishers based on balanced projections by Baignères *et al.* [16].

Example 3.12 (Squared Euclidean imbalance)**.** Let $\mathsf{F} : G \to H$ be a permutation and $\mathsf{P}_G : G \to Z_G$ and $\mathsf{P}_H : H \to Z_H$ balanced projections.

As discussed in Section 3.4.2, these projections correspond to subspaces $U = \mathrm{Span}\{\delta_z \circ \mathsf{P}_G \mid z \in Z_G\}$ and $V = \mathrm{Span}\{\delta_z \circ \mathsf{P}_H \mid z \in Z_H\}$ by the pullback construction. The approximation map $\langle V, U\rangle_{\mathsf{F}}$ can be represented as a matrix with coordinates

$$\frac{\langle \delta_y \circ \mathsf{P}_H, T^{\mathsf{F}}(\delta_x \circ \mathsf{P}_G)\rangle_H}{\|\delta_y \circ \mathsf{P}_H\|_H \, \|\delta_x \circ \mathsf{P}_G\|_G}$$

$$= \sqrt{\frac{\Pr\left[\mathsf{P}_G(\boldsymbol{z}_G) = x\right]}{\Pr\left[\mathsf{P}_H(\boldsymbol{z}_H) = y\right]}} \Pr\left[\mathsf{P}_H(\mathsf{F}(\boldsymbol{z}_G)) = y \,|\, \mathsf{P}_G(\boldsymbol{z}_G) = x\right],$$

where $\boldsymbol{z}_G$ is uniform random on G and $\boldsymbol{z}_H$ is uniform random on H. Since the approximations considered by Baignères *et al.* are balanced, $\Pr\left[\mathsf{P}_G(\boldsymbol{z}_G) = x\right] = |Z_G|/|G|$ and $\Pr\left[\mathsf{P}_H(\boldsymbol{z}_H) = y\right] = |Z_H|/|H|$, and the prefactor simplifies to $\sqrt{|Z_G|/|Z_H|}$. It follows that the Frobenius norm of $\langle V, U\rangle_{\mathsf{F}}$ is given by

$$\|\langle V, U\rangle_{\mathsf{F}}\|_{\mathsf{fr}}^2 = \frac{|Z_G|}{|Z_H|} \sum_{\substack{x \in Z_G \\ y \in Z_H}} \Pr\left[\mathsf{P}_H(\mathsf{F}(\boldsymbol{z}_H)) = y \,|\, \mathsf{P}_G(\boldsymbol{z}_G) = x\right]^2.$$

In particular, $\|\langle U, V\rangle_{\mathsf{F}}\|_{\mathsf{fr}}^2 - 1$ is equal to the *squared Euclidean imbalance* as defined by Baignères *et al.* [16, Definition 7]. The term -1 is due to the trivial invariant corresponding to the uniform distribution. ▷

Although $r = d$ results in the lowest data complexity, it can be useful to choose $r < d$. Indeed, decreasing r reduces the time complexity of the attack. This trade-off is especially attractive when some of the principal correlations are comparatively small.

Finally, it should be emphasized that the above results only apply to known-plaintext distinguishers. The chosen-plaintext data complexity can be much lower.

3.6 Trails

As discussed in Section 2.5.2, the standard approach to compute the approximation map of a non-perfect, nonzero-correlation approximation of a function $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$ is based on gluing together approximations for the functions $\mathsf{F}_1, \ldots, \mathsf{F}_r$ using the dominant trail approximation. This technique was formalized in Corollary 2.1 for the general case, and carries over to this chapter without changes.

This section investigates two special cases of Corollary 2.1 that are important in the context of linear cryptanalysis. First, Section 3.6.1 derives a general piling-up principle from Corollary 2.1 with a single dominant trail. Second, Section 3.6.2 briefly discusses the 'constructive interference' phenomenon and applies the piling-up principle to analyze the constructive interference of linear trails in functions with invariants.

3.6.1 Piling-up principle

For a single dominant trail and using approximations with orthogonal complements, Corollary 2.1 reduces to Corollary 3.5 below. This result is called the piling-up principle after the special case of ordinary linear cryptanalysis.

Corollary 3.5 (Piling-up principle)**.** *Let* $(U_1, U_2, \ldots, U_{r+1})$ *be a trail for a function* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$*. The approximation map of the approximation* (U_{r+1}, U_1) *of* F *satisfies*

$$\langle U_{r+1}, U_1 \rangle_{\mathsf{F}} = \langle U_{r+1}, U_r \rangle_{\mathsf{F}_r} \cdots \langle U_3, U_2 \rangle_{\mathsf{F}_2} \langle U_2, U_1 \rangle_{\mathsf{F}_1} + E \,,$$

where the error term E *is given by*

$$E = \sum_{V_2, \ldots, V_r} \langle U_{r+1}, V_r \rangle_{\mathsf{F}_r} \cdots \langle V_3, V_2 \rangle_{\mathsf{F}_2} \langle V_2, U_1 \rangle_{\mathsf{F}_1} \,,$$

where $(V_2, \ldots, V_r) \in \prod_{i=2}^{r} \{U_i, U_i^{\perp}\}$ *such that* $V_i = U_i^{\perp}$ *for at least one* i*.*

Several piling-up principles for variants of linear cryptanalysis have been proposed in the literature, including for nonlinear approximations [159], partitioning cryptanalysis [160] and projection functions [16, 280]. Traditionally, these piling-up principles have been justified by invoking a Markov chain assumption. That is, one multiplies transition probabilities as if they correspond to independent events. Since all transitions are actually deterministic[4], this assumption can never be true. Furthermore, when it fails, it is often hard to understand why or how to resolve the problem.

The dominant trail interpretation of the piling-up principle is theoretically more sound, but it was previously limited to the case of ordinary linear cryptanalysis. For example, Beierle *et al.* [26] propose to compute the correlation of nonlinear approximation by applying ordinary linear cryptanalysis to a nonlinear transformed variant of a cipher. However, there is no canonical

[4]It is perhaps relevant here to stress that round keys are fixed throughout an attack, with a few exceptions such as in tweakable block ciphers based on the TWEAKKEY framework [169]. However, in the latter case, round keys are even partially controlled by the adversary.

representative set of all nonlinear approximations, so this introduces an undesirably arbitrary choice. A basis-free perspective helps to overcome this difficulty.

Corollary 3.5 provides a general piling-up principle that is applicable in all of the above-mentioned cases, but it also offers an alternative motivation based on dominant trails rather than independence assumptions. The premise is that each approximation in a trail corresponds to a transformation of its input space, followed by an orthogonal projection on the input space of the next approximation. Each of these successive projections introduces an error, but orthogonal projection is optimal in the sense that it keeps the inner product between the state and its approximation maximal and the norm of the error minimal. Ultimately, the approximation relies on the assumption that the contribution of the chosen trail is dominant compared to that of the trails involving one-or-more complementary subspaces.

The one-dimensional case of Corollary 3.5, with U_i spanned by a character χ_i, was already discussed in Section 3.3.3. The composition result of Beierle *et al.* [26, Theorem 3] for nonlinear approximations is another special case of Corollary 3.5 for one-dimensional approximations. A few examples of the higher-dimensional case can be found in the literature. Two examples are discussed below: the piling-up principle for properties based on projections functions as proposed by Baignères *et al.* [16] and Wagner [280], and some cases of multiple-linear cryptanalysis.

Example 3.13 (Projection functions)**.** Suppose all spaces U_i are pullbacks along balanced projection functions $\mathsf{P}_i : G_i \to Z_i$. As shown in Example 3.12, relative to the bases $\{\delta_x \circ \mathsf{P}_i / \|\delta_x \circ \mathsf{P}_i\|_{G_i} \mid x \in Z_i\}$ for U_i, the map $\langle U_{i+1}, U_i \rangle_{\mathsf{F}_i}$ can be represented by a matrix with coordinates

$$\sqrt{\frac{|Z_i|}{|Z_{i+1}|}} \Pr\left[\mathsf{P}_{i+1}(\mathsf{F}(\boldsymbol{z})) = y \,|\, \mathsf{P}_i(\boldsymbol{z}) = x\right],$$

where $\boldsymbol{z}$ is uniform random on G_i. That is, up to a constant scaling of rows and columns, $\langle U_{i+1}, U_i \rangle_{\mathsf{F}_i}$ is the transition matrix considered in [16, 280]. These works follow the Markov chain assumption, which leads to using the product of round transition matrices as an approximation for the true transition matrix. The row and column scaling indeed cancel out, so that Corollary 3.5 yields the same result up to scaling of rows and columns. ▷

Example 3.14 (Multiple linear cryptanalysis)**.** For any multiple linear approximation, the coordinate representation of $\langle U_{i+1}, U_i \rangle_{\mathsf{F}_i}$ in the Fourier basis is a submatrix of the correlation matrix C^{F_i}. Hence, Corollary 3.5 suggests multiplying submatrices of correlation matrices.

This approach has been (sometimes indirectly) used in several works, notably in the multiple linear cryptanalysis of PRESENT [90], PUFFIN [195] and SPONGENT [70]. For these ciphers, strong approximations can be found by taking into account all trails with masks of Hamming weight one. This approach is often combined with key-averaging, leading to multiplying matrices of squared correlations. However, a careful analysis of the key-dependency would be both feasible and preferable in most cases. ▷

3.6.2 Linear approximations from invariants

A minimal condition for the applicability of the piling-up approximation is that one chooses the best trail from a predetermined class of candidates, where the principal correlations can be used as a measure of quality. Indeed, the error term in Corollary 3.5 can be large if other trails are better or comparable.

However, it is also possible that the class of candidate trails is too limited to obtain a good estimate for $\langle U_{r+1}, U_1 \rangle_{\mathsf{F}}$. In the context of linear cryptanalysis, this is related to a phenomenon that was called 'constructive interference' by Daemen and Rijmen [104]. This phenomenon occurs when a large number of linear trails with a small correlation result in a linear approximation with an extraordinary large correlation. The qualification 'extraordinary' is important. An approximation with n dominant trails with correlation $\pm c$, is expected to have an overall correlation of $\pm\sqrt{n}\, c$. However, if all n trails have the same sign, then the correlation will be $\pm n\, c$. This is constructive interference.

In some cases, constructive interference can be explained by considering different types of trails. For example, a good linear approximation may be explained by a trail of nonlinear approximations. In Chapter 6, an example of a perfect linear approximation over full-round Midori-64 with modified round constants will be presented. However, full-round Midori-64 does not admit any high-correlation linear *trails*. This observation can be thought of as an extreme case of a more general phenomenon. At Crypto 2012, Abdelraheem *et al.* [1] showed that invariant subspaces give rise to linear approximations with higher-than-expected correlation. The same observation was later generalized to plateaued nonlinear invariants by Beierle *et al.* [26]. Plateaued nonlinear invariants are characterized by a flat Walsh-Hadamard transformation, taking only two values up to sign. The results of Abdelraheem *et al.* [1] and Beierle *et al.* [26] can be summarized and generalized as follows.

Theorem 3.9. *Let* $\mathsf{F} : G \to G$ *be a function on a finite commutative group* G*. Let* u *in* $\mathbb{C}^G$ *be any function with Fourier transform* $\widehat{u} = \mathscr{F}_G\, u$ *such that* $|\widehat{u}(\chi)| = 1/\sqrt{|\operatorname{supp} \widehat{u}|}$ *for all* χ *in* $\operatorname{supp} \widehat{u}$ *and zero elsewhere. If* $\operatorname{Span}\{u\}$ *is*

an invariant of F*, then there exist characters* χ *and* ψ *in* $\operatorname{supp}\widehat{u}$ *such that* $|C^{\mathsf{F}}_{\chi,\psi}| \geq 1/|\operatorname{supp}\widehat{u}|$.

Proof. By Definition 2.20, it holds that (the sum is over χ and ψ in $\operatorname{supp}\widehat{u}$)

$$1 = \left|\langle \widehat{u}, C^{\mathsf{F}}\widehat{u}\rangle\right| = \left|\sum_{\chi,\psi} \overline{\widehat{u}(\chi)}\widehat{u}(\psi)C^{\mathsf{F}}_{\chi,\psi}\right| \leq |\operatorname{supp}\,\widehat{u}| \max_{\chi,\psi} \left|C^{\mathsf{F}}_{\chi,\psi}\right|.$$

It follows that $|C^{\mathsf{F}}_{\chi,\psi}| \geq 1/|\operatorname{supp}\widehat{u}|$ for at least one pair (χ,ψ). □

Note that the same result is spread over two theorems in previous work [26, Theorem 4 and 5]: one for invariant subspaces, and one for plateaued nonlinear invariants. This illustrates the convenience of the general definitions. To apply the results to the case of invariant subspaces, one only needs to know that the Fourier transformation of the indicator function of a subgroup H of G is flat with support size $|G|/|H|$. This follows from the Poisson-summation formula [262, Theorem 1]. See also the first entry of Table 3.2 for $G = \mathbb{F}_2^n$.

Theorem 3.9 and the results above illustrate that a strong approximation using one kind of property can result in unexpectedly good approximations using other properties. This can be understood using Corollary 3.5. For example, let $\operatorname{Span}\{u\}$ with $\|u\|_G = 1$ be any invariant of F with $\widehat{u} = \mathscr{F}_G\, u$. Consider an ordinary linear approximation, *i.e.* a pair $(\operatorname{Span}\{\psi\}, \operatorname{Span}\{\chi\})$ where ψ, χ are characters. The correlation of the linear approximation over F can be estimated using the following trail:

$$\psi \xrightarrow[\langle u,\,\psi\rangle_G]{I} u \xrightarrow[1]{T^{\mathsf{F}}} u \xrightarrow[\langle \chi,\,u\rangle_G]{I} \chi.$$

Corollary 3.5 yields the estimate $|\langle u,\psi\rangle_G\langle\chi,u\rangle_G| = |\widehat{u}(\psi)\widehat{u}(\chi)|$ for the absolute correlation. If $\widehat{u}$ is flat as in Theorem 3.9, then the piling-up approximation suggests that all approximations with ψ and χ in $\operatorname{supp}\widehat{u}$ will have a correlation of roughly $1/|\operatorname{supp}\widehat{u}|$. In fact, this resolves a question of Beierle *et al.*, who note that "our arguments are non-constructive and therefore, we are not able to identify those highly-biased linear approximations" [26, §1]. In fact, it is easy to identify the highly-biased approximations in practice: generically, *any* approximation with ψ and χ in $\operatorname{supp}\widehat{u}$ will do.

3.7 Rank-one approximations

It is often convenient to represent the domain of a cipher as an array of m cells, because most of the operations in the cipher act on the cells in an independent

way. In fact, in ciphers such as the AES, only the MixColumns step results in diffusion between cells. That is, let $G = H^m$ for some commutative group H. Recall that up to canonical isomorphism, $\mathbb{C}^{H^m} = (\mathbb{C}^H)^{\otimes m}$ and similarly for the dual group.

Example 3.15. The probability distribution of a state with independent cells having distributions $p_1, \ldots, p_m$, is represented by the rank-one tensor $p_1 \otimes \cdots \otimes p_m$ (see Section 2.2.3 for definitions). ▷

A rank-one approximation (U, V) is any approximation such that U and V are spanned by a rank-one tensor. No further conditions are imposed on U and V. An important class of rank-one approximations is obtained from balanced functions $\mathsf{P} : H^m \to Z$ such that $\mathsf{P}(x_1, \ldots, x_m) = \sum_{i=1}^m \mathsf{P}_i(x_i)$. As shown in Table 3.2 for $H = \mathbb{F}_2^n$, one can associate a vector space to P spanned by a function $\chi \circ \mathsf{P} = \bigotimes_{i=1}^m \chi \circ \mathsf{P}_i$ with χ a character of Z. Equivalently, the Fourier transformation of the corresponding vector space is spanned by

$$\mathscr{F}_G(\chi \circ \mathsf{P}) = \bigotimes_{i=1}^m \mathscr{F}_H(\chi \circ \mathsf{P}_i),$$

where $\mathscr{F}_H(\chi \circ \mathsf{P}_i)$ is the Walsh-Hadamard transformation of P_i when $H = \mathbb{F}_2^n$ and $Z = \mathbb{F}_2$. The invariants that will be discussed in Chapter 6 and the nonlinear approximations considered by Beierle *et al.* [26] and in Section 3.8 are of this type.

3.7.1 Theoretical analysis of rank-one trails

By Corollary 3.1 (1), the correlation matrix of a layer of m identical S-boxes S is equal to $(C^{\mathsf{S}})^{\otimes m}$. Indeed, correlation matrices are themselves tensors and the tensor rank (not to be confused with matrix rank) of $(C^{\mathsf{S}})^{\otimes m}$ is one. This expresses the fact that the S-box layer preserves independence of cells. A similar result holds for the key-addition step. Whereas the S-box layer preserves the rank-one structure of approximations, the linear layer tends to increase the rank. In fact, it is reasonable to interpret the rank as a measure of diffusion between the state cells. Since the correlation matrix of any function $\mathsf{F} : H^m \to H^m$ is a tensor, it can be decomposed as

$$C^{\mathsf{F}} = \sum_{i=1}^{r} \lambda_i \bigotimes_{j=1}^m C_{i,j},$$

where $C_{i,j}$ are $|H| \times |H|$ matrices and r is the tensor rank of C^{F}.

Lemma 3.1. *Let* $\mathsf{F} : (\mathbb{F}_2^n)^m \to (\mathbb{F}_2^n)^m$ *be a function such that* $\mathsf{F} = (\mathsf{G}, \mathsf{G}, \ldots, \mathsf{G})$ *for some* $\mathsf{G} : \mathbb{F}_2^n \to \mathbb{F}_2^n$*. If* $C^{\mathsf{G}} = \sum_{i=1}^{r} \lambda_i \bigotimes_{j=1}^{n} C_{i,j}$*, then*

$$C^{\mathsf{F}} = \sum_{i_1,\ldots,i_m} (\textstyle\prod_{k=1}^{m} \lambda_{i_k}) \bigotimes_{k=1}^{m} \bigotimes_{j=1}^{n} C_{i_k,j},$$

where $i_1, \ldots, i_m \in \{1, \ldots, r\}$*. In particular,* C^{F} *has tensor rank at most* r^m*.*

Proof. By Corollary 3.1 (1), it holds that $C^{\mathsf{F}} = (C^{\mathsf{G}})^{\otimes m}$. The result follows by expanding this expression using the multilinearity of tensor products. □

Lemma 3.1 can be used to obtain a decomposition of the correlation matrix of the MixColumn map of the block ciphers Midori-64 [18] and MANTIS [29] into 2^8 rank-one terms. This map $\mathsf{M} : (\mathbb{F}_2^4)^4 \to (\mathbb{F}_2^4)^4$ can be represented by the following matrix over $\mathbb{F}_{2^4}$:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

Up to a reordering of the input bits, one can think of M as a map $\widetilde{\mathsf{M}} = (\mathsf{L}, \mathsf{L}, \mathsf{L}, \mathsf{L})$ where L corresponds to the same matrix as above, but over $\mathbb{F}_2$. Specifically, $\widetilde{\mathsf{M}} = \sigma \mathsf{M} \sigma$ where $\sigma : (\mathbb{F}_2^4)^4 \to (\mathbb{F}_2^4)^4$ is the bit permutation defined by $\sigma_i(x_1, \ldots, x_4) = (x_{1,i}, \ldots, x_{4,i})$. Since C^{L} is a 16×16 matrix, one can check that

$$C^{\mathsf{L}} = \frac{1}{2}\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{\otimes 4} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{\otimes 4} + \begin{bmatrix} 1 & 0 \\ 0 & \text{-1} \end{bmatrix}^{\otimes 4} - \begin{bmatrix} 0 & 1 \\ \text{-1} & 0 \end{bmatrix}^{\otimes 4} \right).$$

To see this, it is helpful to observe that C^{L} is symmetric as a tensor. Since $\widetilde{\mathsf{M}} = \sigma \mathsf{M} \sigma$ where σ is a linear map corresponding to a reordering of bits, it follows from Theorem 3.5 (2) and Lemma 3.1 that

$$C^{\mathsf{M}} = 2^{-4} \sum_{i_1,i_2,i_3,i_4} (\textstyle\prod_{j=1}^{4} \lambda_{i_j}) \left[\bigotimes_{j=1}^{4} C_{i_j}\right]^{\otimes 4}.$$

where $i_1, \ldots, i_4 \in \{1, \ldots, 4\}$, $\lambda_1 = \lambda_2 = \lambda_3 = 1$ and $\lambda_4 = -1$ and

$$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 1 & 0 \\ 0 & \text{-1} \end{bmatrix}, \quad C_4 = \begin{bmatrix} 0 & 1 \\ \text{-1} & 0 \end{bmatrix}.$$

Hence, the tensor rank of C^{M} is at most 2^8. This is significantly lower than the worst-case of 2^{16}. Practically speaking, this enables a detailed analysis of rank-one approximations for Midori-64 in Section 3.8.3. In fact, one can show that this decomposition is minimal *i.e.* the rank of C^{M} is equal to 2^8.

Lemma 3.2 (Lemma 3.5 in [114]). *Let $V_1, \ldots, V_d$ be finite-dimensional vector spaces over $\mathbb{C}$. If $x_{i,1}, \ldots, x_{i,r}$ in V_i are linearly independent for $i = 1, \ldots, d$, then the vector $\sum_{i=1}^{r} x_{1,i} \otimes x_{2,i} \otimes \cdots \otimes x_{d,i}$ in $\bigotimes_{i=1}^{d} V_i$ has tensor rank r.*

To see why Lemma 3.2 implies the result, let V_i be the vector space of 16×16 matrices over $\mathbb{C}$. This is an inner product space under the Frobenius inner product. It is easy to check that the matrices C_i defined above are mutually orthogonal with respect to this inner product. This implies the mutual orthogonality of the matrices $\left(\bigotimes_{j=1}^{4} C_{i_j}\right)^{\otimes 4}$. The result follows by the linear independence of orthogonal vectors.

3.7.2 Automated analysis of rank-one trails

Let $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$ be a permutation on H^m. By Corollary 3.5, an optimal rank-one trail for F can be found by solving the following optimization problem:

$$\begin{aligned} &\text{maximize } \sum_{i=1}^{r} \log_2 \left|\left\langle \textstyle\bigotimes_{j=1}^{m} v_{i+1,j},\, C^{\mathsf{F}_i} \bigotimes_{j=1}^{m} v_{i,j} \right\rangle\right| \\ &\text{subject to } \|v_{i,j}\|_2 = 1 \text{ for } i = 1, \ldots, r+1,\ j = 1, \ldots, m \\ &\qquad\qquad v_{i,j}(\mathbb{1}) = 0 \text{ for } (i,j) \in A \text{ and } v_{i,j} = \delta_{\mathbb{1}} \text{ otherwise,} \end{aligned}$$

where the last condition ensures that the vectors $v_{i,j}$ are active and balanced, *i.e.* orthogonal to $\delta_{\mathbb{1}}$, on a predetermined pattern of cells A. Clearly, at least one cell must be active to obtain a nontrivial result. In practice, it is better to take the logarithm of the objective function in order to avoid vanishing gradients.

The above is an optimization problem over the product of several copies of the $(|H| - 1)$-dimensional unit sphere. This domain is a Riemannian manifold, and common iterative numerical optimization techniques such as steepest descent and conjugate gradients have been generalized to this setting [253]. This is the basic approach behind the automated method proposed in this section. An implementation of this method for $H = \mathbb{F}_2^n$ can be found online[5]. It relies on the Pymanopt library [269].

The power of this method lies in the fact that it enables iterative convergence to an optimal trail. This is made possible because the general nature of rank-one approximations results in a relaxed, continuous optimization problem rather than a discrete one. Although it is sometimes necessary to ensure that the outermost vectors of the trail correspond to (for example) Boolean functions,

[5] `https://github.com/TimBeyne/Geometric-approach`

there is no reason to impose the same condition on vectors which are internal to the trail.

Example 3.16. The tool can be applied to find rank-one invariants of arbitrary functions with a limited number of input and output bits, which is a difficult problem in general. For example, Figure 3.2 shows the iterative convergence towards an invariant of the Midori-64 linear layer. This process takes about a second on an ordinary computer. By optimizing over the sphere of unit-norm vectors in the eigenspaces $E_\lambda(C^{\mathsf{S}})$ of the correlation matrix C^{S}, joint invariants for the linear and S-box layer can be found. An alternative approach to finding such invariants will be discussed in Chapter 6.

The standard Riemannian variant of the conjugate gradients algorithm was chosen as the optimization method. For further details such as the line search method, the reader is referred to the PYMANOPT source code (no custom optimizations were introduced). The tool also implements a barrier method to find *all* rank-one invariants for a given linear layer. ▷

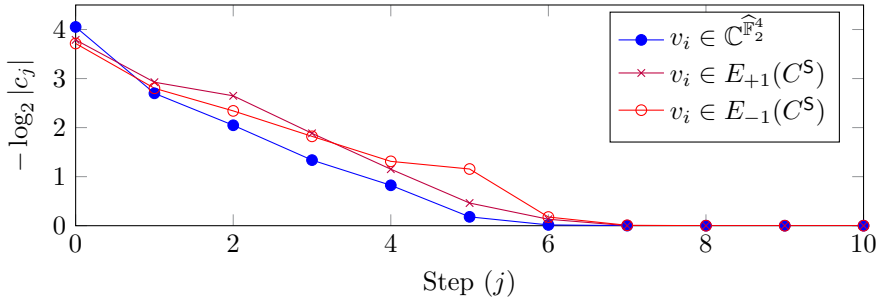


Figure 3.2: Correlation c_j at each step of the optimization process for finding invariants of the form $v_1 \otimes v_2 \otimes v_3 \otimes v_4$ with $v_i(\mathbb{1}) = 0$ for the Midori-64 linear layer.

A number of challenges remain for larger problems. These include addressing key-dependence and convergence issues.

In many cases, it is possible to fix the key and analyze the key-dependence afterwards. Due to Theorem 2.12, using the Fourier transformation simplifies this process. The disadvantage of this approach is that it does not ensure that the approximation will hold for many keys. This can be resolved by optimizing over spheres $B\,\mathbb{S}^{2^n}$ where B is a matrix whose columns are an orthonormal basis for an invariant subspace of several key-additions.

Another issue is that the optimization problem may have many local optima. Lack of global convergence is mainly an issue when the number of variables

in the problem is large – for the examples discussed in this thesis, this issue was not encountered. For large problems, several restarts may be necessary to find a globally optimal solution. The tool automates this process, but restarting necessarily slows down convergence. For this reason, it is advisable to predetermine the activity pattern and to enforce symmetries wherever possible.

3.8 Open problem of Beierle *et al.*

This section explains observations of Beierle *et al.* [26] on a nonlinear approximation of two rounds of Midori-64. More broadly, the results lead to a deeper understanding of many nonlinear approximations of the Midori-64 round function.

3.8.1 Problem statement

Beierle *et al.* [26, Section 4.4] consider a nonlinear approximation over two rounds of Midori-64, restricted to a single column of the state. Denote this function by F. Its correlation matrix is equal to

$$C^{\mathsf{F}} = C^{\mathsf{M}}[C^{\mathsf{S}}]^{\otimes 4} C^{l} C^{\mathsf{M}}[C^{\mathsf{S}}]^{\otimes 4} C^{k},$$

where k and l are 16-bit keys, S is the S-box and M the matrix defined in Section 3.7.1. Recall from Section 3.6.1 that Beierle *et al.* describe nonlinear approximations using linear properties of a nonlinearly transformed representation of the cipher. The details of their approach will not be discussed here; the geometric framework will be used instead. The nonlinear functions considered by Beierle *et al.* are of the form $\sum_{i=1}^{4} \mathsf{P}_i(x)$ with $\mathsf{P}_i : \mathbb{F}_2^4 \to \mathbb{F}_2$ and consequently, as discussed in Section 3.7 on page 88, correspond to approximations spanned by rank-one vectors. Specifically, the pair of nonlinear functions considered in [26, Section 4.4] corresponds to a one-dimensional approximation $(\mathrm{Span}\{u \otimes v^{\otimes 3}\}, \mathrm{Span}\{u \otimes v^{\otimes 3}\})$ for F with

$$u = 1/4 \cdot (0, 1, 0, -1, 0, 1, 0, -1, 0, -1, 0, 1, 0, -1, 0, -3)$$

$$v = 1/2 \cdot (0, 0, 0, -1, 0, 0, 0, -1, 0, \quad 0, 0, 1, 0, \quad 0, 0, -1)\,.$$

The coordinates above are given relative to the Fourier basis $\{\chi_w/2^4 \mid w \in \mathbb{F}_2^4\}$ with lexicographic ordering of w. Note that v is an eigenvector of C^{S}. Beierle *et al.* estimate the correlation of the above approximation using the following two-round trail, which has absolute correlation at least $(9/32)^2$:

$$u \otimes v^{\otimes 3} \xrightarrow[\pm 1 \text{ or } \pm 1/2]{[C^{\mathsf{S}}]^{\otimes 4} C^{k}} u \otimes v^{\otimes 3} \xrightarrow[9/16]{C^{\mathsf{M}}} u \otimes v^{\otimes 3} \xrightarrow[\pm 1 \text{ or } \pm 1/2]{[C^{\mathsf{S}}]^{\otimes 4} C^{l}} u \otimes v^{\otimes 3} \xrightarrow[9/16]{C^{\mathsf{M}}} u \otimes v^{\otimes 3}. \quad (3.4)$$

The computation of the correlation over C^{M} was done by a direct evaluation of the inner product $\langle u \otimes v^{\otimes 3}, C^{\mathsf{M}}\, u \otimes v^{\otimes 3} \rangle$. This trail was believed to hold whenever k and l are in $\mathbb{F}_2^4 \times K^3$, with $K = \{(0,0,x,y) \mid x,y \in \mathbb{F}_2\}$. The weak key set K ensures the invariance of the tensor product factor v under key addition. Based on the above, one estimates an absolute correlation of at least $(9/32)^2$ over F. However, Beierle *et al.* experimentally observe that this estimate is not accurate:

(i) When $l \in (\mathbb{F}_2^4 \setminus K) \times K^3$, the correlation is found to equal zero.

(ii) For other keys, the correlation takes on various values, but is always significantly larger than the estimated minimum of 81/1024. Specifically, for k and l in K^4, the correlation ranges from 35/64 to $40/64 = 5/8$. For other keys, it lies between 39/256 and 65/256.

In their conclusion, the authors remark that understanding this phenomenon is "a major open problem".

3.8.2 Optimal rank-one trail

As shown in Section 3.7.1, the effect of the linear layer is nontrivial and this makes finding an optimal rank-one trail difficult. Hence, a simple explanation for observation (ii) could be that the trail (3.4) proposed by Beierle *et al.* is not a good guess. Using the tool from Section 3.7.2, it is easy to find the optimal rank-one trail – ignoring the effect of key-addition for now. The tool yields the following trail with absolute correlation at most 9/16:

$$u \otimes v^{\otimes 3} \xrightarrow[\pm 3/4 \text{ or } \pm 1/4]{[C^{\mathsf{S}}]^{\otimes 4} C^k} v^{\otimes 4} \xrightarrow[1]{C^{\mathsf{M}}} v^{\otimes 4} \xrightarrow[\pm 1]{[C^{\mathsf{S}}]^{\otimes 4} C^l} v^{\otimes 4} \xrightarrow[3/4]{C^{\mathsf{M}}} u \otimes v^{\otimes 3}.$$

A short calculation shows that the third step requires that $l \in K^4$, otherwise the trail has correlation zero. Furthermore, the correlation 3/4 in the first step occurs if and only if $k \in K^4$. In hindsight, one might have guessed the above trail without detailed analysis: the choice of $v^{\otimes 4}$ as an intermediate step is natural, since $v^{\otimes 4}$ is an invariant for the round function. This is an instance of the general phenomenon discussed in the last paragraph of Section 3.6.2.

3.8.3 Theoretical analysis of the problem

The correlations predicted by the rank-one trail obtained in Section 3.8.2 are within 10 to 30% of the observed correlations reported by Beierle *et al.* [26, Tables

1–4]. However, the trail does not yet explain the zero-correlation approximation. In this section, the results from Section 3.7.1 will be used to find a *minimal and complete* set of rank-one trails for the approximation.

The propagation of $u \otimes v^{\otimes 3}$ under F will first be analyzed. For the zero-correlation case, the miss-in-the-middle strategy can be used. It will then be shown that a relatively short formula for the exact key-dependent correlation of the approximation can be computed.

Let $k = (k_1, k_2, \ldots, k_{16})$ and $l = (l_1, l_2, \ldots, l_{16})$. The results in Section 3.7.1 can be used to compute the image of $u \otimes v^{\otimes 3}$ under one round:

$$C^{\mathsf{M}}[C^{\mathsf{S}}]^{\otimes 4} C^k \, u \otimes v^{\otimes 3} = -\nu \, C^{\mathsf{M}}\big(C^{\mathsf{S}} C^{(k_1,\ldots,k_4)} u\big) \otimes v^{\otimes 3} = \nu \, v \otimes \big(\textstyle\sum_{i=1}^{16} c_i \, v_i^{\otimes 3}\big),$$

where $\nu = -\prod_{i=2}^{4} (-1)^{k_{4i-1}+k_{4i}}$. The coefficients c_i and the vectors v_i are listed in Table 3.3. Note that, because C^{M} has rank 2^8, one initially obtains 2^8 terms. However, this can be reduced to 16 by grouping terms appropriately. This can be done manually by exploiting the structure of the rank-decomposition, but Sage code to automate this can be found online[6]. Since the vectors v_i are mutually orthogonal, Lemma 3.2 implies that the above decomposition is minimal. Interestingly, not all of the vectors v_i correspond to Boolean functions or probability distributions.

A similar computation can be performed for the inverse of the second round. Specifically, recalling that S and M are involutions,

$$C^{l}[C^{\mathsf{S}}]^{\otimes 4} C^{\mathsf{M}} \, u \otimes v^{\otimes 3} = \mu \, C^{(l_1,\ldots,l_4)} v \otimes \big(\textstyle\sum_{i=1}^{8} d_i \bigotimes_{j=1}^{3} (C^{(l_{4j},\ldots,l_{4j+4})} w_i)\big).$$

The coefficients d_i and the vectors w_i are listed in Table 3.4 and $\mu = (-1)^{l_3+l_4+1}$. The minimality of the above decomposition can again be established using Lemma 3.2.

3.8.4 Zero-correlation approximation

Let $U = \mathrm{Span}\{v\} \otimes (\mathbb{C}^{\widehat{\mathbb{F}_2^4}})^{\otimes 3}$ and $V = \mathrm{Span}\{C^{(l_1,\ldots,l_4)} \, v\} \otimes (\mathbb{C}^{\widehat{\mathbb{F}_2^4}})^{\otimes 3}$. The decompositions above clearly imply the following inclusions:

$$C^{\mathsf{M}}[C^{\mathsf{S}}]^{\otimes 4} C^k \, u \otimes v^{\otimes 3} \in U \quad \text{and} \quad C^{l}[C^{\mathsf{S}}]^{\otimes 4} C^{\mathsf{M}} \, u \otimes v^{\otimes 3} \in V.$$

Consequently, if U and V are orthogonal, then the miss-in-the-middle principle implies that the approximation has correlation zero. This happens whenever

[6] https://github.com/TimBeyne/Geometric-approach/blob/main/midori_rankone.sage

Table 3.3: Vectors v_i and corresponding coefficients in the forward decomposition, with $\kappa_i = (-1)^{k_i}$.

i	$2\,v_i$	$\kappa_4\,c_i$
1	$(0,0,0,\ \ 1,0,0,0,-1,0,0,0,\ \ 1,0,0,0,\ \ 1)$	$1/32\,(3\kappa_1\kappa_2\kappa_3+\kappa_1\kappa_2-\kappa_1\kappa_3-\kappa_1+2\kappa_2)$
2	$(0,0,0,\ \ 1,0,0,0,-1,0,0,0,-1,0,0,0,-1)$	$-1/32\,(\kappa_1\kappa_3-2\kappa_2\kappa_3+\kappa_1+2\kappa_2+\kappa_3+1)$
3	$(0,0,0,-1,0,0,0,-1,0,0,0,\ \ 1,0,0,0,-1)$	$-1/16\,\kappa_3\,(3\kappa_1\kappa_2+\kappa_1+\kappa_2+1)$
4	$(0,0,0,-1,0,0,0,-1,0,0,0,-1,0,0,0,\ \ 1)$	$1/32\,(3\kappa_1\kappa_2\kappa_3-\kappa_1\kappa_2+2\kappa_1-\kappa_3+1)$
5	$(0,0,\ \ 1,0,0,0,-1,0,0,0,\ \ 1,0,0,0,\ \ 1,0)$	$1/32\,(2\kappa_1\kappa_2+\kappa_1\kappa_3-\kappa_1-\kappa_3-1)$
6	$(0,0,\ \ 1,0,0,0,-1,0,0,0,-1,0,0,0,-1,0)$	$-1/32\,(3\kappa_1\kappa_2\kappa_3-\kappa_1\kappa_2+\kappa_1\kappa_3-2\kappa_2\kappa_3-\kappa_1)$
7	$(0,0,-1,0,0,0,-1,0,0,0,\ \ 1,0,0,0,-1,0)$	$-1/32\,(3\kappa_1\kappa_2\kappa_3+\kappa_1\kappa_2-2\kappa_2\kappa_3-2\kappa_2+\kappa_3-1)$
8	$(0,0,-1,0,0,0,-1,0,0,0,-1,0,0,0,\ \ 1,0)$	$-1/16\,(\kappa_2-1)$
9	$(0,\ \ 1,0,0,0,-1,0,0,0,\ \ 1,0,0,0,\ \ 1,0,0)$	$1/32\,\kappa_1\,(3\kappa_2\kappa_3+\kappa_2-\kappa_3+1)$
10	$(0,\ \ 1,0,0,0,-1,0,0,0,-1,0,0,0,-1,0,0)$	$-1/32\,(2\kappa_1\kappa_2+\kappa_1\kappa_3+\kappa_1-\kappa_3+1)$
11	$(0,-1,0,0,0,-1,0,0,0,\ \ 1,0,0,0,-1,0,0)$	$1/16\,\kappa_3\,(3\kappa_1\kappa_2-1)$
12	$(0,-1,0,0,0,-1,0,0,0,-1,0,0,0,\ \ 1,0,0)$	$1/32\,(3\kappa_1\kappa_2\kappa_3+\kappa_1\kappa_2-2\kappa_1\kappa_3+\kappa_3+1)$
13	$(\ \ 1,0,0,0,-1,0,0,0,\ \ 1,0,0,0,\ \ 1,0,0,0)$	$-1/32\,(\kappa_1\kappa_3-\kappa_1-\kappa_3+1)$
14	$(\ \ 1,0,0,0,-1,0,0,0,-1,0,0,0,-1,0,0,0)$	$-1/32\,\kappa_1\,(3\kappa_2\kappa_3-\kappa_2-\kappa_3-1)$
15	$(-1,0,0,0,-1,0,0,0,\ \ 1,0,0,0,-1,0,0,0)$	$-1/32\,(3\kappa_1\kappa_2\kappa_3-\kappa_1\kappa_2-\kappa_3-1)$
16	$(-1,0,0,0,-1,0,0,0,-1,0,0,0,\ \ 1,0,0,0)$	$1/16\,(\kappa_1-1)$

Table 3.4: Vectors w_i and corresponding coefficients in the backward decomposition, with $\lambda_i = (-1)^{l_{4j+i}}$.

i	$4\,C^{l_{4j}\|\cdots\|l_{4j+4}} w_i$	d_i
1	$2(0, 0, 0, -1, 0, 0, 0, -1, 0, 0, 0, 1, 0, 0, 0, -1)$	$-3/8\lambda_3\lambda_4$
2	$2(0, 0, -\lambda_3, -\lambda_2\lambda_3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -\lambda_3, \lambda_3\lambda_4)$	$1/8$
3	$(0, 0, 2\lambda_3, 0, -2, 0, 0, 0, -1, \lambda_4, \lambda_3, \lambda_3\lambda_4, 1, \lambda_4, -\lambda_3, \lambda_3\lambda_4)$	$1/8$
4	$(0, 0, 0, 0, 0, -2\lambda_4, 2\lambda_3, 0, 1, -\lambda_4, -\lambda_3, \lambda_3\lambda_4, 1, \lambda_4, \lambda_3, \lambda_3\lambda_4)$	$1/8$
5	$(0, \lambda_4, 0, -\lambda_3\lambda_4, -1, 2\lambda_4, \lambda_3, 2\lambda_3\lambda_4, 0, -\lambda_4, 0, \lambda_3\lambda_4, -1, 0, \lambda_4, 0)$	$-1/8$
6	$(0, -\lambda_4, 0, -\lambda_3\lambda_4, 1, 0, \lambda_3, 0, -1, 2\lambda_4, \lambda_3, 0, 0, -\lambda_4, 2\lambda_3, \lambda_3\lambda_4)$	$1/8$
7	$(0, -\lambda_4, 0, \lambda_3\lambda_4, 1, 0, -\lambda_3, 0, 1, 0, \lambda_3, 2\lambda_3\lambda_4, -2, \lambda_4, 0, \lambda_3\lambda_4)$	$1/8$
8	$(0, \lambda_4, 0, \lambda_3\lambda_4, 1, 0, \lambda_3, 0, 0, -\lambda_4, 2\lambda_3, \lambda_3\lambda_4, 1, -2\lambda_4, -\lambda_3, 0)$	$-1/8$

$\langle v, C^{(l_1,\ldots,l_4)}\, v\rangle = 0$. That is,

$$\begin{aligned}&\big\langle (0,0,0,1,0,0,0,1,0,0,0,-1,0,0,0,1),\\ &\ (0,0,0,1,0,0,0,(-1)^{l_2},0,0,0,(-1)^{1+l_1},0,0,0,(-1)^{l_1+l_2})\big\rangle\\ &= 1+(-1)^{l_1}+(-1)^{l_2}+(-1)^{l_1+l_2},\end{aligned}$$

which equals zero unless $l_1 = l_2 = 0$. This explains the condition $l \in (\mathbb{F}_2^4 \backslash K) \times K^3$ observed by Beierle *et al.* [26].

3.8.5 Refining the correlation estimate

Assume that $l \in K^4$, so that the correlation is nonzero. A closer inspection of the vectors v_i and w_j reveals that $|\langle v_i, C^{(l_{4j},\ldots,l_{4j+4})} w_j\rangle| \leq 1/2$ unless $i = 3$ and $j = 1$. That is, when the inner product

$$\big\langle C^l C^{\mathsf{M}} [C^{\mathsf{S}}]^{\otimes 4}\, u \otimes v^{\otimes 3}, [C^{\mathsf{S}}]^{\otimes 4} C^{\mathsf{M}} C^k\, u \otimes v^{\otimes 3}\big\rangle$$

is expanded using the decomposition above, the term corresponding to $c_3 d_1$ has a weight of one whereas all other terms have weight at most 2^{-3}. Since $v_3 = w_1 = v$, this term corresponds to the trail from Section 3.8.2.

The correlation estimate can be improved by including additional trails. In principle, all 128 terms in the expanded inner product between the forward and backward expressions can be computed. A Sage script that computes a short formula for the exact key-dependent correlation of the approximation can be found online[7].

In fact, due to the low rank of C^{M}, the same technique can be used to analyze all rank-one approximations of F. This includes all linear approximations. In general, the minimal number of rank-one trails can be higher or lower than 16×8 depending on the choice of the input and output property.

[7] `https://github.com/TimBeyne/Geometric-approach`

4

Differential cryptanalysis

In this chapter, the geometric approach from Chapter 2 is applied to differential cryptanalysis. Unlike for linear cryptanalysis, the one-dimensional theory already leads to new results and is consequently the focus of this chapter. The systematic application of Section 2.4 to differential cryptanalysis leads to the theory of *quasidifferential trails*, which keep track of probabilistic linear relations on the values satisfying a differential characteristic in a theoretically sound way. It is shown that the fixed-key probability of a differential can be expressed as the sum of the correlations of its quasidifferential trails.

This chapter is based on the paper "Differential cryptanalysis in the fixed-key model" [56] from Crypto 2022, which was joint work with Vincent Rijmen. The main differences are that this chapter considers differential cryptanalysis on arbitrary finite commutative groups rather than $\mathbb{F}_2^n$ only, and the omission of the applications to Rectangle, KNOT, Speck and Simon. The latter results are discussed in Chapter 8 instead.

4.1 Introduction

The central problem of differential cryptanalysis is to count the number of inputs of a function for which a given input difference results in a particular output difference or, what amounts to the same, to compute the probability of a differential. For functions that can be written as a composition of simple operations, the standard procedure is to analyze sequences of intermediate differences or *characteristics*. The probability of a characteristic is then heuristically estimated by multiplying the probabilities of the intermediate differentials. In the context of block ciphers, Lai, Massey and Murphy [191] showed that this procedure yields the correct value of the *key-averaged probability* for Markov ciphers.

However, since the key is fixed throughout a differential attack, even the average data complexity cannot be computed from the average probability of differentials alone. Hence, Lai *et al.* [191] introduced an additional assumption known as the *hypothesis of stochastic equivalence*. It states that the probability for each key

is close to the average probability. In practice, it turns out that the probability can vary significantly between keys. Hence, standard assumptions may lead to incorrect conclusions. Furthermore, averages may hide weak key attacks that can considerably degrade security. Finally, the same formalism is used even when there is no key, such as for cryptographic permutations, or when the cryptanalyst has full control over the key, such as in many hash functions.

Daemen and Rijmen [105] showed that the fixed-key probability of two-round characteristics of the AES is either zero or 2^h, with h an integer independent of the key. Such characteristics are called *plateau characteristics*, and have been used in several other contexts [75, 83, 205, 217, 259]. Although plateau characteristics are the only systematic method to analyze fixed-key probabilities for S-box-based ciphers, their scope remains limited. They assume that the input or output values satisfying a differential over the S-box form an affine space. Furthermore, their analysis becomes difficult for more than two rounds.

For constructions relying on modular additions, several techniques were developed in the context of collision attacks on hash functions. These methods keep track of additional information about the values satisfying a characteristic. For example, the breakthrough results of Wang and Yu [283] rely on *signed differences*. De Cannière and Rechberger [112] extended these to *generalized differences*, allowing arbitrary constraints to be imposed on individual bits. Leurent [199] proposed a framework for ARX-constructions based on two-bit conditions. Xu *et al.* [287] introduced *signed sums*, which are single-bit conditions. Despite their merit, these techniques have significant limitations. Imposing conditions directly on values becomes difficult for keyed functions, since key-additions result in conditions that potentially depend on many unknown bits. Hence, these methods are limited to keyless functions except for localized or key-independent effects. Furthermore, the conditions that are imposed cannot fully explain the probability of a characteristic, and the right choice of the type of conditions to use depends on the function under analysis.

From a theoretical viewpoint, it can be argued that the standard approach to differential cryptanalysis is incomplete, since it does not offer any tools to compute probabilities beyond the average case. Furthermore, as mentioned above, existing techniques such as plateau characteristics and generalized differences still have important limitations. This is in contrast to linear cryptanalysis, where it is known that the correlation of a linear approximation is precisely equal to the sum of the correlations of all its linear trails.

To achieve parity with linear cryptanalysis, this chapter applies the geometric approach from Chapter 2 – and Section 2.4 in particular – to differential cryptanalysis. Section 4.3 constructs a preferred 'quasidifferential basis' that diagonalizes round key additions. This leads to an extension of the difference-

distribution table that will be called the *quasidifferential transition matrix*. It represents a pushforward operator relative to the quasidifferential basis. Hence, quasidifferential transition matrices are analogous to correlation matrices in linear cryptanalysis. Their main properties are discussed in Section 4.3.2. By applying the results from Section 2.4.3, quasidifferential transition matrices in turn lead to a notion of *quasidifferential trails*. Corollary 4.2 shows that the sum of the correlations of all quasidifferential trails in a differential is equal to its exact fixed-key probability. This is an immediate but powerful consequence of Theorem 2.6.

The remaining sections of this chapter likewise focus on the one-dimensional theory of quasidifferential trails. An efficient algorithm to compute the quasidifferential transition matrix of a given function is given in Section 4.4. Section 4.5 shows that the probability of a differential characteristic is the sum of the correlations of all quasidifferential trails in the characteristic. This is a refinement of the abovementioned result for differentials. A few quasidifferential trails often capture the essence of the key-dependence. For example, Knudsen [180] already observed significant deviations from the hypothesis of stochastic equivalence for the characteristics used in the differential analysis of DES. This effect is explained in Section 4.5.2 by taking into account one additional one-round quasidifferential trail.

4.2 Mathematical setting

Unlike linear cryptanalysis, differential cryptanalysis is concerned with pairs of values rather than individual values. Hence, the geometric approach from Chapter 2 will be applied to the function $(x, y) \mapsto (\mathsf{F}(x), \mathsf{F}(y))$ with $\mathsf{F} : G \to H$ the primitive. As in Chapter 3, it will be assumed that G and H are finite commutative groups. In this case, the pushforward operator of the function F is equal to $T^{\mathsf{F}} \otimes T^{\mathsf{F}}$. Many results in this chapter can be generalized to arbitrary functions from $G \oplus G$ to $H \oplus H$, which may be interesting in the context of related key and related cipher attacks. To keep the presentation simple, only functions from G to H will be considered in this chapter.

Following Definition 2.6, cryptanalytic properties are pairs of subspaces of $\mathbb{C}^{G \oplus G} = \mathbb{C}^G \otimes \mathbb{C}^G$ and $(\mathbb{C}^{H \oplus H})^\vee = (\mathbb{C}^H)^\vee \otimes (\mathbb{C}^H)^\vee$. As in Chapter 3, the field $\mathbb{C}$ is equipped with its ordinary absolute value function $|\cdot|$.

The vector spaces $\mathbb{C}^{G \oplus G}$ and $\mathbb{C}^{H \oplus H}$ are equipped with the Euclidean norm, up to a constant factor. This choice is discussed in more detail in Section 4.2.1.

Finally, the group action that will lead to the one-dimensional theory is defined in Section 4.2.2.

4.2.1 Motivation for the Euclidean norm

Due to its self-duality property, the Euclidean norm is a natural choice from a theoretical point of view. This was already discussed in Section 3.2.1. Nevertheless, it is not necessarily the only reasonable choice and the statistical motivation from Chapter 3 does not carry over completely to differential cryptanalysis. In any case, the norm plays a less prominent role in the one-dimensional theory – which is the focus of this chapter.

Throughout this chapter, the Euclidean norm will be scaled by a factor $\sqrt{|G|}$. This is to ensure consistency with Chapter 3, since fixing the input difference to zero reduces differential cryptanalysis to linear cryptanalysis. Specifically, the norm of u in $\mathbb{C}^{G \oplus G}$ is equal to

$$\|u\|_G = \sqrt{|G|}\,\|u\|_2\,,$$

which is self-dual with respect to the inner product $\langle v, u\rangle_G = |G|\,\langle v, u\rangle$. A similar definition is used for $\mathbb{C}^{H \oplus H}$. As in Chapter 3, this implies that $v \mapsto \langle v, \cdot\rangle_H$ is an isometric anti-isomorphism between $\mathbb{C}^{H \oplus H}$ and $(\mathbb{C}^{H \oplus H})^\vee$.

4.2.2 Group action

Since G is a group, any t in G acts on $G \oplus G$ by $(x, y) \mapsto (x - t, y - t)$. This action extends to $\mathbb{C}^{G \oplus G}$ by

$$\big((T^t \otimes T^t)\, f\big)(x, y) = f(x + t, y + t)\,,$$

where the positive sign is due to the fact that $T^t \otimes T^t$ describes forward propagation.

The action $(x, y) \mapsto (x - t, y - t)$ extends the action $x \mapsto x - t$ on G from Chapter 3. Since most primitives involve key and constant additions, it is a natural choice. However, the order of G is only the square root of the dimension of $\mathbb{C}^{G \oplus G}$, so the action of G does not determine a complete basis. The remaining degrees of freedom will be used up in Section 4.3.1

4.3 One-dimensional theory

This section applies the one-dimensional theory from Section 2.4 to the setting that was introduced in Section 4.2. Section 4.3.1 shows how the group action from Section 4.2.2 leads to a preferred basis by following Section 2.4.4. This is less straightforward than in Chapter 3, because the group action alone does not lead to a unique choice. The resulting basis is called the quasidifferential basis, and expressing pushforward operators relative to this basis yields quasidifferential transition matrices. The properties of these matrices are discussed in Section 4.3.2. The resulting one-dimensional theory of trails is described in Section 4.3.3. It is an extension of the standard description of differential cryptanalysis, which is only valid on average for independent and uniform random round keys.

4.3.1 Quasidifferential basis

In ordinary differential cryptanalysis, input pairs are chosen from a set $A = \{(x, x+a) \mid x \in G\}$ and the number of output pairs that are contained in a set $B = \{(x, x+b) \mid x \in G\}$ is counted. This corresponds to the cryptanalytic property $(\mathrm{Span}\{\mathbb{1}_A\}, \mathrm{Span}\{\langle \mathbb{1}_B, \cdot \rangle_H\})$.

Due to the above, an appropriate basis for $\mathbb{C}^{G \oplus G}$ (and likewise for $\mathbb{C}^{H \oplus H}$) should include the indicator functions $(x, y) \mapsto \delta_a(y - x)$ for all a in G. Clearly, these functions alone do not form a basis. However, the group action defined in Section 4.2.2 leads to additional conditions. Below, these conditions will be used to construct a complete basis for $\mathbb{C}^{G \oplus G}$.

Recall from Section 2.4.4 that $\mathbb{C}^{G \oplus G}$ together with the G-action defined by T^t for t in G is a representation of G. Furthermore, the elements of the following subrepresentation are fixed by all $T^t \otimes T^t$ with t in G:

$$\left\{(x, y) \mapsto u(y - x) \mid u \in \mathbb{C}^G\right\} \subset \mathbb{C}^{G \oplus G}$$

Let $\varphi : \mathbb{C}^G \otimes \mathbb{C}^G \to \mathbb{C}^{G \oplus G}$ be the isomorphism of representations defined by $\varphi(u \otimes v) : (x, y) \mapsto u(x)\, v(y - x)$, so that $\mathbb{C}^{G \oplus G} \cong_\varphi \mathbb{C}^G \otimes \mathbb{C}^G$. Since the action of G on $\mathbb{C}^G$ is diagonalized by the character basis (see Section 3.3.1), the representation $\mathbb{C}^{G \oplus G}$ can be decomposed as the following direct sum of subrepresentations:

$$\mathbb{C}^{G \oplus G} \cong_\varphi \bigoplus_{\chi \in \widehat{G}} \mathrm{Span}\{\chi\} \otimes \mathbb{C}^G .$$

Hence, any choice of $|G|$ bases for $\mathbb{C}^G$ yields a basis that diagonalizes $T^t \otimes T^t$ for all t in G. For simplicity, assume that the same basis is used for each

term in the decomposition. If the basis is required to include the functions $(x, y) \mapsto \delta_a(y - x)$, then one obtains Definition 4.1 uniquely up to scaling.

Definition 4.1 (Quasidifferential basis)**.** Let G be a finite commutative group. For every χ in $\widehat{G}$ and a in G, the function $q_{\chi,a} : G \oplus G \to \mathbb{C}$ is defined by

$$q_{\chi,a}(x, y) = \chi(x)\, \delta_a(y - x)/|G|\,.$$

The set of all functions $q_{\chi,a}$ will be called the quasidifferential basis for $\mathbb{C}^{G\oplus G}$.

The functions $q_{\chi,a}$ are not only linearly independent, but also orthogonal. This is shown in Theorem 4.1, which also states the important translation-invariance property.

Theorem 4.1. *The quasidifferential basis defined in Definition 4.1 is translation-invariant and orthogonal. Specifically:*

(1) For all (ψ, a) *and* (χ, b) *in* $\widehat{G}\oplus G$*, it holds that* $\langle q_{\chi,b},\, q_{\psi,a}\rangle_G = \delta_\chi(\psi)\, \delta_b(a)$*.*

(2) For all (χ, a) *in* $\widehat{G} \oplus G$ *and* t *in* G*, it holds that*

$$q_{\chi,a}(x + t, y + t) = \chi(t)\, q_{\chi,a}(x, y)\,.$$

Proof. The first result follows immediately from the orthogonality of group characters, *i.e.* Theorem 3.4 (2). Explicitly,

$$\langle q_{\psi,b},\, q_{\chi,a}\rangle_G = \frac{1}{|G|} \sum_{(x,y)\in G\oplus G} \overline{\psi(x)}\, \delta_b(y - x)\, \chi(x)\, \delta_a(y - x)\,.$$

Indeed, if $a \neq b$, then $y - x = a$ and $y - x = b$ never hold simultaneously. If $a = b$, then the result follows from the orthogonality of the characters χ and ψ. The translation-invariance follows from the fact that $\chi(x + t) = \chi(t)\, \chi(x)$. □

The change-of-basis transformation $\mathscr{Q}_G : \mathbb{C}^{G\oplus G} \to \mathbb{C}^{\widehat{G}\oplus G}$ from the standard basis to the quasidifferential basis is defined by $\mathscr{Q}_G\, q_{\chi,a} = \delta_{(\chi,a)}$ as in Definition 2.10. Equivalently, $\mathscr{Q}_G$ can be defined as in Definition 4.2, analogous to the Fourier transformation in Definition 3.5.

Definition 4.2 (Quasidifferential change-of-basis)**.** The quasidifferential change-of-basis transformation is the linear map $\mathscr{Q}_G : \mathbb{C}^{G\oplus G} \to \mathbb{C}^{\widehat{G}\oplus G}$ defined by

$$\big(\mathscr{Q}_G f\big)(\chi, a) = \big\langle q_{\chi,a},\, f\big\rangle_G = \sum_{x\in G} \overline{\chi(x)} f(x, x + a)\,,$$

for all χ in $\widehat{G}$ and a in G.

4.3.2 Quasidifferential transition matrices

Let $\mathsf{F} : G \to H$ be a function. Following Section 2.4.1, the pushforward operator $T^{\mathsf{F}} \otimes T^{\mathsf{F}}$ can be expressed relative to the quasidifferential basis. This results in Definition 4.3. This definition should be compared with Definitions 2.11 and 3.6.

Definition 4.3 (Quasidifferential transition matrix)**.** Let $\mathsf{F} : G \to H$ be a function between finite commutative groups G and H. Define $D^{\mathsf{F}} : \mathbb{C}^{\widehat{G}\oplus G} \to \mathbb{C}^{\widehat{G}\oplus G}$ as the pushforward operator of F relative to the quasidifferential basis. That is, $D^{\mathsf{F}} = \mathscr{Q}_H(T^{\mathsf{F}} \otimes T^{\mathsf{F}})\mathscr{Q}_G^{-1}$.

The quasidifferential transition matrix of F is the coordinate representation of D^{F} with respect to the standard bases of $\mathbb{C}^{\widehat{G}\oplus G}$ and $\mathbb{C}^{\widehat{H}\oplus H}$.

To make Definition 4.3 more concrete, we compute the coordinates of D^{F}. By the same conventions as in Chapters 2 and 3, the coordinates of D^{F} will be indexed by pairs (ψ, a) in $\widehat{G} \oplus G$ and (χ, b) in $\widehat{H} \oplus H$. By the orthogonality of the quasidifferential basis (Theorem 4.1 (1)), it holds that $\mathscr{Q}_H \delta_{(\chi,b)} = |H|\, q_{\chi,b}$ and consequently

$$D^{\mathsf{F}}_{(\chi,b),\,(\psi,a)} = \langle \delta_{(\chi,b)}, D^{\mathsf{F}} \delta_{(\psi,a)} \rangle = \langle q_{\chi,b} \circ \mathsf{F},\, q_{\psi,a} \rangle_H \,.$$

Working this out yields the following expression:

$$\begin{aligned} D^{\mathsf{F}}_{(\chi,b),\,(\psi,a)} &= \frac{1}{|G|} \sum_{(x,y)\in G\oplus G} \psi(x)\overline{\chi(\mathsf{F}(x))}\, \delta_a(y-x)\delta_b(\mathsf{F}(y)-\mathsf{F}(x)) \\ &= \frac{1}{|G|} \sum_{\substack{x\in G \\ \mathsf{F}(x+a)=\mathsf{F}(x)+b}} \overline{\chi(\mathsf{F}(x))}\psi(x)\,. \end{aligned} \tag{4.1}$$

For $\psi = \mathbb{1}_G$ and $\chi = \mathbb{1}_H$, (4.1) reduces to the probability of the differential with input difference a and output difference b. That is,

$$D^{\mathsf{F}}_{(\mathbb{1}_H,b),(\mathbb{1}_G,a)} = \mathsf{DDT}^{\mathsf{F}}_{a,b}/|G|\,.$$

For $a = 0$ and $b = 0$, one obtains the coordinates of the correlation matrix of F. Specifically,

$$D^{\mathsf{F}}_{(\chi,0),(\psi,0)} = C^{\mathsf{F}}_{\chi,\psi}\,.$$

More generally, the right hand side of (4.1) can be interpreted as a kind of correlation matrix for the function F but restricted to the right pair set of the differential (a, b).

Example 4.1. For $G = \mathbb{F}_2^n$ and $H = \mathbb{F}_2^m$, (4.1) shows that $D^{\mathsf{F}}_{(\chi,b),(\psi,a)}$ equals

$$\big(2\Pr[v^{\mathsf{T}}\mathsf{F}(\boldsymbol{x}) = u^{\mathsf{T}}\boldsymbol{x} \mid \mathsf{F}(\boldsymbol{x}+a) = \mathsf{F}(\boldsymbol{x}) + b] - 1\big)\Pr[\mathsf{F}(\boldsymbol{x}+a) = \mathsf{F}(\boldsymbol{x}) + b]\,,$$

when $\chi(x) = (-1)^{v^{\mathsf{T}}x}$ and $\psi(x) = (-1)^{u^{\mathsf{T}}x}$ and with $\boldsymbol{x}$ uniform random on $\mathbb{F}_2^n$. The first factor is the correlation of the linear approximation (ψ, χ), but conditional on the event that the differential (a, b) holds. The second factor is the probability of the differential (a, b). That is, the coordinates of D^{F} express the correlations of probabilistic linear relations ('linear approximations') between the input and output values of the right pairs. ▷

The following results summarize the main properties of quasidifferential transition matrices. Corollary 4.1 is a consequence of Theorem 2.5. Theorem 4.2 is specific to the quasidifferential basis, although property (1) is identical to Theorem 3.5 (1) for correlation matrices. This is because it only relies on the orthogonality of the basis.

Corollary 4.1. *The quasidifferential transition matrix D^{F} of $\mathsf{F} : G \to H$ has the following properties:*

(1) If $\mathsf{F} = (\mathsf{F}_1, \ldots, \mathsf{F}_n)$, then $D^{\mathsf{F}} = \bigotimes_{i=1}^{n} D^{\mathsf{F}_i}$.

(2) If $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$, then $D^{\mathsf{F}} = D^{\mathsf{F}_r} \cdots D^{\mathsf{F}_2} D^{\mathsf{F}_1}$.

Theorem 4.2. *The quasidifferential transition matrix D^{F} of $\mathsf{F} : G \to H$ has the following properties:*

(1) If F is a bijection, then D^{F} is an orthogonal matrix.

(2) If $\mathsf{F}(x) = x - t$ for t in G, then $D^{\mathsf{F}}_{(\chi,b),(\psi,a)} = \chi(t)\,\delta_{\chi}(\psi)\,\delta_b(a)$.

(3) If F is a homomorphism, then $D^{\mathsf{F}}_{(\chi,b),(\psi,a)} = \delta_{\chi\circ\mathsf{F}}(\psi)\,\delta_b(\mathsf{F}(a))$.

Proof. Property (1) follows from the fact that T^{F} is a permutation matrix when F is a bijection and the fact that $\mathscr{Q}_G^{-1}$ and $\mathscr{Q}_H$ are unitary matrices with respect to appropriate inner products by Theorem 4.1 (1). The second property is due to the translation invariance and orthogonality of the quasidifferential basis (Theorem 4.1 (2)). Finally, Property (3) can be deduced from (4.1):

$$D^{\mathsf{F}}_{(\chi,b),\,(\psi,a)} = \frac{1}{|G|} \sum_{\substack{x\in G \\ \mathsf{F}(x+a)=\mathsf{F}(x)+b}} \overline{(\chi\circ\mathsf{F})(x)}\,\psi(x) = \delta_{\chi\circ\mathsf{F}}(\psi)\,\delta_b(\mathsf{F}(a))\,,$$

where the second equality follows from the orthogonality of characters and the fact that $\mathsf{F}(x + a) = \mathsf{F}(x) + b$ if and only if $b = \mathsf{F}(a)$. □

Example 4.2. Consider the S-box $\mathsf{S} : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ of the lightweight block cipher Rectangle, shown in Table 4.1. The 256×256 quasidifferential transition matrix of S is shown in Figure 4.1, with colors representing the absolute value of the entries. The integer indices correspond to pairs (χ_u, a) by the map $(\chi_u, a) \mapsto \mathsf{int}(u) + 16 \times \mathsf{int}(a)$, where $\mathsf{int}(u) = \sum_{i=1}^{4} u_i 2^{4-i}$ and $\chi_u(x) = (-1)^{u^\top x}$.

Table 4.1: The S-box of Rectangle (hexadecimal representation).

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$\mathsf{S}(x)$	6	5	c	a	1	e	7	9	b	0	3	d	8	f	4	2

Figure 4.1 immediately reveals a number of properties of quasidifferential transition matrices. The top-left square in Figure 4.1 corresponds to the correlation matrix of S. Each block shows the correlations of probabilistic linear relations between the input and output values for the right pairs. Hence, Figure 4.1 is a 'magnified' version of the difference-distribution table of S. ▷

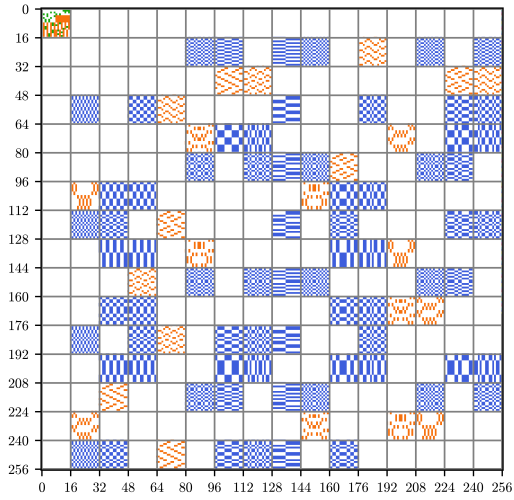


Figure 4.1: The quasidifferential transition matrix D^{S} of the Rectangle S-box S. Blue cells correspond to values of absolute value 1/8, orange cells to 1/4, and green cells to 1/2. Empty cells correspond to zeros.

4.3.3 Approximations and trails

A differential corresponds to a property (U, V) with $U = \mathrm{Span}\{q_{\mathbb{1},a}\}$ and $V = \mathrm{Span}\{\langle q_{\mathbb{1},b}, \cdot \rangle_H\}$. This can be generalized by considering nontrivial characters. Following Section 2.4.3, evaluating such a property is equivalent to computing $D^{\mathsf{F}}_{(\chi,b),(\psi,a)}$. This quantity will be called the correlation of the quasidifferential approximation $((\chi, a), (\psi, b))$.

Throughout this section, $\mathsf{F}_i : G_i \to G_{i+1}$ denotes a function between finite commutative groups G_i and G_{i+1}. As in Section 2.4.3, if $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$, then the correlation of an approximation of F can be estimated using trails. This is enabled by Theorem 2.6, which is repeated in Corollary 2.1 for the special case of the quasidifferential basis.

Corollary 4.2 (Sum of quasidifferential trails, *cf.* Theorem 2.6)**.** *If* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$*, then the correlation* $D^{\mathsf{F}}_{\varpi_{r+1},\varpi_1}$ *is equal to*

$$D^{\mathsf{F}}_{\varpi_{r+1},\varpi_1} = \sum_{\varpi_2,\ldots,\varpi_r} \prod_{i=1}^{r} D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i} ,$$

where the sum ranges over all intermediate pairs of characters and differences.

Following the notational conventions from Section 2.4.3, sequences $(\varpi_1, \ldots, \varpi_{r+1})$ of character-difference pairs $\varpi_i = (\chi_i, a_i)$ are called quasidifferential trails and their correlation is defined as $\prod_{i=1}^{r} D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i}$.

To illustrate the difference between Corollary 4.2 and the traditional approach to differential cryptanalysis, it is helpful to split the sum over trails into two parts as follows:

$$D^{\mathsf{F}}_{\varpi_{r+1},\varpi_1} = \sum_{\substack{\varpi_2,\ldots,\varpi_r \\ \chi_i=\mathbb{1} \text{ for all } i}} \prod_{i=1}^{r} D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i} + \sum_{\substack{\varpi_2,\ldots,\varpi_r \\ \chi_i\neq\mathbb{1} \text{ for some } i}} \prod_{i=1}^{r} D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i} . \qquad (4.2)$$

where $\varpi_i = (\chi_i, a_i)$ in each sum and for differentials, $\chi_1 = \mathbb{1}$ and $\chi_{r+1} = \mathbb{1}$. Traditionally, only the first term in (4.2) is considered. Indeed, a quasidifferential trail with $\chi_i = \mathbb{1}$ for all i is just a sequence of intermediate differences or 'differential trail'. The correlation of such quasidifferential trails is equal to the product of the one-round probabilities for differences $a_1, \ldots, a_{r+1}$. Hence,

$$\sum_{\substack{\varpi_2,\ldots,\varpi_r \\ \chi_i=\mathbb{1} \text{ for all } i}} \prod_{i=1}^{r} D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i} = \sum_{a_2,\ldots,a_r} \prod_{i=1}^{r} \Pr\left[\mathsf{F}_i(\boldsymbol{x}_i + a_i) = \mathsf{F}_i(\boldsymbol{x}_i) + a_{i+1}\right],$$

with $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r$ uniform random. The right-hand side above is the standard approximation for the probability of a differential. In fact, each term in the sum is meant to approximate the true probability of a characteristic, *i.e.* the probability that a particular sequence of intermediate differences is realized. The second term in (4.2) is then a correction to the error made in this approximation. Quasidifferential trails can also be used to compute the probability of differential characteristics; this is discussed in Section 4.5.

In practice, Corollary 4.2 is used by truncating the sum to a subset of dominant trails. This leads to Corollary 4.3, which is a special case of Corollary 2.1.

Corollary 4.3 (Dominant trail approximation *cf.* Corollary 2.1)**.** *Let* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$*. For all subsets* Λ *of the set* Ω *of all trails from* ϖ_1 *to* ϖ_{r+1}*,*

$$\left| D^{\mathsf{F}}_{\varpi_{r+1},\varpi_1} - \sum_{\varpi \in \Lambda} \prod_{i=1}^{r} D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i} \right| \leq \left| \sum_{\varpi \in \Omega \setminus \Lambda} \prod_{i=1}^{r} D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i} \right| ,$$

with $\varpi = (\varpi_1, \ldots, \varpi_{r+1})$*.*

Since the standard approach to differential cryptanalysis can be interpreted as an application of Corollary 4.3 that only considers trails with $\chi_i = \mathbb{1}$, important trails might be overlooked. This potentially results in incorrect conclusions. As discussed in more detail below, for ciphers with uniform and independent round keys, the additional quasidifferential trails do not affect the average probability. However, as mentioned in Section 4.1, average probabilities are generally insufficient to compute success probabilities and data-complexities of differential attacks. In contrast, Corollary 4.3 allows one to estimate the fixed-key probability of a differential.

If $\mathsf{F}_k = \mathsf{R}_r \circ \cdots \circ \mathsf{R}_1$ is a key-dependent function with k in $\bigoplus_{i=1}^{r} G_i$ and round functions $\mathsf{R}_i : G_i \to G_{i+1}$ defined by $\mathsf{R}_i(x) = \mathsf{F}_i(x + k_i)$ for $i = 1, \ldots, r$, then

$$D^{\mathsf{F}_k}_{\varpi_{r+1},\varpi_1} = \sum_{\varpi_2,\ldots,\varpi_r} \prod_{i=1}^{r} \chi_i(k_i)\, D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i} , \tag{4.3}$$

with $\varpi_i = (\chi_i, a_i)$. After averaging with respect to independent and uniform random round keys, (4.3) agrees with the standard approach of adding the products of one-round probabilities. This result can be generalized as follows.

Theorem 4.3. *Let* $\mathsf{F}_k = \mathsf{R}_r \circ \cdots \circ \mathsf{R}_1$ *with* $\mathsf{R}_i(x) = \mathsf{F}_i(x) + k_i$*. If* $\boldsymbol{k} = (\boldsymbol{k}_1, \ldots, \boldsymbol{k}_r)$ *is a random variable such that* $(\boldsymbol{k}_2, \ldots, \boldsymbol{k}_r)$ *is uniform random on a subset* K

of $\bigoplus_{i=2}^{r} G_i$*, then*

$$\Pr\left[\mathsf{F}_{\boldsymbol{k}}(\boldsymbol{x}+a)=\mathsf{F}_{\boldsymbol{k}}(\boldsymbol{x})+b\right]=\sum_{\substack{\chi_2,\dots,\chi_r\\ a_2,\dots,a_r\\ \chi_2\cdots\chi_r\in K^1}}\prod_{i=1}^{r} D^{\mathsf{F}_i}_{(\chi_{i+1},a_{i+1}),(\chi_i,a_i)}\,,$$

where $\chi_1=\mathbb{1}$*,* $\chi_{r+1}=\mathbb{1}$ *and the probability is over a uniform random* $\boldsymbol{x}$ *and over the keys* $\boldsymbol{k}_1,\dots,\boldsymbol{k}_r$*. In particular, for* $K=\bigoplus_{i=2}^{r} G_i$*, only quasidifferential trails with* $\chi_i=\mathbb{1}$ *contribute to the key-averaged probability of the differential.*

Proof. The result follows by averaging (4.3) with respect to the round keys:

$$\mathsf{E}_{\boldsymbol{k}} D^{\mathsf{F}_{\boldsymbol{k}}}_{\varpi_{r+1},\varpi_1}=\sum_{\varpi_2,\dots,\varpi_r}\left(\mathsf{E}_{\boldsymbol{k}}\textstyle\prod_{i=1}^{r}\chi_i(\boldsymbol{k}_i)\right)\displaystyle\prod_{i=1}^{r} D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i}\,,$$

with $\varpi_i=(\chi_i,a_i)$. The factor $\mathsf{E}_{\boldsymbol{k}}\prod_{i=1}^{r}\chi_i(\boldsymbol{k}_i)$ is zero unless $\chi_2\cdots\chi_r\in K^1$. □

Finally, (4.3) allows computing the variance of the probability of a differential:

$$\mathsf{E}_{\boldsymbol{k}}\left[D^{\mathsf{F}_{\boldsymbol{k}}}_{\varpi_{r+1},\varpi_1}\right]^2+\mathsf{Var}_{\boldsymbol{k}}\left[D^{\mathsf{F}_{\boldsymbol{k}}}_{\varpi_{r+1},\varpi_1}\right]=\sum_{\varpi_2,\dots,\varpi_r}\prod_{i=1}^{r}\left(D^{\mathsf{F}_i}_{\varpi_{i+1},\varpi_i}\right)^2.$$

This result is analogous to (3.1) in Chapter 3, *i.e.* the well-known result of Nyberg [224] about the variance of the correlation of linear approximations.

4.4 Computing quasidifferential transition matrices

The differential cryptanalysis of specific primitives using quasidifferential trails requires calculating the quasidifferential transition matrix for each round transformation. For affine functions, Theorem 4.2 (2) and (3) show how to compute the quasidifferential transition matrix.

In general, calculating the quasidifferential transition matrix is nontrivial because the dimensions of the matrix D^{F} scales with the size of the domain and codomain of F. In the following section, it is shown that this is not an issue for many primitives: an efficient method to compute the quasidifferential transition matrix for small (such as 4- or 8-bit) S-boxes is given, and larger functions often have structure that makes it possible to compute individual coordinates of D^{F} easily.

4.4.1 Small functions

If $\mathsf{F} : G \to H$ is a function between groups $G = \bigoplus_{i=1}^{n} G_i$ and $H = \bigoplus_{i=1}^{m} H_i$ with G_i and H_i small groups, then the matrix D^{F} can be computed using a number of operations roughly proportional to its number of elements. Specifically, the matrix D^{F} can be computed in $\mathcal{O}(|G|^2|H|^2 \sum_{i=1}^{n}(|G_i|^2 + |H_i|^2))$ time using a method similar to the fast Fourier transform.

Specifically, the matrices $\mathscr{Q}_G$ and $\mathscr{Q}_H$ satisfy $\mathscr{Q}_G = \bigotimes_{i=1}^{n} \mathscr{Q}_{G_i}$ and $\mathscr{Q}_H = \bigotimes_{i=1}^{n} \mathscr{Q}_{H_i}$. It follows that there exists an efficient algorithm for multiplication with $\mathscr{Q}_G$ and its inverse, analogous to the fast Fourier transform algorithm. This algorithm is based on the decomposition

$$\mathscr{Q}_G = (\mathscr{Q}_{G_1} \otimes I_{G_2} \otimes \cdots \otimes I_{G_n})(I_{G_1} \otimes \mathscr{Q}_{G_2} \otimes \cdots \otimes I_{G_n}) \cdots (I_{G_1} \otimes I_{G_2} \otimes \cdots \otimes \mathscr{Q}_{G_n}),$$

with I_{G_i} the identity map on $\mathbb{C}^{G_i \oplus G_i}$. Up to constant factors, the cost of computing n consecutive matrix-vector products with the matrices above is equal to

$$\sum_{i=1}^{n} |G_i|^4 \prod_{j \neq i} |G_i|^2 = |G|^2 \sum_{i=1}^{n} |G_i|^2 .$$

Hence, since $D^{\mathsf{F}} = \mathscr{Q}_H \, (T^{\mathsf{F}} \otimes T^{\mathsf{F}}) \, \mathscr{Q}_G^{-1}$ by Definition 4.3, the matrix D^{F} can be computed by applying this divide-and-conquer multiplication algorithm to both the rows and columns of $T^{\mathsf{F}} \otimes T^{\mathsf{F}}$. A Sage implementation of this algorithm for $G = \mathbb{F}_2^n$ and $H = \mathbb{F}_2^n$ can be found online[1].

Every finite commutative group can be decomposed as a direct sum of cyclic group of prime-power order. For large prime powers, it would be useful to have a more efficient algorithm than direct matrix-multiplication. Such algorithms exist for the Fourier transformation. This is left as future work.

4.4.2 Large functions with structure

For a typical nonlinear layer consisting of the parallel applications of several small S-boxes, Corollary 4.1 (1) can be used to efficiently evaluate individual coordinates of the quasidifferential transition matrix.

Modular additions (between values in $\mathbb{F}_2^n$) are another popular component in many block ciphers. Since these additions usually operate on too many bits, the method from Section 4.4.1 is not applicable. Nevertheless, modular additions are sufficiently structured so that the coordinates of the quasidifferential transition matrix can be computed using a relatively simple formula. This formula will be derived in Section 8.2.2.

[1] `https://github.com/TimBeyne/quasidifferential-trails`

4.5 Differential characteristics

In Section 4.3.3, it was shown that the fixed-key probability of a differential is equal to the sum of the correlations of its quasidifferential trails. However, it is often convenient to decompose the probability of differentials as the sum of the probabilities of its differential characteristics. Specifically, for $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$, the probability of a differential with input difference a_1 and output difference a_{r+1} equals

$$\Pr[\mathsf{F}(\boldsymbol{x}_1 + a_1) = \mathsf{F}(\boldsymbol{x}_1) + a_{r+1}] = \sum_{a_2,\ldots,a_r} \Pr\left[\bigwedge_{i=1}^{r} \mathsf{F}_i(\boldsymbol{x}_i + a_i) = \mathsf{F}_i(\boldsymbol{x}_i) + a_{i+1}\right],$$

with $\boldsymbol{x}_i = \mathsf{F}_{i-1}(\boldsymbol{x}_{i-1})$ for $i = 2, \ldots, r$ and $\boldsymbol{x}_1$ uniform random.

In Section 4.5.1, it is shown how the fixed-key probability of a differential characteristic can be computed using quasidifferential trails. Hence, the decomposition into differential characteristics can be maintained when working with quasidifferential trails. Section 4.5.2 analyzes the key-dependence of the differential characteristic used by Biham and Shamir [58] in their attack on DES. This serves as a first application of quasidifferential trails and is mainly intended to illustrate how the technique works in practice. More advanced applications are given in Chapter 8. Finally, some additional properties of quasidifferential trails are discussed in Section 4.5.3.

4.5.1 Exact probabilities from quasidifferential trails

Corollary 4.2 implies that the sum of the correlations of all quasidifferential trails with input and output character-difference pairs $\varpi_1 = (\mathbb{1}, a_1)$ and $\varpi_{r+1} = (\mathbb{1}, a_{r+1})$ respectively, is equal to the exact probability of the differential with input difference a_1 and output difference a_{r+1}. Theorem 4.4 shows that quasidifferential trails can also be used to compute the probability of a characteristic, likewise by summing their correlations.

Theorem 4.4. *Let* $\mathsf{F} = \mathsf{F}_r \circ \ldots \circ \mathsf{F}_1$. *The probability of a characteristic* $(a_1, \ldots, a_{r+1})$ *is equal to the sum of the correlations of all quasidifferential trails with the same intermediate differences as the characteristic:*

$$\Pr\left[\bigwedge_{i=1}^{r} \mathsf{F}_i(\boldsymbol{x}_i + a_i) = \mathsf{F}_i(\boldsymbol{x}_i) + a_{i+1}\right] = \sum_{\chi_2,\ldots,\chi_r} \prod_{i=1}^{r} D^{\mathsf{F}_i}_{(\chi_{i+1},a_{i+1}),(\chi_i,a_i)},$$

where $\chi_1 = \mathbb{1}$, $\chi_{r+1} = \mathbb{1}$ *and* $\boldsymbol{x}_i = \mathsf{F}_{i-1}(\boldsymbol{x}_{i-1})$ *for* $i = 2, \ldots, r$ *with* $\boldsymbol{x}_1$ *uniform random.*

Proof. Suppose that $\mathsf{F}_i : G_i \to G_{i+1}$. Substituting (4.1) in the right-hand side above yields

$$\prod_{i=1}^{r} D^{\mathsf{F}_i}_{(\chi_{i+1},a_{i+1}),(\chi_i,a_i)} = \frac{1}{\prod_{i=1}^{r}|G_i|} \sum_{\substack{x_1,\dots,x_r \\ \mathsf{F}(x_i+a_i)=\mathsf{F}(x_i)+a_{i+1}}} \prod_{i=1}^{r} \overline{\chi_{i+1}(\mathsf{F}_i(x_i))}\chi_i(x_i)\,.$$

Summing over $\chi_2,\dots,\chi_r$ then results in the equation

$$\begin{aligned}
&\sum_{\chi_2,\dots,\chi_r} \prod_{i=1}^{r} D^{\mathsf{F}_i}_{(\chi_{i+1},a_{i+1}),(\chi_i,a_i)} \\
&= \frac{1}{\prod_{i=1}^{r}|G_i|} \sum_{\substack{x_1,\dots,x_r \\ \mathsf{F}(x_i+a_i)=\mathsf{F}(x_i)+a_{i+1}}} \prod_{i=1}^{r} \sum_{\chi_i} \overline{\chi_i(\mathsf{F}_i(x_i))}\chi_i(x_{i+1}) \\
&= \frac{1}{|G_1|} \sum_{\substack{x_1,\dots,x_r \\ \mathsf{F}(x_i+a_i)=\mathsf{F}(x_i)+a_{i+1}}} \prod_{i=1}^{r} \delta_{x_{i+1}}(\mathsf{F}_i(x_i))\,.
\end{aligned}$$

Writing the right-hand side in terms of probabilities gives desired the result. □

Theorem 4.4 can also be obtained using the following intuitive argument, illustrated in Figure 4.2. Let $\mathsf{G} = (\mathsf{F}_1, \mathsf{F}_2 \circ \mathsf{F}_1, \dots, \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1)$. A differential for G with input difference a_1 and output difference $(a_2,\dots,a_{r+1})$ is equivalent to a characteristic for $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$ with intermediate differences $a_2,\dots,a_r$. For the linear function $\mathsf{L}(x) = (x,x)$, Theorem 4.2 (3) yields $D^{\mathsf{L}}_{(\chi,b),(\psi,a)} = \delta_\psi(\chi_1\chi_2)\,\delta_{b_1}(a)\delta_{b_2}(a)$ with $\chi(x,y) = \chi_1(x)\,\chi_2(y)$ and $b = (b_1,b_2)$. Hence, all trails through G with nonzero correlation are of the form shown in Figure 4.2 and the result follows from Corollary 4.2.

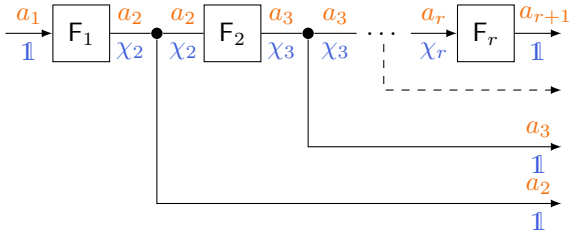


Figure 4.2: Quasidifferential trail through the function G. Differences are indicated in orange (above), masks in blue (below).

4.5.2 Differential cryptanalysis of DES

As a first example of quasidifferential trails and Theorem 4.4, we consider the effect of key-dependence on the differential cryptanalysis of DES by Biham and Shamir [58, 60]. The example in this section is particularly simple, but more advanced applications will be discussed in Chapter 8.

Recall from Example 1.2 that the differential cryptanalysis of DES is based on an iterative characteristic of the form shown in Figure 4.3. There exist two differences that achieve the same maximal average probability of approximately $2^{-7.87}$. For simplicity (the other case is similar), we will consider the difference $a = \texttt{0x19600000}$. The key-dependence of this characteristic was already noted by Knudsen [180, §5], who explained it using an argument specific to DES. Below, it will be shown that the general methodology of quasidifferential trails automatically provides a simple explanation.

The round function F_k of DES consists of a linear expansion function $\mathsf{E} : \mathbb{F}_2^{32} \to \mathbb{F}_2^{48}$, which duplicates certain bits, followed by the key addition and a nonlinear layer S consisting of eight 6-bit to 4-bit S-boxes. Finally, the S-box layer is followed by a bit-permutation P. The key-averaged probability of the characteristic in Figure 4.3 is easily computed from the difference-distribution tables of the first three S-boxes: $14/64 \times 8/64 \times 10/64 = 1120/64^3$.

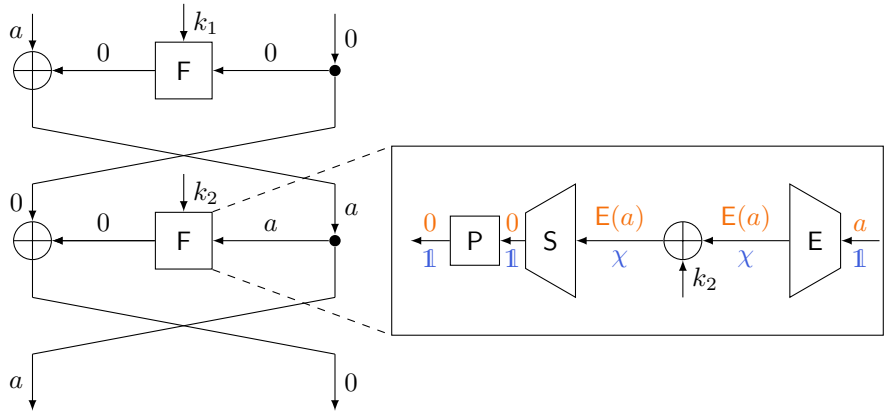


Figure 4.3: Iterative characteristic for two rounds of DES.

However, the structure of the round function of DES leads to one-round quasidifferential trails, as shown on the right side of Figure 4.3. In particular, since E is not surjective, there exist characters $\chi \neq \mathbb{1}_{\mathbb{F}_2^{48}}$ such that $\chi \circ \mathsf{E} = \mathbb{1}_{\mathbb{F}_2^{32}}$. For the difference a mentioned above, there exists one such quasidifferential trail with $\chi(x) = (-1)^{u^\mathsf{T} x}$ for $u = \texttt{0x001400000000}$. The correlation of this

trail can be computed from the quasidifferential transition matrices of the first three S-boxes and equals $\chi(k_2)\,14/64 \times -8/64 \times 6/64 = -\chi(k_2)\,672/64^3$. It follows that a full description of the probability of the characteristic over $2r$ rounds is given by

$$\prod_{i=1}^{r} \left(\frac{1120}{64^3} - (-1)^{k_{2i,12}+k_{2i,14}} \frac{672}{64^3} \right).$$

Although for every two rounds only two trails are especially important, these trails can be combined in many ways. In particular, the expression above is equivalent to a sum over 2^r quasidifferential trails. This is a typical way in which a relatively small local effect can result in significant variations in the overall probability of a characteristic.

Due to the above, the probability of the 13-round differential used in the differential attack of Biham and Shamir [60] is roughly 17 times larger for one in 64 keys and more than 244 times smaller than the average probability for an equal number of keys, as previously observed by Knudsen [180].

It is natural to wonder if there exist other quasidifferential trails with large absolute correlation. For example, a more general three-round effect can occur when $\chi \circ \mathsf{E} \neq \mathbb{1}_{\mathbb{F}_2^{32}}$. However, most quasidifferential trails activating four or less additional S-boxes have correlation zero because the correlation of a linear approximation with input mask $\mathtt{1}$ or $\mathtt{32}$ and output mask $\mathtt{1}, \mathtt{2}, \mathtt{4}$ or $\mathtt{8}$ is zero for all S-boxes. This follows from the fact that the S-boxes are permutations when the first and last input bits are fixed. It can be checked that the best three-round quasidifferential trail of this type has absolute correlation at most $2^{-19.41}$.

4.5.3 Further properties of quasidifferential trails

As discussed in Section 4.3.2 and Example 4.1 in particular, the coordinates of D^{F} can be interpreted as the correlations of linear approximations between the input and output values for the right pairs of a differential. Quasidifferential trails provide a way to connect such approximations through a sequence of functions.

Since $|D^{\mathsf{F}}_{(\chi,b),(\psi,a)}|$ never exceeds the probability of the differential (a, b), the quasidifferential trails with the highest correlation tend to have nontrivial characters in only a few rounds. We refer to these quasidifferential trails as 'local'. In general, the best quasidifferential trails typically activate as few S-boxes as possible. An S-box is active if either the output character or the input difference is nontrivial.

Quasidifferential trails with absolute correlation equal to the correlation of the corresponding differential trail are of particular interest. They correspond to deterministic linear relations on the intermediate values of right pairs. Perhaps surprisingly, many ciphers admit such quasidifferential trails. One reason for this is that the differentials of many popular S-boxes are *planar* [105]. That is, the right values form an affine space (in general, a coset of a subgroup). Propagating this affine space is the basis of the plateau characteristics approach [105], but is difficult to do for more than two rounds. Theorem 4.5 is related to these quasidifferential trails and will be useful in Chapter 8.

Theorem 4.5. *For a function* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$ *and a characteristic* $a_1, \ldots, a_{r+1}$ *with correlation* p *(as quasidifferential trail), it holds that:*

(1) If $(\chi_1, a_1), \ldots, (\chi_{r+1}, a_{r+1})$ *is a quasidifferential trail with correlation* $\lambda\, p$ *where* $|\lambda| = 1$*, then for every quasidifferential trail* $(\psi_1, a_1), \ldots, (\psi_{r+1}, a_{r+1})$ *with correlation* c*, the correlation of the quasidifferential trail* $(\chi_1\psi_1, a_1), \ldots, (\chi_{r+1}\psi_{r+1}, a_{r+1})$ *is* $\lambda\, c$*.*

(2) If the correlations of any number of quasidifferential trails with differences $a_1, \ldots, a_{r+1}$ *and absolute correlation* p *sum to zero, then the probability of the characteristic* $a_1, \ldots, a_{r+1}$ *is zero.*

Proof. By Theorem 4.4 the second property follows from the first one, since it implies that the set of all quasidifferential trails can be partitioned into subsets whose correlations sum to zero. For the first property, note that the correlation of the quasidifferential trail $(\chi_1, a_1), \ldots, (\chi_{r+1}, a_{r+1})$ equals λp if and only if $D^{\mathsf{F}_i}_{(\chi_{i+1}, a_{i+1}),(\chi_i, a_i)} = \lambda_i\, D^{\mathsf{F}_i}_{(\mathbb{1}, a_{i+1}),(\mathbb{1}, a_i)}$ for $i = 1, \ldots, r$ and $\prod_{i=1}^{r} \lambda_i = \lambda$.

By (4.1), this implies that $\chi_{i+1}(\mathsf{F}_i(x)) = \lambda_i\, \chi_i(x)$ for all x such that $\mathsf{F}_i(x + a_i) = \mathsf{F}_i(x) + a_{i+1}$. Hence, again by (4.1), the correlation of the i^{th} transition of the quasidifferential trail $(\chi_1\psi_1, a_1), \ldots, (\chi_{r+1}\psi_{r+1}, a_{r+1})$ is multiplied by a factor λ_i. The result then follows from $\lambda = \prod_{i=1}^{r} \lambda_i$. □

As mentioned above, plateau characteristics are related to the special quasidifferential trails considered in Theorem 4.5. The following example works this out explicitly.

Example 4.3 (Plateau characteristics)**.** Let (a, b, c) be a differential characteristic for a function $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ such that $\mathsf{F} = \mathsf{F}_2 \circ \mathsf{Add}_k \circ \mathsf{F}_1$ with $\mathsf{Add}_k(x) = x + k$ a key-addition function. If the sets $A = \mathsf{F}_1(\{x \in \mathbb{F}_2^n \mid \mathsf{F}_1(x + a) = \mathsf{F}_1(x) + b\})$ and $B = \{x \in \mathbb{F}_2^n \mid \mathsf{F}_2(x + b) = \mathsf{F}_2(x) + c\}$ are affine spaces, then (a, b, c) is called a plateau characteristic [105].

Let $A = p_A + V_A$ and $B = p_B + V_B$ with V_A and V_B vector spaces. The main result of plateau characteristics [105, Theorem 1] is that the fixed-key probability of the characteristic (a, b, c) is equal to

$$\begin{cases} 2^{\dim(V_A \cap V_B) - n} & \text{if } k \in p_A + p_B + V_A + V_B\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

From the point of view of quasidifferential trails, the fact that A and B are affine implies that

$$D^{\mathsf{F}_1}_{(\chi,b),(\mathbb{1},a)} = \chi(p_A)\mathbb{1}_{V_A^1}(\chi)/|V_A^1|$$

$$D^{\mathsf{F}_2}_{(\mathbb{1},c),(\chi,b)} = \chi(p_B)\mathbb{1}_{V_B^1}(\chi)/|V_B^1|\,.$$

Hence, there is a quasidifferential trail $((\mathbb{1}, a), (\chi, b), (\mathbb{1}, c))$ with nonzero correlation for every character χ in $V_A^1 \cap V_B^1$. In fact, the absolute correlation of these quasidifferential trails is equal to the key-averaged probability of the characteristic. By Theorem 4.4, the probability of the differential equals

$$\frac{1}{|V_A^1|}\frac{1}{|V_B^1|}\sum_{\chi \in V_A^1 \cap V_B^1} \chi(k + p_A + p_B) = \mathbb{1}_{V_A+V_B}(k + p_A + p_B)/|V_A^1 + V_B^1|\,,$$

since $V_A^1 \cap V_B^1 = (V_A + V_B)^1$. Furthermore, since $|V_A^1 + V_B^1| = 2^{n-\dim(V_A \cap V_B)}$, the result is the same as the probability obtained using the plateau characteristic approach. Since every character corresponds to a particular linear combination, quasidifferential trails correspond to the equations that are satisfied by the right values whereas plateau characteristics are based on the values themselves. If most S-boxes are inactive, then this is useful because a small number of equations can characterize a large set of right values. ▷

Finally, we briefly consider how strong quasidifferential trails can exist for a large number of rounds of a cipher. For every active S-box in a quasidifferential trail that is not active in the corresponding characteristic, the correlation of the trail contains a factor equal to the correlation of an ordinary linear approximation over that S-box. These approximations never have absolute correlation one, since the S-box is a nonlinear function. Hence, to avoid activating too many differentially inactive S-boxes, the masks of the quasidifferential trail should follow the differences as closely as possible. By Theorem 4.2 (3), one structural property that makes this more likely in ciphers defined over $\mathbb{F}_2^n$ is if the linear layer L satisfies $\mathsf{L}^{-1} = \mathsf{L}^{\mathsf{T}}$. Such 'self-dual' linear layers, including all bit-permutations, are in common use. Insights such as these can be used by designers to avoid strong key-dependency or, should they choose to do so, to amplify key-dependent effects on purpose.

5

Integral cryptanalysis

This chapter develops an extension of integral cryptanalysis by applying Chapter 2 to cryptanalytic properties that are defined over an extension field of the p-adic numbers. This leads to a one-dimensional theory that differs from linear and differential cryptanalysis in two important ways. Specifically, the p-adic metric is non-Archimidean as opposed to Archimidean and the preferred basis is not specifically chosen to simplify the key-addition operation. The theory relies on the assumption that the primitive is defined over a finite commutative inverse monoid. For primitives defined over $\mathbb{F}_2$, reducing the one-dimensional theory modulo two yields the contemporary description of integral cryptanalysis based on division trails.

The results in this chapter have not yet been published. The modulo-two reduction of the theory for $\mathbb{F}_2^n$ was worked out by Michiel Verbauwhede in his master's thesis [276], which was jointly supervised with Chaoyun Li. This chapter does not go into the practial details of the modulo-two reduced case. I thank Wouter Castryck for discussions about p-adic estimates of character sums, which play a role in Section 5.4.

5.1 Introduction

The theory of integral cryptanalysis has come a long way since the introduction of the Square attack by Daemen, Knudsen and Rijmen at FSE 1997 [102]. As described by Knudsen and Wagner [184], the original approach was based on the propagation of a set of plaintexts with some constant parts and some saturated parts through a cipher, ultimately resulting in a set of ciphertexts with a part that is saturated or sums to zero.

A different approach to obtain such 'zero-sums' was introduced by Knudsen [181] a few years earlier. It is based on the algebraic observation that the d^{th}-order derivative of a function of degree d is constant. Higher-order derivatives were first introduced by Lai [190], although he suggested to use them for statistical attacks similar to differential cryptanalysis. Knudsen's algebraic point of view encouraged a large body of work on degree bounds [77, 78, 85].

A partial consolidation of the two aforementioned approaches was realized by Todo, with the introduction of the division property [264]. A multiset $S \subseteq \mathbb{F}_2^n$ has the division property of order k if all polynomials of degree strictly less than k sum to zero on S. In particular, the division property of order two is equivalent to the zero-sum property. The division property of order n expresses that every value in S occurs an equal number of times modulo two. It corresponds to the saturated property if and only if S is a nonempty *set*.

Refinements of the conventional division property were introduced by Todo and Morii [268] (bit-based division property) and Boura and Canteaut [79] (parity sets). The most precise formulation of the division property, the three-subset division property without unknown subset [158], can be understood as a method to compute the coefficients of monomials in the algebraic normal form of a product of one or more coordinates of a vectorial Boolean function F. The underlying assumption is that F is a composition of several simpler functions: $\mathsf{F} = \mathsf{F}_r \circ \cdots \mathsf{F}_2 \circ \mathsf{F}_1$. The method can be reformulated in purely algebraic terms using the concept of monomial trails [166].

This chapter starts from the observation that the theory of integral cryptanalysis is incomplete. In particular, the division property and its refinements *do not* encompass the saturation property completely because all counting is performed modulo two. From a practical viewpoint, this means that the analysis may lead to the conclusion that a bit just sums to zero, even if it actually satisfies a stronger property such as being saturated. This is especially important for key-recovery attacks because zero-sums allows for comparatively little filtering of candidate keys. The same limitation is reflected theoretically: although division and monomial trails already suggest the existence of a one-dimensional theory along the lines of Chapter 2, the field $\mathbb{F}_2$ only admits the trivial absolute value function. Hence, such a theory would appear to be less rich than either linear or differential cryptanalysis.

To overcome these limitations, this chapter proposes a general one-dimensional theory based on the action of a finite commutative inverse monoid. To obtain a generalization of integral cryptanalysis, the monoid is instantiated as $\mathbb{F}_q^n$ with its coordinate-wise product. This can be motivated by the observation that the multiplicative characters of $\mathbb{F}_q^n$ are (lifted) monomials. For ordinary integral cryptanalysis, $q = 2$. However, the characters of a monoid are not orthogonal unless the monoid is a group. This leads to difficulties if the theory is constructed over the complex numbers. In particular, it is not possible to make the change-of-basis transformation length-preserving. It turns out that these issues can be avoided by working over an algebraic extension of the p-adic numbers for an appropriate choice of the prime number p.

In Section 5.3, a pair of dual bases diagonalizing the action of an arbitrary

commutative inverse monoid is explicitly constructed. This is worked out over the p-adic numbers, although most of the results generalize to other fields. Representing the pushforward operator of a function relative to the new basis yields a new transition matrix that is called the *ultrametric transition matrix*. The corresponding trails are called ultrametric trails, and satisfy all of the usual properties. However, the fact that the p-adic absolute value function satisfies the ultrametric triangle inequality has important consequences for the dominant trail approximation (Corollary 5.4).

Section 5.4 specializes the theory to multiplicative monoids of the form $\mathbb{F}_q^n$, with q a power of p. The properties of ultrametric transition matrices are investigated. Theorem 5.8 shows that the modulo-p reduction of their coordinates is related to the algebraic normal form. Theorem 5.9 determines the ultrametric transition matrix for addition by a constant, and Theorem 5.10 bounds the absolute values of the coordinates of ultrametric transition matrices in terms of the degree. Sections 5.4.3 and 5.4.4 explore approximations and trails, and show that the theory reduces modulo two to ordinary integral cryptanalysis when $q = 2$. This also leads to natural generalizations of notions such as parity sets (Example 5.7) and the conventional division property (Definition 5.3). The case $q > 2$ is increasingly relevant due the development of arithmetization-oriented primitives, which are discussed in Chapter 10. Although it will be left as future work, the techniques from this chapter could be used to improve some of the attacks in Chapter 10.

As a proof of concept, Section 5.5 revisits the integral cryptanalysis of PRESENT. Based on the analysis of ultrametric trails, it is shown that the integral distinguishers exhibited by Boura and Canteaut [79] are stronger than previously believed. This example also demonstrates that the analysis of ultrametric trails can be automated using off-the-shelf SMT solvers.

5.2 Mathematical setting

Throughout this chapter, $\mathsf{F} : M \to N$ is a function between finite commutative inverse monoids M and N. Section 5.2.1 reviews some results about the structure of monoids that will be important in the remainder of this chapter.

Like for linear and differential cryptanalysis, the properties that are considered in integral cryptanalysis only involve the integers or more generally the rational numbers. However, unlike in Chapters 3 and 4, distances are measured using the p-adic absolute value function. Furthermore, it is mathematically more convenient to work over the metric completion of $\mathbb{Q}$. As explained in Section 5.2.2 below, this completion is the field of p-adic numbers $\mathbb{Q}_p$. In fact, depending on

the monoids M and N, it may be necessary to work over an algebraic extension of $\mathbb{Q}_p$. For convenience, one can work in an algebraically closed extension $\mathbb{C}_p$ of $\mathbb{Q}_p$, analogous to the complex numbers. The field $\mathbb{C}_p$ is defined in Section 5.2.2.

Since $\mathbb{C}_p$ is a field with an absolute value function (see Section 5.2.2), there is a corresponding definition of norm on the vector spaces $\mathbb{C}_p^M$ and $\mathbb{C}_p^N$. The choice of norm is briefly discussed in Section 5.2.3. Finally, Section 5.2.4 introduces and motivates the monoid action that will lead to a preferred basis in Section 5.3.

5.2.1 Monoids

Recall from Section 2.4.4 that a commutative monoid M is *inverse* if for every x in M, there exists a y such that $x^2y = x$. The typical way in which such monoids come up is as the multiplicative structure of a finite commutative algebra over a field, as in the following example. To emphasize this, all monoids will be denoted multiplicatively.

Example 5.1. The vector space $\mathbb{F}_q^n$ is an algebra with multiplication defined by $(x_1, \ldots, x_n)(y_1, \ldots, y_n) = (x_1y_1, \ldots, x_ny_n)$. With this multiplication operation, the set $\mathbb{F}_q^n$ is a commutative inverse monoid. ▷

Let $E_M = \{e \in M \mid e^2 = e\}$ be the set of idempotents of M. Every commutative inverse monoid M is partially ordered with $x \leq y$ if and only if $x \in yE_M$. The inverse property ensures that $\leq$ is reflexive, see [256, Proposition 3.9].

Example 5.2. The set of idempotents of the multiplicative monoid $\mathbb{F}_q^n$ is equal to $\{0, 1\}^n$. The partial order on $\{0, 1\}^n$ is equivalent to the inclusion order on the subsets of an n-element set, *i.e.* it is a Boolean algebra. ▷

The partial order on M will be used in Section 5.3.1 to determine the inverse of the change-of-basis transformation. Specifically, this result will rely on a generalization of the inclusion-exclusion principle, known as Möbius inversion. The systematic investigation of this combinatorial technique was initiated by Rota [245]. Theorem 5.1 can be extended to infinite partially ordered sets as long as each interval is finite. In particular, the term *Möbius function* comes from the case where P is the ring of integers ordered by divisibility.

Theorem 5.1 (Möbius inversion [245])**.** *Let P be a finite partially ordered set and k a field. There exists a function $\mu : P \times P \to k$ such that if two functions $f : P \to k$ and $g : P \to k$ satisfy*

$$g(x) = \sum_{\substack{y \in P \\ y \geq x}} f(y)\,,$$

then they also satisfy

$$f(x) = \sum_{\substack{y \in P \\ y \geq x}} \mu(x, y) g(y) \, .$$

The function μ is called the Möbius function of P and satisfies the recurrence relation $\mu(x, y) = -\sum_{x \leq z < y} \mu(x, z)$ with $\mu(x, x) = 1$.

For any x in M, there exists a positive integer n such that x^n is idempotent. This can be shown by elementary means, see for instance [256, Corollary 1.2]. The unique idempotent corresponding to x is denoted by x^ω. The following lemma shows that the order of x and y is largely determined by the order of x^ω and y^ω.

Lemma 5.1. *Let M be a finite commutative inverse monoid. For all x and y in M, it holds that $x \leq y$ if and only if $x^\omega \leq y^\omega$ and $x = x^\omega y$.*

Proof. If $x \leq y$, then there exists an idempotent element e in E_M such that $x = ey$. Hence, $x^\omega = ey^\omega$ or equivalently $x^\omega \leq y^\omega$. Furthermore, $x = ey = ey^\omega y = x^\omega y$. Conversely, let $e = x^\omega$, then $ey = x^\omega y = x$ whence $x \leq y$. □

Lemma 5.1 implies that $\mu(x, y) = \mu(x^\omega, y^\omega)\, \delta_x(x^\omega y)$. Indeed,

$$\mu(x, y) = -\delta_x(x^\omega y) \sum_{x^\omega \leq z^\omega < y^\omega} \mu(x^\omega y, z^\omega y)$$

Since $\mu(x, x) = \mu(x^\omega, x^\omega)$, the result follows by recursively applying this equality.

Example 5.3. Consider $M = \mathbb{F}_q^n$ as in Examples 5.1 and 5.2. Since E_M is a Boolean algebra, $\mu(x^\omega, y^\omega) = (-1)^{\mathsf{wt}(x) - \mathsf{wt}(y)}$ if $x^\omega \leq y^\omega$ with $\mathsf{wt}(x)$ the Hamming weight of x [245]. Hence, if $x \leq y$, then

$$\mu(x, y) = (-1)^{\mathsf{wt}(x) - \mathsf{wt}(y)} \, .$$

Otherwise, $\mu(x, y) = 0$. ▷

For each idempotent element e, the set eM is a monoid with identity e and the same operation as M. Below, $(eM)^\times$ denotes the group of units of this monoid. The following result shows that the structure of every finite commutative inverse monoid is determined by the subgroups corresponding to its idempotent elements. This will be useful to compute the characters of monoids.

Lemma 5.2. *Let M be a finite commutative inverse monoid. Every x in M is contained in precisely one group of the form $(eM)^\times$ with e in E_M, namely for $e = x^\omega$. Furthermore, $ey \in (eM)^\times$ for all y in M such that $y^\omega \geq e$.*

Proof. Let n be a positive integer such that $x^n = e$. This implies that $x^{n-1}x = e$ and hence $x \in (eM)^\times$. Furthermore, if $x \in (fM)^\times$, then $f = x^\omega = e$. Hence, $\{(eM)^\times \mid e \in E_M\}$ is indeed a partition of M.

If $e \leq y^\omega$, then $ey^\omega = e$ because e and y^ω are idempotent elements. Hence, if z is the inverse of y in $(y^\omega M)^\times$, then $(ey)(ez) = ey^\omega = e$. It follows that ey is an element of $(eM)^\times$. □

5.2.2 The field of p-adic numbers

Recall from Example 2.1 that the p-adic absolute value of x in $\mathbb{Q}$ is $|x|_p = p^{-e}$ for $x = p^e\, a/b$ with a and b indivisible by p. One approach to defining the p-adic numbers has already been informally introduced above: it is the metric completion of $\mathbb{Q}$ with respect to the absolute value function $|\cdot|_p$. For a detailed exposition of this approach, the reader is referred to the first chapter of Koblitz's book on p-adic numbers [185].

Alternatively, the p-adic numbers can be defined in a more algebraic way. Define the p-adic integers $\mathbb{Z}_p$ as the set of all infinite sequences $(x_1, x_2, \ldots)$ with x_i in $\mathbb{Z}/p^i\mathbb{Z}$ and $x_i \equiv x_{i+1} \pmod{p^i}$. That is, $\mathbb{Z}_p = \varprojlim \mathbb{Z}/p^n\mathbb{Z}$. If addition and multiplication are defined component-wise, then the p-adic integers $\mathbb{Z}_p$ form an integral domain with a unique maximal ideal $p\mathbb{Z}_p$. The field of fractions of $\mathbb{Z}_p$ is the field of p-adic numbers $\mathbb{Q}_p$.

The construction of the preferred basis in Section 5.3.1 requires roots of unity in $\mathbb{Q}_p$. The following result shows that the $(p-1)^{\text{th}}$ roots of unity exist in $\mathbb{Q}_p$. The proof is based on lifting the solutions of $x^{p-1} \equiv 1 \pmod{p}$ to $\mathbb{Z}_p$ using Hensel's lemma.

Theorem 5.2 (p-adic roots of unity)**.** *The only roots of unity in $\mathbb{Z}_2$ are ± 1. For every odd prime p, the only roots of unity in $\mathbb{Z}_p$ are the $(p-1)^{th}$ roots of unity. Furthermore, each root of unity is congruent to a unique integer in $\{1, \ldots, p-1\}$ modulo p.*

Due to Theorem 5.2, one can define a multiplicative function $\tau : \mathbb{F}_p \to \mathbb{Q}_p$ such that $\tau(x)$ is the unique solution of $\tau(x)^p = \tau(x)$ with $\tau(x) \equiv x \pmod{p}$. In particular, if $x \neq 0$, then $\tau(x)$ is the $(p-1)^{\text{th}}$ root of unity congruent to x. The function τ is called the Teichmüller character at p and $\tau(x)$ is called the Teichmüller representative of x.

If $\mathbb{Q}_p$ does not contain enough roots of unity, it may be necessary to work in a finite algebraic extension of $\mathbb{Q}_p$. For every such extension field k of degree

$d = [k : \mathbb{Q}_p]$, the p-adic absolute value function can be extended to k by

$$|x|_p = |N_{k/\mathbb{Q}_p}(x)|_p^{1/d}\,, \tag{5.1}$$

with $N_{k/\mathbb{Q}_p}$ the field norm. Indeed, the field norm is multiplicative.

An algebraic extension k of $\mathbb{Q}_p$ is called unramified if $|x|_p$ is an integer power of p for every x in k. The only unramified extension of $\mathbb{Q}_p$ of degree e is $\mathbb{Q}_p(\zeta)$, with ζ a primitive $(p^e-1)^{\text{th}}$ root of unity. A proof can be found in [185, §3]. The ring of integers of $\mathbb{Q}_p(\zeta)$ is $\mathbb{Z}_p[\zeta]$ and $\mathbb{Z}_p(\zeta)/(p)$ is called the residue field of $\mathbb{Q}_p(\zeta)$. It is the finite field $\mathbb{F}_p(\zeta)$ of order p^e. Hence, Teichmüller characters can be extended to $\mathbb{F}_p(\zeta)$. In particular, for x in $\mathbb{F}_p(\zeta)$, let $\tau(x)$ be the unique $(p^e-1)^{\text{th}}$ root of unity in $\mathbb{Q}_p(\zeta)$ such that $\tau(x) \equiv x \pmod{p}$.

To ensure that enough roots of unity are always available, it can be convenient to work with the algebraic closure $\overline{\mathbb{Q}}_p$ of $\mathbb{Q}_p$. The absolute value function of x in $\overline{\mathbb{Q}}_p$ can be defined as in (5.1) with $k = \mathbb{Q}_p(x)$. The metric completion of the algebraic closure is denoted by $\mathbb{C}_p$ and is itself algebraically closed. The field $\mathbb{C}_p$ is analogous to the complex numbers. Technically, its elements are equivalence classes of Cauchy sequences in $\overline{\mathbb{Q}}_p$. The absolute value function of a Cauchy sequence $(x_1, x_2, \ldots)$ is defined as $\lim_{n\to\infty} |x_n|_p$.

5.2.3 Motivation for the norm

The norm of a function u in $\mathbb{C}_p^M$ will be defined as

$$\|u\|_M = \max_{x\in M} |u(x)|_p\,.$$

This choice enables a proper comparison of cryptanalytic properties that involve counting modulo powers of p.

Unlike the Euclidean norm that was used in Chapters 3 and 4, the maximum norm does not come from an inner product on $\mathbb{C}_p^M$. In fact, there is no suitable notion of inner products over $\mathbb{C}_p$. Hence, choosing an isomorphism between the dual space $(\mathbb{C}_p^M)^\vee$ and $\mathbb{C}_p^M$ would be arbitrary and will consequently be avoided.

Nevertheless, the $\|\cdot\|_M$-norm has a self-duality property similar to the Euclidean norm. Specifically, the dual norm $\|\cdot\|_M^\vee$ satisfies

$$\|v\|_M^\vee = \max_{x\in M} |v(\delta_x)|_p\,.$$

5.2.4 Monoid action

The monoid action that will be considered is the multiplication $x \mapsto mx$ for m in M. As in Chapters 3 and 4, this action extends to $(\mathbb{C}_p^M)^\vee$ by

$$\big((T^m)^\vee v\big)(u) = v(x \mapsto u(mx))\,,$$

with v in $(\mathbb{C}_p^M)^\vee$ and u in $\mathbb{C}_p^M$. The corresponding action on $\mathbb{C}_p^M$ is given by

$$\big(T^m u\big)(x) = \sum_{\substack{y \in M \\ my = x}} u(y)\,.$$

If m has an inverse, then $(T^m u)(x) = u(m^{-1}x)$. Hence, unlike for group actions, the actions on $\mathbb{C}_p^M$ and $(\mathbb{C}_p^M)^\vee$ are markedly different.

As usual, diagonalizing the action $x \mapsto mx$ is an attempt to minimize the number of trails. However, unlike in Chapters 3 and 4, the monoid action does not usually correspond to key-addition. This may seem to be problematic, as it implies that the key-addition operation is likely to result in a large number of key-dependent trails. Nevertheless, it will be shown in Section 5.3.3 that this problem is partially avoided because the absolute value function is ultrametric. Instead, there is a strong algebraic motivation for simplifying multiplications. Every function on $\mathbb{F}_q^n$ can be expressed as a multivariate polynomial over $\mathbb{F}_q$. The character basis leads to sparse representations when the number of monomials is small.

5.3 One-dimensional theory

In this section, the one-dimensional theory from Chapter 2 is applied in the setting that was introduced in Section 5.2 above. This leads to a theory of trails that relies only on the combination of a p-adic extension field and the action of a finite commutative inverse monoid. The monoid is specialized to $\mathbb{F}_q^n$ in Section 5.4, but the results in this section are more general. This leads to a more complete understanding of the $\mathbb{F}_q^n$ case, and makes it easier to discuss the general consequences of the differences between group actions versus monoid actions and Archimidean versus non-Archimidean absolute value functions.

5.3.1 Character basis

It was shown in Section 2.4.4 that the characters of a commutative inverse monoid M themselves form a commutative inverse monoid $\widehat{M}$. Throughout

this chapter, the characters are constructed relative to the field $\mathbb{C}_p$. It is worth keeping in mind that several properties of group characters, including self-duality and orthogonality (Theorem 3.4 (1) and (2)), do not hold for monoid characters.

For each character χ in the dual monoid $\widehat{M}$, one can define a corresponding linear functional b^χ by $b^\chi(\delta_x) = \chi(x)$. By Theorem 2.8, the vectors b^χ form a basis for $\mathbb{C}_p^M$ that simultaneously diagonalizes $(T^m)^\vee$ for all m in M.

The dual basis of $\{b^\chi \mid \chi \in \widehat{M}\}$ will be denoted by $\{b_\chi \mid \chi \in \widehat{M}\}$, with $b^\chi(b_\psi) = \delta_\chi(\psi)$. This basis likewise consists of simultaneous eigenvectors for T^m with m in M. Following Definition 2.10, the corresponding change-of-basis transformation $\mathscr{U}_M : \mathbb{C}_p^M \to \mathbb{C}_p^{\widehat{M}}$ is defined by $\mathscr{U}_M\, b_\chi = \delta_\chi$. An alternative definition, comparable to Definitions 3.5 and 4.2, is given below.

Definition 5.1. Let p be a prime and M a finite commutative inverse monoid. The ultrametric change-of-basis transformation $\mathscr{U}_M : \mathbb{C}_p^M \to \mathbb{C}_p^{\widehat{M}}$ is defined by

$$\big(\mathscr{U}_M f\big)(\chi) = \sum_{x \in M} \chi(x) f(x)\,.$$

The corresponding dual change-of-basis transformation is $\mathscr{U}_M^{-\vee}$, and satisfies $\mathscr{U}_M^{-\vee}\, b^\chi = \delta^\chi$. An explicit definition of $\mathscr{U}_M^{-\vee}$ is more difficult, as it depends on and implies a closed-form formula for b_χ. An inverse formula for Definition 5.1 (and hence a closed-form expression for b_χ) is derived below. The analysis is based on Steinberg's approach to the decomposition of representations of inverse monoids [256, §9.3].

The following result constructs the characters of M from the characters of the groups $(eM)^\times$ with identity e in E_M. It is a special case of the Clifford-Munn-Ponizovskiĭ correspondence for the representations of inverse monoids [256, §5.2].

Theorem 5.3 (Clifford-Munn-Ponizovskiĭ for characters)**.** *Let M be a finite commutative inverse monoid. Every character $\chi : M \to \mathbb{C}_p$ of M is an extension of a group character $\psi : (eM)^\times \to \mathbb{C}_p$ for some idempotent element e. Specifically,*

$$\chi(x) = \begin{cases} \psi(ex) & \text{if } e \leq x^\omega\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Furthermore, the characters obtained for different choices of ψ are distinct.

Proof. The function χ is well-defined by Lemma 5.2, from which it also follows that $\sum_{e \in E_M} |(eM)^\times| = |M|$. The multiplicativity of χ follows from the fact that ψ is a homomorphism, and from the fact that $e \leq f_1 f_2$ is equivalent to $e \leq f_1$ and $e \leq f_2$ when e, f_1 and f_2 are idempotent. The distinctness property follows

from the distinctness of group characters and the fact that group characters are nonzero everywhere. □

Theorem 5.3 can be used to obtain an inverse formula for Definition 5.1.

Theorem 5.4 (Inversion formula)**.** *Let p be a prime and M a finite commutative inverse monoid with Möbius function μ. The inverse of $\mathscr{U}_M : \mathbb{C}_p^M \to \mathbb{C}_p^{\widehat{M}}$ from Definition 5.1 satisfies*

$$\big(\mathscr{U}_M^{-1} f\big)(x) = \sum_{\substack{y \in M \\ y \geq x}} \mu(x, y) \frac{1}{|G_y|} \sum_{\chi \in \widehat{G}_y} f(\chi)/\chi(y)\,,$$

where $G_y = (y^\omega M)^\times$ is the maximal group with identity y^ω containing y.

Proof. By Definition 5.1, the function $\mathscr{U}_M^{-1} f$ satisfies

$$f(\chi) = \sum_{x \in M} \chi(x) \big(\mathscr{U}_M^{-1} f\big)(x)\,.$$

This relation can be inverted by combining the properties of group characters with Möbius inversion. In particular, let y be an element of M and let $e = y^\omega$. Furthermore, let $G_y = (eM)^\times$. For every such y, one has

$$\sum_{\chi \in \widehat{G}_y} f(\chi)/\chi(y) = \sum_{x \in M} \big(\mathscr{U}_M^{-1} f\big)(x) \sum_{\chi \in \widehat{G}_y} \chi(x)/\chi(y)\,.$$

The right-hand side can be worked out as follows:

$$\sum_{x \in M} \big(\mathscr{U}_M^{-1} f\big)(x) \sum_{\chi \in \widehat{G}_y} \chi(x)/\chi(y) = \sum_{\substack{x \in M \\ x^\omega \geq e}} \big(\mathscr{U}_M^{-1} f\big)(x) \sum_{\chi \in \widehat{G}_y} \chi(exy^{-1})\,,$$

with x^ω the smallest idempotent power of x and y^{-1} the inverse of y in G_y. Since $\sum_{\chi \in \widehat{G}_y} \chi(z) = |G_y|\, \delta_e(z)$ for any z in G_y, the right-hand side is equal to

$$|G_y| \sum_{\substack{x \in M \\ x^\omega \geq e}} \big(\mathscr{U}_M^{-1} f\big)(x) \delta_y(ex) = |G_y| \sum_{\substack{x \in M \\ x \geq y}} \big(\mathscr{U}_M^{-1} f\big)(x)\,.$$

The equality above follows from Lemma 5.1: one has $y \leq x$ if and only if $e \leq x^\omega$ and $ex = ey$. Dividing both sides by $|G_y|$ yields

$$\sum_{\substack{x \in M \\ x \geq y}} \big(\mathscr{U}_M^{-1} f\big)(x) = \frac{1}{|G_y|} \sum_{\chi \in \widehat{G}_y} f(\chi)/\chi(y)\,.$$

The result then follows from the Möbius inversion formula (Theorem 5.1). □

Applying Theorem 5.4 to $f = \delta_\chi$ yields a closed-form expression for b_χ. In particular, if χ extends a character of $G_e = (eM)^\times$, then

$$b_\chi(x) = \frac{1}{|G_e|} \sum_{\substack{y \geq x \\ y^\omega = e}} \mu(x,y)/\chi(y) = \frac{\mu(x^\omega, e)}{|G_e|} \sum_{\substack{y \in G_e \\ x^\omega y = x}} 1/\chi(y)\,. \tag{5.2}$$

The following corollary of Theorem 5.4 should be compared to the orthogonality of the Fourier transformation on groups (Definition 3.5). Although there is no inner product on $\mathbb{C}_p^M$, the transformation $\mathscr{U}_M$ is norm-preserving just like the Fourier transformation.

Corollary 5.1. *Let M be a finite commutative inverse monoid. If p is a prime not dividing $|(eM)^\times|$ for every idempotent e, then $\mathscr{U}_M$ is an isometry. That is, for all f in $\mathbb{C}_p^M$, it holds that $\|\mathscr{U}_M f\|_{\widehat{M}} = \|f\|_M$.*

Proof. Let χ be a character of M. By Theorem 5.3, $\chi(x)$ is either zero or a root of unity, such that $|\chi(x)|_p \leq 1$. Furthermore, it follows from Definition 2.20 and the ultrametric triangle inequality that $|(\mathscr{U}_M f)(\chi)|_p \leq \|f\|_M$. Hence, $\|\mathscr{U}_M f\|_{\widehat{M}} \leq \|f\|_M$.

Conversely, by Theorem 5.4 and because the Möbius function is integer-valued,

$$|f(x)|_p = |(\mathscr{U}_M^{-1}\mathscr{U}_M f)(x)|_p \leq \|\mathscr{U}_M f\|_{\widehat{M}} \max_{e \in E_M} |1/|(eM)^\times||_p\,.$$

Since p does not divide $|(eM)^\times|$, it follows that $\|f\|_M \leq \|\mathscr{U}_M f\|_{\widehat{M}}$. □

The analogue of Corollary 5.1 over the complex numbers and for the Euclidean norm is only true if M is a group. If the prime p is appropriately chosen, then working over the p-adic numbers bypasses this issue. Length-preservation is useful in practice, since it implies that the principal correlations of an approximation can be computed directly with respect to the ultrametric basis.

5.3.2 Ultrametric transition matrices

Let $\mathsf{F} : M \to N$ be a function and p a prime satisfying the conditions in Corollary 5.1. Following Definition 2.11, the pushforward operator T^{F} can be expressed relative to the basis from Section 5.3.1. The matrix representation of the resulting operator will be called the *ultrametric transition matrix* of F.

Definition 5.2 (Ultrametric transition matrix)**.** Let $\mathsf{F} : M \to N$ be a function between finite commutative inverse monoids M and N. Let $A^{\mathsf{F}} : \mathbb{C}_p^M \to \mathbb{C}_p^N$

be the pushforward operator of F relative to the ultrametric basis. That is, $A^{\mathsf{F}} = \mathscr{U}_N \, T^{\mathsf{F}} \, \mathscr{U}_M^{-1}$

The forward ultrametric transition matrix of F is the coordinate representation of A^{F} with respect to the standard bases of $\mathbb{C}_p^{\widehat{M}}$ and $\mathbb{C}_p^{\widehat{N}}$. Likewise, the backward ultrametric transition matrix of F is the coordinate representation of $A^{\mathsf{F}^\vee}$ with respect to the standard bases of $(\mathbb{C}_p^{\widehat{M}})^\vee$ and $(\mathbb{C}_p^{\widehat{N}})^\vee$.

Ultrametric transition matrices satisfy the standard properties from Theorem 2.4. For completeness, these properties are reproduced in the following corollary.

Corollary 5.2 (Properties of ultrametric transition matrices)**.** *The ultrametric transition matrix A^{F} of $\mathsf{F} : M \to N$ has the following properties:*

(1) If $\mathsf{F}(x_1, \ldots, x_n) = (\mathsf{F}_1(x_1), \ldots, \mathsf{F}_n(x_n))$, then $A^{\mathsf{F}} = \bigotimes_{i=1}^n A^{\mathsf{F}_i}$.

(2) If $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$, then $A^{\mathsf{F}} = A^{\mathsf{F}_r} \cdots A^{\mathsf{F}_2} A^{\mathsf{F}_1}$.

The following result states some additional properties that are specific to ultrametric transition matrices. These properties should be compared to Theorems 3.5 and 4.2.

Theorem 5.5 (Properties of ultrametric transition matrices)**.** *The ultrametric transition matrix A^{F} of $\mathsf{F} : M \to N$ has the following properties:*

(1) If F is a bijection, then A^{F} is an isometry.

(2) If F is a monoid homomorphism, then $A^{\mathsf{F}}_{\chi,\psi} = \delta_{\chi \circ \mathsf{F}}(\psi)$.

(3) If $M = N$ and $\mathsf{F}(x) = mx$ for some constant m in M, then A^{F} is a diagonal matrix with $A^{\mathsf{F}}_{\chi,\chi} = \chi(m)$.

Proof. Property (1) follows from the fact that $\mathscr{U}_M^{-1}$, $\mathscr{U}_N$ and T^{F} (if F is a permutation) are isometries and consequently so is their composition.

The proof of (2) and (3) is identical to the proof of the corresponding properties in Theorem 3.5. Specifically, (2) follows from the fact that if F is a monoid homomorphism and χ a character of M, then $\chi \circ \mathsf{F}$ is a character of N. Property (3) is due to the definition of $\mathscr{U}_M$. Indeed, $T^{\mathsf{F}} = T^m$. □

5.3.3 Approximations and trails

The properties considered in this section are of the form (U, V) with $U = \mathrm{Span}\{b_\chi\}$ and $V = \mathrm{Span}\{b^\psi\}$ with χ in $\widehat{M}$ and ψ in $\widehat{N}$. The cryptanalytic significance of such properties will be discussed in Section 5.4.3.

Following the template of Section 2.4.3, the composition property of ultrametric transition matrices leads to a one-dimensional theory of trails. In particular, one has the following result. A similar result can be given for backward trails. One should keep in mind that backward propagation is not the same as forward propagation through the inverse.

Corollary 5.3 (Sum of ultrametric trails, *cf.* Theorem 2.6)**.** *If* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$, *then the correlation* $A^{\mathsf{F}}_{\chi_{r+1},\chi_1}$ *is equal to*

$$A^{\mathsf{F}}_{\chi_{r+1},\chi_1} = \sum_{\chi_2,\ldots,\chi_r} \prod_{i=1}^{r} A^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} ,$$

where the sum ranges over all intermediate monoid characters.

Although Corollary 5.3 is standard, the fact that A^{F} is defined over a non-Archimidean field has important implications. In particular, Corollary 5.4 shows that the ultrametric triangle inequality can be used to bound the error term in the dominant trail approximation without relying on heuristics.

Corollary 5.4 (Dominant trail approximation *cf.* Corollary 2.1)**.** *Let* $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_2 \circ \mathsf{F}_1$. *For all subsets* Λ *of the set* Ω *of all trails from* χ_1 *to* χ_{r+1},

$$\left| A^{\mathsf{F}}_{\chi_{r+1},\chi_1} - \sum_{\chi\in\Lambda} \prod_{i=1}^{r} A^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} \right|_p \leq \left| \sum_{\chi\in\Omega\setminus\Lambda} \prod_{i=1}^{r} A^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} \right|_p \leq \max_{\chi\in\Omega\setminus\Lambda} \prod_{i=1}^{r} |A^{\mathsf{F}_i}_{\chi_{i+1},\chi_i}|_p ,$$

with $\chi = (\chi_1, \ldots, \chi_{r+1})$.

An important difference with Chapters 3 and 4 is that the most common way to use Corollary 5.4 is with $\Lambda = \emptyset$. To obtain a meaningful result, it is then necessary to show that all trails have absolute correlation less than $1/p$. This can be interpreted as 'approximate' zero-correlation cryptanalysis. In contrast, small correlations are hard to exploit over an Archimidean field because the error term is generally difficult to bound. As Section 5.4 below explains, cube attacks [121] can be considered to be an example with nonempty Λ.

5.4 Integral cryptanalysis on $\mathbb{F}_q^n$

This section applies the one-dimensional theory from Section 5.3 to the multiplicative monoid $\mathbb{F}_q^n$, with q a power of the prime number p. The resulting theory is a strict generalization of ordinary integral cryptanalysis, in the sense

that the reduction modulo two of the theory for $\mathbb{F}_2^n$ yields familiar concepts such as parity sets and division trails. A similar reduction of the $\mathbb{F}_q^n$-theory modulo p leads to natural generalizations of these concepts. The reduction is equivalent to ignoring all trails with absolute correlation lower than $1/p$.

5.4.1 Characters of $\mathbb{F}_q^n$

The characters of $\mathbb{F}_q^n$ can be obtained from Theorem 5.3. This leads to the following result. Recall from Section 5.2.2 that $\tau : \mathbb{F}_q \to \mathbb{C}_p$ denotes the Teichmüller character.

Theorem 5.6. *Every multiplicative character* $\chi : \mathbb{F}_q^n \to \mathbb{C}_p$ *is of the form*

$$\chi(x) = \tau(x^u) = \prod_{i=1}^{n} \tau(x_i^{u_i}) \,,$$

with $u_1, \ldots, u_n$ *in* $\{0, 1, \ldots, q-1\}$.

Proof. Since the Teichmüller character τ is a multiplicative function, every function $x \mapsto \tau(x^u)$ is a character of $\mathbb{F}_q^\times$. Choosing u in $\{1, 2, \ldots, q-1\}$ yields a complete set of characters for $\mathbb{F}_q^\times$. Furthermore, the only character of the trivial group $\{0\}$ is $x \mapsto 1 = \tau(x^0)$.

Recall from Example 5.2 that the set of idempotent elements of $\mathbb{F}_q^n$ is $\{0,1\}^n$. Hence, for every idempotent e, the group $(e\mathbb{F}_q^n)^\times$ is a direct product of n groups of the form $\{0\}$ or $\mathbb{F}_q^\times$. Hence, it follows from Theorem 5.3 that every character of $\mathbb{F}_q^n$ is of the form $x \mapsto \prod_{i=1}^n \tau(x_i^{u_i})$ with $u_1, \ldots, u_n$ in $\{0, 1, \ldots, q-1\}$. □

Theorem 5.6 immediately yields an explicit description of the dual basis functions b^χ. To obtain a similar characterization of the basis functions b_χ, the inversion formula from Theorem 5.4 can be used. The following additional notation is introduced for convenience. For a character $\chi : x \mapsto \tau(x^u)$ of $\mathbb{F}_q$, let $\chi^+ : \mathbb{F}_q \to \mathbb{C}_p$ be the function defined by

$$\chi^+(x) = \begin{cases} \delta_0(u) + (1-q)\,\delta_{q-1}(u) & \text{if } x = 0 \\ 1/\chi(x) & \text{otherwise}\,. \end{cases}$$

Furthermore, for a character $\chi = \chi_1 \otimes \cdots \otimes \chi_n$ of $\mathbb{F}_q^n$ with $\chi_1, \ldots, \chi_n$ characters of $\mathbb{F}_q$, let $\chi^+ = \chi_1^+ \otimes \cdots \otimes \chi_q^+$. With this notation, the following result holds.

Theorem 5.7. *For every character $\chi : x \mapsto \tau(x^u)$ of $\mathbb{F}_q^n$, the corresponding basisfunction b_χ satisfies (with $\mathsf{wt}(u)$ the Hamming weight of u)*

$$b_\chi(x) = \begin{cases} \chi^+(x)/(q-1)^{\mathsf{wt}(u)} & \text{if } x \in u\mathbb{F}_q^n \\ 0 & \text{otherwise}. \end{cases}$$

Proof. It suffices to prove the result for $\mathbb{F}_q$, since the result for $\mathbb{F}_q^n$ then follows by taking tensor products. By Theorem 5.4 and (5.2) in particular, if χ extends a character of $\mathbb{F}_q^\times$ (that is, $u \neq 0$), then

$$b_\chi(x) = \frac{(-1)^{\mathsf{wt}(x)-1}}{q-1} \sum_{\substack{y \in \mathbb{F}_q^\times \\ x^\omega y = x}} 1/\chi(y)\,.$$

If $x \neq 0$ then the sum reduces to $1/\chi(x)$. If $x = 0$, then by the orthogonality of group characters, the sum is equal to $(q-1)\delta_{q-1}(u)$. Hence, $b_\chi(x) = \chi^+(x)/(q-1)$.

If χ extends the character of the trivial group (that is, $u = 0$), then $b_\chi(0) = 1 = \chi^+(0)$ and $b_\chi(x) = 0$ for all nonzero x. □

5.4.2 Ultrametric transition matrices

Let $\mathsf{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be a function, $\psi : x \mapsto \tau(x^u)$ a character of $\mathbb{F}_q^n$, and χ a character of $\mathbb{F}_q^m$. By Theorem 5.7, the corresponding coordinate of the ultrametric transition matrix A^F satisfies

$$A^\mathsf{F}_{\chi,\psi} = \delta^\chi\big(A^\mathsf{F}\delta_\psi\big) = b^\chi\big(T^\mathsf{F} b_\psi\big) = \frac{1}{(q-1)^{\mathsf{wt}(u)}} \sum_{x \in u\mathbb{F}_q^n} \chi(\mathsf{F}(x))\psi^+(x)\,. \qquad (5.3)$$

This expression will be useful to prove several properties of A^F.

Example 5.4 (Numerical normal form)**.** Let $q = 2$, $\psi(x) = \tau(x^u)$, and $\chi(x) = \tau(x^v)$. Since $\psi^+(x) = (-1)^{\mathsf{wt}(u)+\mathsf{wt}(x)}$ for x in $u\mathbb{F}_2^n$, expression (5.3) simplifies to

$$A^\mathsf{F}_{\chi,\psi} = (-1)^{\mathsf{wt}(u)} \sum_{x \in u\mathbb{F}_2^n} (-1)^{\mathsf{wt}(x)}\, \tau(\mathsf{F}^v(x))\,,$$

where $\tau(\mathsf{F}^v(x))$ is simply the integer representation of $\mathsf{F}^v(x)$ in $\{0,1\}$. Hence, $A^\mathsf{F}_{\chi,\psi}$ is the coefficient corresponding to x^u in the *numerical normal form* of the Boolean function F^v. The numerical normal form expresses a Boolean function as a multivariate integer polynomial. It was introduced in the Boolean functions

literature by Carlet and Guillot [87]. The matrix A^{F} can be considered to be an extension of the numerical normal form to vectorial Boolean functions. It is not necessary to consider 2-adic coefficients because $1 \in \mathbb{Q}$. For primes $p > 3$, the $(p-1)^{\text{th}}$ roots of unity exist in $\mathbb{Q}_p$ but not in $\mathbb{Q}$. ▷

By the Chinese remainder theorem, every function from $\mathbb{F}_q^n$ to $\mathbb{F}_q$ can be represented as a unique polynomial in $\mathbb{F}_q[x_1, \ldots, x_n]/(x_1^q - x_1, \ldots, x_n^q - x_n)$. For $q = 2$, this is called the *algebraic normal form*. Below, the same terminology will be used for the general case.

Theorem 5.8 (Reduction to algebraic normal form)**.** *Let* $\mathsf{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ *be a function. For all multiplicative characters* $\psi : x \mapsto \tau(x^u)$ *of* $\mathbb{F}_q^n$ *and* $\chi : x \mapsto \tau(x^v)$ *of* $\mathbb{F}_q^m$*, the coordinate* $A^{\mathsf{F}}_{\chi,\psi}$ *is congruent modulo* p *to the coefficient of* x^u *in the algebraic normal form of* F^v*.*

Proof. By the definition of A^{F}, the functional $b^\chi \circ T^{\mathsf{F}}$ can be decomposed as

$$b^\chi \circ T^{\mathsf{F}} = \sum_{\psi \in \widehat{\mathbb{F}_q^n}} A^{\mathsf{F}}_{\chi,\psi}\, b^\psi \,.$$

Evaluating in δ_x and reducing modulo p yields

$$\mathsf{F}^v(x) \equiv \sum_{u \in \{0,\ldots,q-1\}^n} A^{\mathsf{F}}_{\chi,\psi_u}\, x^u \pmod{p} \,.$$

where $\psi_u(x) = \tau(x^u)$. This is the algebraic normal form of F^v. □

For $q = 2$, the modulo-two reduction of A^{F} plays a central role in ordinary integral cryptanalysis and for the bit-based division property in particular. It was used by Boura and Canteaut [79] in the form of a table[1] to describe the propagation of parity sets. The point of view that this table is a matrix-representation of the pushforward operator of F over the residue field $\mathbb{F}_2$ was worked out by Michiel Verbauwhede in his master's thesis [276].

There is an efficient algorithm to calculate A^{F} when n is much larger than q. Specifically, because $\mathscr{U}_{\mathbb{F}_q^n} = \mathscr{U}_{\mathbb{F}_q}^{\otimes n}$, there is a fast algorithm to apply $\mathscr{U}_{\mathbb{F}_q^n}$. For $q = 2$, the modulo-2 reduction of this algorithm is well-known. For large q, further improvements are likely possible using algorithms inspired by the fast Fourier transform on $\mathbb{F}_q^\times$. This is left as future work.

A direct calculation of A^{F} is often difficult, but can sometimes be avoided for special functions such as translations and low-degree polynomials in general.

[1] Although their analysis uses another table that includes the effect of key-additions.

When relying on the dominant trail approximation (Corollary 5.4), it is often sufficient to know the p-adic absolute values of the coordinates of A^{F}. Below, upper bounds on these absolute values are derived.

In the following results, the p-weight of an integer x is the sum of its base-p digits. It will be denoted by $\mathsf{wt}_p(x)$. Similarly, the p-weight of an exponent u in $\{0, 1, \ldots, q-1\}^n$ is defined as $\mathsf{wt}_p(u) = \sum_{i=1}^{n} \mathsf{wt}_p(u_i)$.

If F is a translation, then the sums in (5.3) are related to Jacobi sums. This leads to the following theorem, which gives the exact absolute values of the coordinates of A^{F}. Extending this result to functions $\mathsf{F} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is a straightforward application of Corollary 5.2 (1).

Theorem 5.9 (Translation)**.** *Let $\mathsf{F} : \mathbb{F}_q \to \mathbb{F}_q$ be defined by $\mathsf{F}(x) = x + t$ with t a nonzero constant in $\mathbb{F}_q$ and let $\psi : x \mapsto \tau(x^u)$ and $\chi : x \mapsto \tau(x^v)$ be multiplicative characters of $\mathbb{F}_q$. If $v \neq 0$ or $u = 0$, then*

$$\operatorname{ord}_p\left(A^{\mathsf{F}}_{\chi,\psi}\right) = \frac{\mathsf{wt}_p(w) + \mathsf{wt}_p(u) - \mathsf{wt}_p(v)}{p-1},$$

where $w \in \{0, \ldots, q-2\}$ and $w \equiv v - u \pmod{q-1}$. Otherwise, $A^{\mathsf{F}}_{\chi,\psi} = 0$.

Proof. If $u = 0$, then the result is trivial. For $u \neq 0$, (5.3) yields

$$A^{\mathsf{F}}_{\chi,\psi} = -\chi(t)\delta_{q-1}(u) + \frac{1}{q-1}\sum_{x \in \mathbb{F}_q^\times} \chi(x+t)/\psi(x)\,.$$

If $u = q-1$ and $v \neq 0$, then $A^{\mathsf{F}}_{\chi,\psi} = -q\chi(t)/(q-1) + \delta_{q-1}(v)$ by standard properties of group characters. Since $\mathsf{wt}_p(w) = \mathsf{wt}_p(v)$ and $\mathsf{wt}_p(u) = e(p-1)$ for $q = p^e$, the result follows. If $u = q-1$ and $v = 0$, then $A^{\mathsf{F}}_{\chi,\psi} = 0$. Hence, it can be assumed that $u \neq 0$ and $u \neq q-1$.

The absolute value of A^{F} satisfies

$$\left|A^{\mathsf{F}}_{\chi,\psi}\right|_p = \left|\sum_{x \in \mathbb{F}_q^\times} \chi(1-x)/\psi(x)\right|_p.$$

The sum on the right-hand side is a Jacobi sum. Let ω be an additive character of $\mathbb{F}_q$ and let $G(\psi) = \sum_{x \in \mathbb{F}_q^\times} \omega(x)/\psi(x)$ be the corresponding Gauss sum. Jacobi and Gauss sums satisfy the following relation if $\chi \neq \psi$ [192, §1.1, GS3]:

$$\left|A^{\mathsf{F}}_{\chi,\psi}\right|_p = \left|\frac{G(1/\chi)G(\psi)}{G(\psi/\chi)}\right|_p.$$

Furthermore, if $\chi = \psi$, then the p-adic absolute value of the Jacobi sum is equal to one. Stickelberger's theorem [192, §1.2, Theorem 2.1] implies that $\operatorname{ord}_p G(\psi) = \mathsf{wt}_p(u)/(p-1)$. Similarly, $\operatorname{ord}_p G(1/\chi) = \mathsf{wt}_p(q-1-v)/(p-1) = e - \mathsf{wt}_p(v)/(p-1)$ and $\operatorname{ord}_p G(\psi/\chi) = e - \mathsf{wt}_p(v-u)/(p-1)$. Hence, the result follows from

$$\operatorname{ord}_p\left(A^{\mathsf{F}}_{\chi,\psi}\right) = \operatorname{ord}_p G(1/\chi) + \operatorname{ord}_p G(\psi) - \operatorname{ord}_p G(\psi/\chi)\,.$$

Furthermore, if $\chi = \psi$, then the result agrees with $\operatorname{ord}_p\left(A^{\mathsf{F}}_{\chi,\psi}\right) = 1$. □

Example 5.5 (Translation)**.** For $q = 2$ and $\mathsf{F}(x) = x + t$, the matrix A^{F} satisfies

$$A^{\mathsf{F}} = \bigotimes_{i=1}^{n} \begin{bmatrix} 1 & 0 \\ t_i & (-1)^{t_i} \end{bmatrix}.$$

If $\psi(x) = x^u$ and $\chi(x) = x^v$, then Theorem 5.9 shows that $|A^{\mathsf{F}}_{\chi,\psi}|_2 = 1$ if $uv = u$ (equivalently, $u \le v$) and $A^{\mathsf{F}}_{\chi,\psi} = 0$ otherwise. ▷

The following result (Theorem 5.10) upper bounds the p-adic absolute value of the coordinates of A^{F} in terms of the degree of F. More precisely, the p-degree of a monomial x^u is equal to $\mathsf{wt}_p(u)$. The p-degree $\deg_p \mathsf{F}$ of F is the maximum p-degree of the monomials with nonzero coefficients in the algebraic normal forms of its coordinates. It is not difficult to see that $\deg_p \mathsf{F}$ is the degree of any polynomial representation of F over $\mathbb{F}_p$, which is independent of the choice of basis for $\mathbb{F}_q$ over $\mathbb{F}_p$.

Theorem 5.10. *Let* $\mathsf{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ *be a function. Let* $\psi : x \mapsto \tau(x^u)$ *and* $\chi : x \mapsto \tau(x^v)$ *be multiplicative characters of* $\mathbb{F}_q^n$ *and* $\mathbb{F}_q^m$ *respectively. If* $d \ge \deg_p \mathsf{F} \ne 0$*, then*

$$\operatorname{ord}_p\left(A^{\mathsf{F}}_{\chi,\psi}\right) \ge \left\lceil \frac{\mathsf{wt}_p(u) - \mathsf{wt}_p(v)d}{(p-1)\,d} \right\rceil.$$

The proof of Theorem 5.10 is based on the following result of Wan [281].

Theorem 5.11 (Wan [281, Theorem 4.1])**.** *Let* $\chi_1, \ldots, \chi_r : \mathbb{F}_q \to \mathbb{C}_p$ *be multiplicative characters of* $\mathbb{F}_q$ *and let* $\mathsf{F}_1, \ldots, \mathsf{F}_r : \mathbb{F}_q^n \to \mathbb{F}_q$ *be functions. If* $q = p^e$ *and* $\chi_i(x) = \tau(x^{u_i})$*, then*

$$\operatorname{ord}_p \sum_{x \in \mathbb{F}_q^n} \textstyle\prod_{i=1}^r \chi_i\big(\mathsf{F}_i(x)\big) \ge \left\lceil \frac{ne + \frac{1}{p-1}\sum_{i=1}^r \mathsf{wt}_p(u_i) d_i}{\max_i d_i} \right\rceil,$$

where $d_i = \deg_p \mathsf{F}_i$ *for all* $i = 1, \ldots, r$.

Proof of Theorem 5.10. By (5.3) and because $|q-1|_p = 1$, it holds that

$$\left|A^{\mathsf{F}}_{\chi,\psi}\right|_p = \left| \sum_{x \,\in\, u\mathbb{F}_q^n} \chi(\mathsf{F}(x))\psi^+(x) \right|_p .$$

Define $\bar{u}$ by $\bar{u}_i = q-1-u_i$ for $i = 1,\ldots,n$ and let $I = \{1 \le i \le n \mid \bar{u}_i = 0\}$. By the definition of ψ^+, it holds that $\psi^+(x) = \tau(x^{\bar{u}}) \prod_{i\in I} \big(1 - q\,\delta_0(x_i)\big)$ for x in $u\mathbb{F}_q^n$. Hence,

$$\left|A^{\mathsf{F}}_{\chi,\psi}\right|_p = \left| \sum_{J\subseteq I} q^{|J|} \sum_{x \,\in\, u\mathbb{F}_q^n} \tau(\mathsf{F}^v(x)x^{\bar{u}}) \right|_p \le \max_{J\subseteq I} q^{-|J|} \left| \sum_{\substack{x\in u\mathbb{F}_q^n \\ \forall i\,\in\, J\colon\, x_i = 0}} \tau\big(\mathsf{F}^v(x)x^{\bar{u}}\big) \right|_p .$$

The inequality above follows from the ultrametric triangle inequality. For any set J, let $\mathsf{G}_J : \mathbb{F}_q^{\mathsf{wt}(u)-|J|} \to \mathbb{F}_q^m$ be the function obtained from F by setting all variables x_i with i in J or $u_i = 0$ equal to zero. There exists a w in $\{0,1,\ldots,q-1\}^{\mathsf{wt}(u)}$ with $\mathsf{wt}_p(w) = \mathsf{wt}_p(\bar{u})$ such that

$$\left|A^{\mathsf{F}}_{\chi,\psi}\right|_p \le \max_{J\subseteq I} q^{-|J|} \left| \sum_{\substack{x\in \mathbb{F}_q^l \\ l\,=\,\mathsf{wt}(u)-|J|}} \tau\big(\mathsf{G}_J^v(x)x^w\big) \right|_p .$$

By Theorem 5.11 the p-order of the sum on the right-hand side is at least

$$\left\lceil \frac{le(p-1) - d\,\mathsf{wt}_p(v) - \mathsf{wt}_p(w)}{(p-1)\,d} \right\rceil = \left\lceil -\frac{|J|e}{d} + \frac{\mathsf{wt}_p(u) - d\,\mathsf{wt}_p(v)}{(p-1)\,d} \right\rceil ,$$

where $q = p^e$. The equality follows from $\mathsf{wt}_p(w) = e(p-1)\,\mathsf{wt}(u) - \mathsf{wt}_p(u)$. Hence,

$$\mathrm{ord}_p\left(A^{\mathsf{F}}_{\chi,\psi}\right) \ge \min_{J\subseteq I} \left(|J|e + \left\lceil -\frac{|J|e}{d} + \frac{\mathsf{wt}_p(u) - d\,\mathsf{wt}_p(v)}{(p-1)\,d} \right\rceil \right) .$$

The result follows because $d \ge 1$. □

It can be verified that there exist functions F for which Theorem 5.10 is tight. Nevertheless, better bounds are often possible for specific functions. Theorem 5.9 is an example of such an improvement.

5.4.3 Approximations

As mentioned in Section 5.3.3, the one-dimensional cryptanalytic properties (U, V) corresponding to the basis functions are of the form $U = \mathrm{Span}\{b_\psi\}$

and $V = \mathrm{Span}\{b^\chi\}$. If $\psi(x) = \tau(x^u)$, then by Theorem 5.7 the function b_ψ corresponds to a weighting of the input set $u\mathbb{F}_q^n$. Specifically, a value x in $u\mathbb{F}_q^n$ is weighted by $\psi^+(x)/(q-1)^{\mathsf{wt}(u)}$. The linear functional b^χ corresponds to the evaluation of a monomial, followed by a lift of the result to $\mathbb{C}_p$.

Example 5.6 (Zero-sum and cube attacks)**.** The principal correlation of a property (U, V) as above is equal to $A^{\mathsf{F}}_{\chi,\psi}$. Suppose that $\chi(x) = \tau(x^v)$ and $\psi(x) = \tau(x^u)$. By Theorem 5.8, the reduction of $A^{\mathsf{F}}_{\chi,\psi}$ modulo p is equal to the coefficient of x^u in the algebraic normal form of F^v.

Since $\chi(x) \equiv x^v \pmod p$ and, for x in $u\mathbb{F}_q^n$, $\psi^+(x)/(q-1)^{\mathsf{wt}(u)} \equiv x^{\bar u} \pmod p$ with $\bar u_i = q - 1 - u_i$ for $i = 1, \ldots, n$, it holds that

$$\sum_{x \,\in\, u\mathbb{F}_q^n} \mathsf{F}^v(x)\, x^{\bar u} \equiv A^{\mathsf{F}}_{\chi,\psi} \pmod p\,.$$

Hence, given a theoretical estimate of $A^{\mathsf{F}}_{\chi,\psi}$ up to an absolute error not exceeding $1/p$, evaluating the sum above results in a distinguisher or allows extracting some key-information.

If $q = 2$, then $x^{\bar u} = 1$ for x in $u\mathbb{F}_2^n$ and one obtains the *cube attack* of Dinur and Shamir [121]. The same principle was used in the earlier *algebraic IV differential attack* of Vielhaber [277]. This attack recovers the coefficient of x^u in the algebraic normal form of F^v by summing over $u\mathbb{F}_2^n$. If $A^{\mathsf{F}}_{\chi,\psi} \equiv 0 \pmod p$, then (U, V) is called a *zero-sum property.* ▷

Examples of cube attacks and zero-sum properties with an input set that is not of the form $u\mathbb{F}_2^n$ can be found throughout the literature. For instance, the input set is often an arbitrary affine space of sufficiently large dimension. It is also possible that q is so large that no sets of the form $u\mathbb{F}_q^n$ can be used directly, but smaller subsets – such as submonoids of $u\mathbb{F}_q^n$ – may be useful.

Hence, one is often interested in cryptanalytic properties (U, V) with $U = \mathrm{Span}\{\mathbb{1}_S\}$ and $V = \mathrm{Span}\{b^\chi\}$, where S is a subset of $\mathbb{F}_q^n$ – not necessarily of the form $u\mathbb{F}_q^n$. Such properties can be analyzed by applying the change-of-basis map $\mathscr{U}_{\mathbb{F}_q^n}$ to $\mathbb{1}_S$. Before discussing this approach in detail, it is worthwhile to work this out explicitly for $q = 2$.

Example 5.7 (Parity sets)**.** If $q = 2$, then $\psi(x) = x^u$. Hence $\widehat{\mathbb{1}}_S = \mathscr{U}_{\mathbb{F}_2^n}\mathbb{1}_S$ satisfies

$$\widehat{\mathbb{1}}_S(\psi) = \sum_{x \in S} \psi(x) = \sum_{x \in S} \tau(x^u)\,.$$

Since the residue field is $\mathbb{F}_2$, the modulo-2 reduction of the function $\widehat{\mathbb{1}}_S$ is completely determined by the exponents of the monomials in its support:

$$\mathcal{U}(S) = \left\{ u \in \mathbb{F}_2^n \;\middle|\; \sum_{x \in S} x^u = 1 \right\}$$

The set $\mathcal{U}(S)$ was called the *parity set* of S by Boura and Canteaut [79]. The set S satisfies the conventional division property of order k if all elements of $\mathcal{U}(S)$ have Hamming weight at least k [79, Definition 3]. ▷

The following discussion generalizes to cases where the correlation of the property (U, V) is approximated by a nonzero value ('cube-like' properties), but for simplicity only the 'approximate zero-correlation' case will be considered here. In this case, the correlation $b^\chi(T^{\mathsf{F}}\mathbb{1}_S)$ is estimated to be zero. The cryptanalyst must determine a bound ε on the error of the estimation. If $\widehat{\mathbb{1}}_S = \mathscr{U}_{\mathbb{F}_q^n}\mathbb{1}_S$, then such a bound takes the form

$$\left|b^\chi\left(T^{\mathsf{F}}\mathbb{1}_S\right)\right|_p = \left| \sum_{\psi \in \widehat{\mathbb{F}}_q^n} A^{\mathsf{F}}_{\chi,\psi}\, \widehat{\mathbb{1}}_S(\psi) \right|_p \leq \varepsilon\,. \tag{5.4}$$

It is worth pointing out that ε upper bounds the principal correlation of the approximation (U, V), because $\|\mathbb{1}_S\|_{\mathbb{F}_q^n} = 1$ and $\|b^\chi\|^\vee_{\mathbb{F}_q^m} = 1$. For $\varepsilon = 1/p^l$, the property (U, V) is equivalent to

$$\sum_{x \in S} \tau(\mathsf{F}^v(x)) \equiv 0 \pmod{p^l}\,.$$

For $l = 1$, one recovers regular zero-sum properties because $\tau(x) \equiv x \pmod{p}$.

Example 5.8. It is worth emphasizing that $q = 2$ is a special case, because the Teichmüller lift is trivial: $\tau(0) = 0$ and $\tau(1) = 1$. Hence, the property (U, V) above is equivalent to $|\{x \in S \mid \mathsf{F}^v(x) = 1\}| \equiv 0 \pmod{2^l}$.

Note in particular that for $S = u\mathbb{F}_2^n$, this differs from the result obtained for $U = \mathrm{Span}\{\delta_\psi\}$ with $\psi(x) = \tau(x^u)$. In this case, a correlation below $1/2^l$ implies

$$\sum_{x \in u\mathbb{F}_2^n} (-1)^{\mathsf{wt}(x)} \tau(\mathsf{F}^v(x)) \equiv 0 \pmod{2^l}\,,$$

since $\psi^+(x) = (-1)^{\mathsf{wt}(u)+\mathsf{wt}(x)}$ for x in $u\mathbb{F}_2^n$. ▷

Based on Example 5.7, it can be argued that $\widehat{\mathbb{1}}_S$ is a natural generalization of the parity set of S. The following definition introduces an extension of the conventional division property, which is related to $\widehat{\mathbb{1}}_S$. Like for parity sets and the conventional division property, one should think of Definition 5.3 as a more compact but less precise characterization of S.

Definition 5.3 (p^l-division property)**.** A multiset S with elements from $\mathbb{F}_q^n$ satisfies the p^l-division property of order k if

$$\sum_{x \in S} \tau(x^u) \equiv 0 \pmod{p^l},$$

for all u in $\{0, 1, \ldots, q-1\}^n$ with $\mathsf{wt}_p(u) < k$.

For $p = 2$ and $l = 1$, Definition 5.3 reduces to the definition of the conventional division property as given by Todo [264, Definition 1]. By Definition 5.1, an equivalent characterization of Definition 5.3 is that $|\widehat{\mathbb{1}_S}(\psi)| < p^{-l}$ for all $\psi : x \mapsto \tau(x^u)$ with $\mathsf{wt}_p(u) < k$, with $\mathbb{1}_S(x)$ the number of occurrences of x in the multiset S.

Note that, unlike the conventional division property, Definition 5.3 does include the saturation property as a special case. The fact that Definition 5.3 depends only on the p-weight of u interacts well with Theorem 5.10. This will be illustrated in Section 5.4.4.

5.4.4 Trails

If $\mathsf{F} = \mathsf{F}_r \circ \cdots \circ \mathsf{F}_1$, then $A^{\mathsf{F}}_{\chi,\psi}$ is equal to the sum of the correlations of all ultrametric trails from ψ to χ (Corollary 5.3). In practice, the dominant trail approximation (Corollary 5.4) is used to estimate $A^{\mathsf{F}}_{\chi,\psi}$. For the approximate zero-correlation case, the set of dominant trails is chosen to be empty ($\Lambda = \emptyset$).

The first inequality in Corollary 5.4 theoretically allows one to determine the exact estimation error. However, this requires enumerating trails, which is often infeasible. Alternatively, the second inequality in Corollary 5.4 can be used. This yields an overestimate of the error, but is easier to compute. For ordinary integral cryptanalysis, both approaches have been explored. For the proof of concept in Section 5.5, only the second method will be used.

The remainder of this section shows how Corollary 5.4 specializes to existing techniques from the literature. A more detailed exposition on the residue-field can be found in the master's thesis of Michiel Verbauwhede [276].

For $q = 2$, Corollary 5.3 shows that $A^{\mathsf{F}}_{\chi_{r+1},\chi_1} \equiv \sum_{\chi \in \Lambda} \prod_{i=1}^{r} A^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} \pmod 2$ if and only if

$$\left| \left\{ \chi \in \Omega \setminus \Lambda \mid \left| \prod_{i=1}^{r} A^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} \right|_p = 1 \right\} \right| \equiv 0 \pmod 2.$$

Counting the number of trails corresponds to the propagation of the bit-based division property without unknown subset [158]. An equivalent algebraic

formulation was proposed by Hu *et al.* [166] under the name *monomial trails*. From the point of view of Corollaries 5.3 and 5.4, monomial trails correspond to backward trails (Definition 2.18). Indeed, the dual basis functions b^χ are essentially monomials: $b^\chi(\delta_x) = x^u$ if $\chi(x) = x^u$. The bit-based division property without unknown subset (equivalently, monomial trails) has the benefit of yielding exact bounds. However, it requires counting the number of trails (modulo two), which is often infeasible.

The second inequality in Corollary 5.4 shows that $A^{\mathsf{F}}_{\chi_{r+1},\chi_1} \equiv \sum_{\chi\in\Lambda} \prod_{i=1}^{r} A^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} \pmod 2$ if all trails χ in $\Omega \setminus \Lambda$ satisfy

$$\prod_{i=1}^{r} A^{\mathsf{F}_i}_{\chi_{i+1},\chi_i} \equiv 0 \pmod 2 .$$

That is, all trails in $\Omega \setminus \Lambda$ have a correlation lower than $1/2$. This corresponds to the propagation of parity sets. Indeed, the modulo-2 reduction of the basis function b_ψ with $\psi(x) = \tau(x^u)$ is the indicator function of the set $u\mathbb{F}_2^n$. It follows from Example 5.7 that the parity set of $u\mathbb{F}_2^n$ is $\{u\}$. The parity set is a set, and so does not account for potential multiplicities of its elements. Nevertheless, it is possible to define *parity multisets* to perform trail counting.

Finally, it is worth revisiting the common ancestor of the abovementioned techniques: the conventional division property [264]. Let $\psi : x \mapsto \tau(x^u)$ and $\chi : x \mapsto \tau(x^v)$. Theorem 5.10 shows that $|A^{\mathsf{F}}_{\psi,\chi}|_p \le 1/p$ whenever $\mathsf{wt}_p(v) < \mathsf{wt}_p(u)/\deg_p \mathsf{F}$. For S-box based ciphers, this suggests splitting the input space $\mathbb{F}_q^n$ as $\mathbb{F}_q^{n_1} \oplus \cdots \oplus \mathbb{F}_q^{n_l}$ and keeping track only of the p-weight of the exponents on each part. A typical application is given in the following example.

Example 5.9 (Degree bounds)**.** Let $\mathsf{F} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ with $\mathbb{F}_q^n = \mathbb{F}_q^m \oplus \cdots \oplus \mathbb{F}_q^m$ with $n = ml$. Suppose that $\mathsf{F} = \mathsf{L} \circ \mathsf{S}$ with $\mathsf{S}(x_1, \ldots, x_l) = (\mathsf{S}_1(x_1), \ldots, \mathsf{S}_l(x_l))$ an S-box layer of p-degree d and L affine over $\mathbb{F}_p$. If all correlation-one trails (χ_u, χ_v, χ_w) satisfy $\mathsf{wt}_p(u) \le t$, then $\deg_p \mathsf{F}^w \le t$.

If (χ_v, χ_w) has correlation one, then by Theorem 5.10, $\mathsf{wt}_p(v) \le \mathsf{wt}_p(w)$. If $\mathsf{wt}_p(w) \le me(p-1)$ and (χ_u, χ_v) has correlation one then $\mathsf{wt}_p(u) \le d\,\mathsf{wt}_p(v) \le d\,\mathsf{wt}_p(w)$. This yields the trivial bound $\deg_p \mathsf{F}^w \le d\,\mathsf{wt}_p(w)$. However, if $\mathsf{wt}_p(w) \ge me(p-1) = c$, then

$$\mathsf{wt}_p(u) \le d \sum_{i=1}^{l} \mathsf{wt}_p(v_i) \le d \left\lfloor \frac{\mathsf{wt}_p(w)}{c} \right\rfloor c + d \left(\mathsf{wt}_p(w) - c \left\lfloor \frac{\mathsf{wt}_p(w)}{c} \right\rfloor \right)$$

This yields an improved upper bound on $\deg_p \mathsf{F}^w$ and can result in a lower overall p-degree when F is iterated several times. ▷

For more complex functions, one can automate the analysis in Example 5.9. The conventional division property leads to a simplified analysis but a less accurate

error upper bound in Corollary 5.4. Using Definition 5.3, this approach can be generalized to take into account trails with correlation less than $1/p^2$.

5.5 Integral cryptanalysis of PRESENT

As a proof of concept, this chapter revisits several integral properties of reduced-round PRESENT [71]. PRESENT is a prototypical substitution-permutation network on $\mathbb{F}_2^{64}$: its round function consists of a layer of 4-bit S-boxes, a bit-permutation and a round key addition (see Figure 1.6). This allows for a simple automated analysis of ultrametric trails.

It should emphasized that integral attacks on PRESENT cover a small number of rounds compared to *e.g.* linear cryptanalysis, and this section does not attempt to overcome this fact. Instead, the goal of this section is to revisit the distinguishers of Boura and Canteaut [79] using the theory developed in the preceding sections. The results demonstrate that they can be improved, and it seems reasonable to expect similar improvements in other cases.

5.5.1 Modelling PRESENT

The linear layer $\mathsf{P} : \mathbb{F}_2^{64} \to \mathbb{F}_2^{64}$ of PRESENT is a bit-permutation, and hence a monoid homomorphism. Hence, by Theorem 5.5 (2), $A^{\mathsf{P}}_{\chi,\psi} = \delta_{\chi \circ \mathsf{P}}(\psi)$. The ultrametric transition matrix of the 4-bit S-box S can be computed using the algorithm from Section 5.4.2. This yields

$$A^{\mathsf{S}} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & -2 & -1 & 2 & 1 & -2 & 0 & 0 & -2 & 4 & 2 & -4 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & -1 & -1 & 1 & 0 & 2 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & -1 & -1 & 2 & 0 & 0 \\
1 & 0 & 0 & -1 & -1 & 0 & 0 & 2 & -1 & 1 & 1 & -1 & 2 & -1 & -2 & 0 \\
0 & 1 & 0 & -1 & 0 & -1 & 0 & 2 & 0 & -1 & 1 & 0 & 0 & 2 & -1 & -2 \\
0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & -1 & 0 \\
1 & -1 & -1 & 2 & 0 & 0 & 1 & -1 & -1 & 2 & 2 & -3 & 0 & -1 & -2 & 2 \\
0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 0 & 0 & 1 & -2 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & -2 & 0 & 1 & 1 & -3 & 0 & -1 & -2 & 4 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & -2 & 0 & 0 & -1 & 2 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 & 2 & 2 & -3 & 1 & -2 & -2 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -2 & 0 & -1 & -1 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1
\end{bmatrix}.$$

By Theorem 5.8, the modulo-2 reduction of the above matrix is the transition matrix for ordinary integral cryptanalysis. The propagation through the key-addition step is described by Theorem 5.9 and Example 5.5 in particular. Since only individual trails will be used for the analysis in this section, it is possible to merge the key-addition with the S-box layer. In this case, one can describe the worst-case propagation through the S-box layer with the table

$\gcd(A^{k_1}A^{\mathsf{S}}, \ldots, A^{k_{16}}A^{\mathsf{S}})$ where $k_1, \ldots, k_{16}$ are all possible values of a 4-bit key and gcd denotes the entrywise greatest common divisor. This table can be efficiently computed using an algorithm similar to the one used to compute A^{S}. It is analogous to the parity set propagation table introduced by Boura and Canteaut [79].

The model was implemented as a Satisfiability Modulo Theories (SMT) problem and solved using Boolector [222]. It is not strictly necessary to automate the solving process: with some effort, the same results can be obtained by hand using an 'approximate' variant of the miss-in-the-middle approach.

For given characters ψ and χ, one solves for trails between ψ and χ with successively lower correlation. If $|A^{\mathsf{F}}_{\chi,\psi}|_2 \leq 1/2^l$, then by Example 5.8

$$\sum_{x \in u\mathbb{F}_2^n} (-1)^{\mathsf{wt}(x)} \tau(\mathsf{F}^v(x)) \equiv 0 \pmod{2^l}.$$

For direct comparison with the results of Boura and Canteaut, it is also interesting to consider the unweighted input set $u\mathbb{F}_2^n$. As shown in Example 5.8, this corresponds to

$$|\{x \in u\mathbb{F}_2^n \mid \mathsf{F}^v(x) = 1\}| \equiv 0 \pmod{2^l}.$$

As shown by (5.4) in Section 5.4.3, it suffices to verify that $|A^{\mathsf{F}}_{\chi,\psi}|_2 \leq 2^{-l+\mathsf{wt}(\bar{u} \wedge w)}$ for all $\psi : x \mapsto \tau(x^w)$ to demonstrate this. Indeed, $(\mathscr{U}_{\mathbb{F}_2^n} \mathbb{1}_{u\mathbb{F}_2^n})(\psi) = 2^{\mathsf{wt}(\bar{u} \wedge w)}$. Importantly, this approach is only an approximation and better results can often be obtained by enumerating trails.

5.5.2 Results

The results for the first output bit ($v = \mathtt{0000000000000001}$) are listed in Table 5.1. Similar results can be obtained for other choices of v. The results in the table assume that the input state is b_ψ with $\psi(x) = \tau(x^u)$. The exponents u were chosen to match the input sets proposed by Boura and Canteaut [79, Table 3]. However, one should keep in mind that the properties in Table 5.1 rely on weighted inputs.

Table 5.1 demonstrates that for more than four rounds, the output bits satisfy a stronger property than a zero-sum. This results in distinguishers with a smaller false-positive rate. Since the choices of u in Table 5.1 ensure that all output bits have the zero-sum property, achieving a smaller false-positive rate may seem unimportant. However, for traditional key-recovery attacks, it is important to obtain a low false-positive rate given only a few bits of the output. With 2^k candidate keys, the difference between four zero-sum bits ($2^k/2^4$ remaining

candidates) and four bits with divisibility by 16 ($2^k/16^4$ remaining candidates) is important.

Table 5.1: Theoretical ($\varepsilon_{\mathsf{the}}$) and experimental ($\varepsilon_{\mathsf{exp}}$) estimates of $A^{\mathsf{F}}_{\chi,\psi}$ for $\psi : x \mapsto \tau(x^u)$ and $\chi : x \mapsto \tau(x^v)$ with $v =$ 0000000000000001, where F is r-round PRESENT.

r	u	$\log_2(\text{data})$	$\text{ord}_2(\varepsilon_{\mathsf{the}})$	$\text{ord}_2(\varepsilon_{\mathsf{exp}})$
4	000000000000000f	4	1	1
5	0000000000000fff0	12	3	3
6	00000000ffffffff	32	4	4
7	fffffffffffff000	52	4	—
8	fffffffffffffffe	63	2	—

The last column of Table 5.1 shows experimental results for the cases where this was feasible. These results represent the worst case over a random choice of several independent round keys. It is important to mention this, since it was observed that there exist large classes of weak keys with stronger properties. A proper analysis of this phenomenon needs to take into account multiple trails.

A straightforward application of Section 5.5.1 to the unweighted input sets $u\mathbb{F}_2^n$ results in the same theoretical bounds as in Table 5.1. Nevertheless, because the set $u\mathbb{F}_2^n$ can be propagated through (part of) the first and second S-box layers with correlation one, the unweighted inputs sets should result in better properties. However, the model from Section 5.5.1 only tracks the worst-case correlation of individual trails and hence cannot account for this effect.

Table 5.2: Theoretical ($\varepsilon_{\mathsf{the}}$) and experimental ($\varepsilon_{\mathsf{exp}}$) estimates of $b^{\chi}(T^{\mathsf{F}}\mathbb{1}_{u\mathbb{F}_2^n})$ for $\chi : x \mapsto \tau(x^v)$ with $v =$ 0000000000000001, where F is r-round PRESENT.

r	u	$\log_2(\text{data})$	$\text{ord}_2(\varepsilon_{\mathsf{the}})$	$\text{ord}_2(\varepsilon_{\mathsf{exp}})$
4	000000000000000f	4	2	3
5	0000000000000fff0	12	4	5
6	00000000ffffffff	32	7	14
7	fffffffffffff000	52	7	—
8	fffffffffffffffe	63	3	—

Slightly better bounds are obtained (without enumerating trails) by modifying the model to skip S-boxes in the first two rounds that are fully saturated, relying only on the fact that $\mathsf{S}(\mathbb{F}_2^4) = \mathbb{F}_2^4$. The results are shown in Table 5.2.

The last column of Table 5.2 shows that there is still a significant gap between the experimental results and the theoretical bounds obtained using the simplified model that uses only individual trails. This gap can be closed by enumerating all high-correlation trails. This is left as future work.

II

Applications

6

Block cipher invariants

Theorem 2.12 shows that over an algebraically closed field of characteristic zero, every forward invariant of a permutation is spanned by eigenvectors of its pushforward operator. Hence, as discussed in Section 3.5.1, invariants are spanned by eigenvectors of correlation matrices. Starting from this observation, this chapter obtains nonlinear invariants for reduced-round Midori-64 and MANTIS, and shows how these properties can be combined with integral cryptanalysis to obtain distinguishers for a larger number of rounds.

This chapter is based on the paper "Block cipher invariants as eigenvectors of correlation matrices" from Asiacrypt 2018 [37] and its extended version that appeared in Journal of Cryptology [39]. At the time of publication, the link between nonlinear invariants and eigenvectors of correlation matrices was new. Since this result is a straightforward consequence of Theorem 2.12, the applications to Midori-64 and MANTIS are the focus of this chapter instead.

6.1 Introduction

Block ciphers are an essential primitive for the construction of many cryptosystems. This leads to a natural desire to optimize them with respect to various application-dependent criteria. Examples include low-latency block ciphers such as PRINCE [76] and MANTIS [29], and the low-power design Midori [18]. Biryukov and Perrin [64] give a broad overview of such *lightweight* primitives.

A common design decision that often helps to reduce latency, energy consumption and other cost measures is the simplification of the key-schedule. This, along with other aspects of lightweight designs, led to the development of new cryptanalytic tools such as *invariant subspaces* [196] and *nonlinear invariants* [266]. These attacks are the subject of this chapter.

At Crypto 2017, it was shown by Beierle, Canteaut, Leander and Rotella that invariant attacks can often be averted by a careful choice of the round constants [27]. Their work, as well as the earlier work by Todo, Leander and

Sasaki on nonlinear invariants [266], invites several questions. This chapter is concerned with three related problems that arise in this context.

1. In their future work sections, Todo *et al.* [266] and Beierle *et al.* [27] both express the desire to generalize the nonlinear invariant attack. One can argue that a deeper theoretical understanding of block cipher invariants is helpful, if not essential, to achieve this goal.

2. One potential generalization is the existence of block cipher invariants which are not invariants under all of the round transformations. It is important to investigate this possibility, because such cases are not covered by the techniques introduced by Beierle *et al.* for choosing the round constants.

3. The previous problem leads to a third question: do such (generalized) invariants *only* impact the security of the cipher for a specific choice of the round constants? The results in this chapter suggest otherwise.

The first of the problems listed above was already addressed in Section 3.5.1: both invariant subspaces and nonlinear invariants are special cases of Definition 2.20. This immediately led to their characterization as eigenvectors of correlation matrices. Indeed, Theorem 2.12 shows that over an algebraically closed field of characteristic zero, every forward invariant is spanned by eigenvectors of its pushforward operator. In Section 6.2, this result is briefly reexamined for permutations on $\mathbb{F}_2^n$ and some additional detail is added.

The specifications of Midori-64 and MANTIS are reviewed in Section 6.3. Section 6.4 takes a closer look at the invariants of Midori-64, leading up to an example of an invariant of the type described in the second problem above. It will be shown in Section 6.4.3 that, with minor changes to the round constants, Midori-64 has an invariant which is not invariant under the round function. It applies to 2^{96} weak keys. Note that this is a significantly larger class of weak keys compared to previous work, *i.e.* 2^{32} for the invariant subspace attack of Guo *et al.* and 2^{64} for the nonlinear invariant attack of Todo *et al.* [266]. In fact, it will be demonstrated that the invariant discussed in Section 6.4.3 corresponds to a linear approximation with maximal correlation. This observation is an extreme example of the constructive interference phenomenon that was described in Section 3.6.2. It will be briefly discussed in Section 6.4.4. In Section 6.4.5, it is shown that the invariant from Section 6.4.3 is valid for an additional class of 2^{64} keys, leading to a total of $2^{96} + 2^{64}$ weak keys. This result is mainly interesting because it provides an example of an invariant which holds for four rounds, but not necessarily for fewer rounds. Hence, it serves as a further illustration of the second problem above.

Finally, Sections 6.5 and 6.6 address the third question listed above. That is, two cryptanalytic results are given to demonstrate that block cipher invariants may impact the security of a block cipher regardless of the choice of round constants.

In Section 6.5, a practical attack on 10 rounds of Midori-64 – for any choice of round constants – will be given. The attack applies to 2^{96} weak keys and requires roughly $1.25 \cdot 2^{21}$ chosen plaintexts. The computational cost is dominated by 2^{56} block cipher calls. Note that the data complexity and especially the computational cost to determine whether a weak key is used, are significantly lower. As discussed by Luykx, Mennink and Paterson [210] at Asiacrypt 2017, this has a significant impact on the multi-key security of the block cipher. A detailed analysis of the data complexity, supported by key-recovery experiments, is provided in Section 6.5.4.

Section 6.6 shows that the full key of MANTIS-4 [29] can be recovered given 346 chosen plaintexts. This attack works for all keys provided that a weak tweak is used. The number of weak tweaks is 2^{32} (out of 2^{64}). The computational cost of this attack is dominated by 2^{56} block cipher calls. If 346 chosen ciphertexts under a related tweak are additionally available, then the key can be recovered with a computational cost of 2^{18} block cipher calls. Section 6.6.3 supports the data complexity estimate by means of key-recovery experiments.

6.2 Invariants as eigenvectors of correlation matrices

Let $\mathsf{E}_k : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a block cipher with key k. Invariant sets and nonlinear invariants were already defined for functions on arbitrary finite commutative groups in Section 3.5.1, but it is worthwhile to review their original definitions.

Recall from Section 3.3.1, and Example 3.5 in particular, that the characters of $\mathbb{F}_2^n$ are given by $\chi_u : x \mapsto (-1)^{u^\mathsf{T} x}$ with u in $\mathbb{F}_2^n$. Throughout this chapter, the dual group of $\mathbb{F}_2^n$ will be identified with $\mathbb{F}_2^n$ through the isomorphism $u \mapsto \chi_u$. As discussed in Section 3.3.1 such an identification is arbitrary, and although it is theoretically inconvenient, it is often useful for applications.

The invariant subspace attack was introduced by Leander, Abdelraheem, AlKhzaimi and Zenner in the context of the PRINTcipher [196]. An invariant subspace of E_k is an affine subspace A of $\mathbb{F}_2^n$ such that

$$\mathsf{E}_k(A) = A\,. \tag{6.1}$$

The keys k for which (6.1) holds, are called weak keys. As shown in Example 3.8, this implies that the indicator function $\mathbb{1}_A$ of A is an eigenvector of T^{E_k} with

eigenvalue one. Equivalently, $C^{\mathsf{E}_k}\widehat{\mathbb{1}}_A = \widehat{\mathbb{1}}_A$ for the Fourier transformation $\widehat{\mathbb{1}}_A$ of $\mathbb{1}_A$. Up to the identification of $\mathbb{F}_2^n$ with its dual group, $\widehat{\mathbb{1}}_A : \mathbb{F}_2^n \to \mathbb{C}$ satisfies

$$\widehat{\mathbb{1}}_A(u)/|V| = (-1)^{a^\mathsf{T} u}\, \mathbb{1}_{V^\perp}(u)$$

with V the vector space such that $A = a + V$ for a in A.

At Asiacrypt 2016, Todo *et al.* introduced the nonlinear invariant attack as an extension of the invariant subspace attack [266]. A Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is called a nonlinear invariant for E_k if there exists a constant c in $\mathbb{F}_2$ such that for all x in $\mathbb{F}_2^n$,

$$f(x) + f(\mathsf{E}_k(x)) = c.$$

Importantly, the constant c may depend on the key k, but not on x. It was shown in Example 3.9 that this implies that $\chi \circ f$ is an eigenvector of T^{E_k} with eigenvalue $\chi(c)$, with χ a character of $\mathbb{F}_2$. Equivalently, the Fourier transformation of $\chi \circ f$ is an eigenvector of C^{E_k}. Concretely, for nontrivial χ,

$$\widehat{(\chi \circ f)}(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{u^\mathsf{T} x + f(x)}\,.$$

That is, the Walsh-Hadamard transformation of f is an eigenvector of C^{E_k}.

It was already explained that finding nonlinear invariants of an unstructured function $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is difficult. The method proposed by Todo *et al.* [266], based on the algebraic normal form, requires $\mathcal{O}(2^{3n})$ time. In light of Chapter 5, it is interesting to note that this method can be interpreted as a direct calculation of the invariant subspaces of the modulo-2 reduction of A^F. The algorithm from Section 3.5.1 requires $\mathcal{O}(n2^{2n})$ time, but this is still too much for most realistic block sizes. To obtain invariants, it is thus necessary to exploit structural properties of the block cipher.

The main structural property that has been exploited in previous work such as [154, 196, 266] is the existence of non-trivial *simultaneous* invariants for the linear layer and the nonlinear layer of a block cipher. It was shown in Theorem 3.7 that if a function is an eigenvector of C^k for several keys k, then this leads to strong restrictions on its support. Due to the identification between $\mathbb{F}_2^n$ and its dual, Theorem 3.7 can be reformulated as follows.

Corollary 6.1. *If v is an eigenvector of C^k for all k in a subset K of $\mathbb{F}_2^n$, then $\operatorname{supp} v \subseteq a + K^\perp$ for some a in $\mathbb{F}_2^n$. Furthermore, v is an eigenvector of C^k with corresponding eigenvalue $(-1)^{a^\mathsf{T} k}$ for all k in the span of K.*

Corollary 6.1 implies that the support of an invariant must be sparse if it is an eigenvector for many round key additions, and this is exasperated by the fact

that the support should be stable under the transpose of the linear layer. These conditions tend to contradict the nonlinearity of the S-box layer. In fact, this approach leads to the arguments of Beierle *et al.* [27] to rule out such invariants.

6.3 Midori-64 and MANTIS

A brief description of Midori-64 is given here. This information will be used extensively in Sections 6.4 and 6.5. Midori-64 is an iterated block cipher with a block size of 64 bits and a key length of 128 bits [18]. It operates on a 64-bit state, which can be represented as a 4×4 array of 4-bit *cells*. The round function consists of the operations SubCell ($\mathfrak{S}$), ShuffleCell (P), MixColumn ($\mathfrak{M}$) and a key addition layer. The overall structure is shown in Figure 6.1.

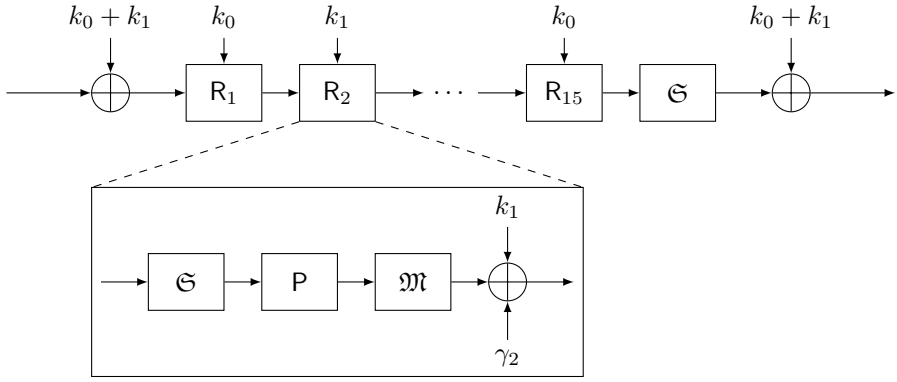


Figure 6.1: The overall structure and round function of Midori-64.

The SubCell ($\mathfrak{S}$) mapping applies a 4-bit S-box S to each cell of the state. The fact that the S-box is an involution will be used in Section 6.4. The algebraic normal form of $\mathsf{S}(x) = (\mathsf{S}_1(x), \mathsf{S}_2(x), \mathsf{S}_3(x), \mathsf{S}_4(x))$ is provided below. These expressions will not be used explicitly, but they can be helpful to verify the calculations in Sections 6.5 and 6.6.

$$\mathsf{S}_1(x_1, x_2, x_3, x_4) = x_1x_2x_3 + x_1x_3x_4 + x_1x_2 + x_1x_3 + x_3x_4 + 1$$

$$\mathsf{S}_2(x_1, x_2, x_3, x_4) = x_1x_2x_3 + x_1x_3x_4 + x_2x_3x_4 + x_1x_4 + x_1 + x_4 + 1$$

$$\mathsf{S}_3(x_1, x_2, x_3, x_4) = x_1x_2 + x_1x_4 + x_2x_4 + x_2 + x_4$$

$$\mathsf{S}_4(x_1, x_2, x_3, x_4) = x_1x_2x_3 + x_1x_3x_4 + x_2x_3x_4 + x_1x_4 + x_2x_4 + x_3\,.$$

The permutation ShuffleCell (P) interchanges the cells of the state. It operates on the state as follows:

s_1	s_5	s_9	s_{13}
s_2	s_6	s_{10}	s_{14}
s_3	s_7	s_{11}	s_{15}
s_4	s_8	s_{12}	s_{16}

$\xrightarrow{\mathsf{P}}$

s_1	s_{15}	s_{10}	s_8
s_{11}	s_5	s_4	s_{14}
s_6	s_{12}	s_{13}	s_3
s_{16}	s_2	s_7	s_9

The MixColumn ($\mathfrak{M}$) transformation acts on each state column independently by the following matrix over $\mathbb{F}_{2^4}$:

$$\mathsf{M} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

That is, each cell of a column of the state is replaced by the exclusive-or of the other cells in the same column. Finally, the round key in round i is alternately taken to be $k_0 + \gamma_i$ or $k_1 + \gamma_i$, where γ_i is a round constant. Importantly, round constants are only added to the least significant (rightmost) bit of each cell, *i.e.* $\gamma_i \in \{0, 1\}^{16}$.

The tweakable block cipher MANTIS [29] is quite similar to Midori-64, having nearly the same round function. Figure 6.2 illustrates the overall structure of MANTIS-4. Unlike in Midori, the round key k_1 is the same in all rounds. Additional whitening keys k_0 and $k_0' = (k_0 \ggg 1) + (k_0 \gg 63)$ are added before the first round and after the last round. The round function is nearly identical to the Midori-64 round function, the difference being that the round keys and constants are added before rather than after the application of $\mathfrak{M}$. The i^{th} round constant of MANTIS is denoted by c_i.

Structurally, MANTIS differs from Midori-64 in two major aspects: it takes an additional tweak as an input, and it is a reflection cipher. The reflection constant is denoted by $\alpha =$ `0x243f6a8885a308d3`. In every round, the tweak is permuted cellwise by a permutation σ. In all other aspects, the tweak is treated in the same way as the round key k_1.

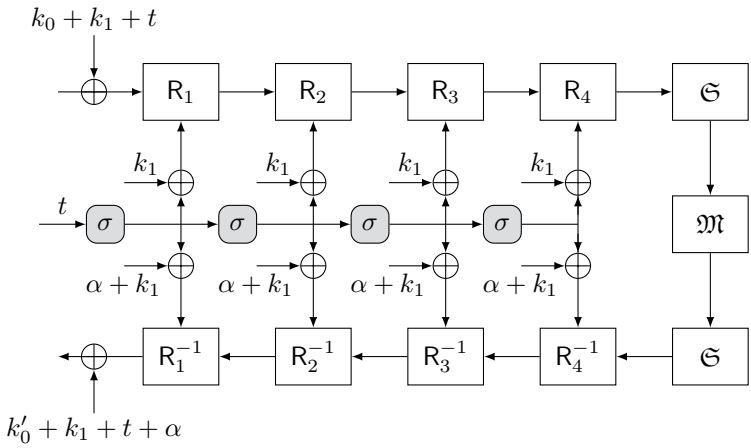


Figure 6.2: Overview of MANTIS-4.

6.4 Invariants for Midori-64

In this section, the invariants of Midori-64 are discussed from the point of view that they are eigenvectors of correlation matrices. As an example, Section 6.4.2 revisits the invariant subspace attack of Guo *et al.* [154] and the nonlinear invariant from Todo *et al.* [266]. In Section 6.4.3, a more general invariant will be obtained. This invariant will be used in Sections 6.5 and 6.6 to obtain attacks on reduced-round Midori-64 and MANTIS.

6.4.1 State representation and round transformations

Up to the identification of $\mathbb{F}_2^n$ with its dual group, the Fourier-domain representation of the Midori-64 state is a function v in $\mathbb{R}^{\mathbb{F}_2^{64}}$. Recall from Section 6.3 that it is convenient to represent the Midori-64 state as a 4×4 array of 4-bit cells. For this reason, coordinate $u = (u_1, \ldots, u_{16})$ of v will be denoted by $v(u) = v(u_1, \ldots, u_{16})$. This notation reflects the fact that we can think of v as a tensor of order 16, *i.e.* $v \in \left(\mathbb{R}^{\mathbb{F}_2^4}\right)^{\otimes 16}$.

From Figure 6.1, the correlation matrix of the Midori-64 round function satisfies

$$C^{\mathsf{R}_i} = C^{l_i + \gamma_i} C^{\mathfrak{M}} C^{\mathsf{P}} C^{\mathfrak{S}},$$

where $l_i = k_0$ when i is odd and $l_i = k_1$ when i is even. Recall that $C^{l_i+\gamma_i}$ is a diagonal matrix. It follows from Corollary 3.1 (1) that $C^{\mathfrak{S}} =$

$\left(C^{\mathsf{S}}\right)^{\otimes 16}$ and $C^{\mathfrak{M}} = \left(C^{\mathsf{M}}\right)^{\otimes 4}$. The matrix C^{S} is a symmetric orthogonal matrix and C^{M} is a symmetric permutation matrix. Specifically, we have $C^{\mathsf{M}}_{u,v} = \delta_u(\mathsf{M}v)$ by Theorem 3.5 (2). Finally, C^{P} is a permutation matrix such that $(C^{\mathsf{P}}v)(u_1, \ldots, u_{16}) = v(u_{\pi^{-1}(1)}, \ldots, u_{\pi^{-1}(16)})$ with π the ShuffleCell permutation.

It is convenient to look only for invariants with *independent cells*, *i.e.* rank-one invariants in the sense of Section 3.7.1. That is, it will be assumed that there exist vectors $v_1, \ldots, v_{16}$ such that

$$v(u_1, \ldots, u_{16}) = \prod_{i=1}^{16} v_i(u_i). \tag{6.2}$$

Equivalently, $v = \bigotimes_{i=1}^{16} v_i$. Of course, this assumption imposes a serious restriction. However, assuming (6.2) greatly simplifies the analysis and is sufficiently general to recover the invariant attacks of Guo *et al.* [154] and Todo *et al.* [266]. Furthermore, more general assumptions are not necessary to obtain the invariant that will be presented in Section 6.4.3.

The invariants considered in Section 6.4.2 will be required to be invariant under $\mathfrak{S}$, $\mathfrak{M}$ and P. Consider the last requirement, *i.e.* v is an eigenvector of C^{P}. Recall that C^P is a permutation matrix such that

$$C^{\mathsf{P}} \bigotimes_{i=1}^{16} v_i = \bigotimes_{i=1}^{16} v_{\pi^{-1}(i)}.$$

If v is symmetric, that is, $v_1 = \cdots = v_{16} = \widetilde{v}$, then $\bigotimes_{i=1}^{16} v_i = \widetilde{v}^{\otimes 16}$ is clearly invariant under C^{P}. It turns out that for the purpose of this paper, it suffices to consider only invariants v such that $v = \widetilde{v}^{\otimes 16}$ for some $\widetilde{v}$ in $\mathbb{R}^{\mathbb{F}_2^{16}}$. Such tensors v are called symmetric. Note that the symmetry assumption is less restrictive than (6.2). Indeed, for any realistic choice of round constants, an asymmetric invariant tends to lead to conflicting requirements on the key after a sufficient number of rounds. Slightly more general invariants can be obtained by requiring that $i \mapsto v_i$ is constant on the cycles of π.

Computing an eigenvector basis for $C^{\mathfrak{S}}$ is not difficult. In the remainder of this section, the symmetric rank-one eigenvectors of $C^{\mathfrak{M}}$ will be listed. In particular, it is not necessary to compute these eigenvectors numerically. The analysis starts from the straightforward result in Lemma 6.1. The main result is stated in Theorem 6.1.

Lemma 6.1. *If $v^{\otimes 4}$ is a real eigenvector of C^{M}, then there exists a scalar α in $\mathbb{R}^\times$ such that all coordinates of v in the standard basis are equal to 0 or $\pm\alpha$.*

Proof. The condition that $v^{\otimes 4}$ is an eigenvector of C^{M} is equivalent to

$$v^{\otimes 4}(u) = \lambda v^{\otimes 4}(\mathsf{M}u).$$

Hence, for all $u_1, \ldots, u_4$ in $\mathbb{F}_2^4$, it holds that

$$\prod_{i=1}^{4} v(u_i) = \lambda \prod_{i=1}^{4} v(\textstyle\sum_{j \neq i} u_j). \tag{6.3}$$

Note that no vector of the form $v^{\otimes 4}$ can correspond to $\lambda = -1$, since it follows from (6.3) that $v(u)^4 = \lambda v(u)^4$ for all u in $\mathbb{F}_2^4$. Since at least one coordinate of v is nonzero, there exists a u such that $v(u) = \alpha \neq 0$. By (6.3), this implies $\alpha v(u')^3 = \alpha^3 v(u')$ for any u' in $\mathbb{F}_2^4$. Consequently, $v(u') \in \{0, \pm\alpha\}$. □

Theorem 6.1. *If $v^{\otimes 4}$ is a real eigenvector of C^{M}, then $\mathcal{A} = \{u \mid v(u) \neq 0\}$ is an affine subspace of $\mathbb{F}_2^4$ and there exists a scalar α in $\mathbb{R}^\times$ such that $v(u) = \pm\alpha$ for all u in $\mathcal{A}$. The converse is also true in the following cases:*

- *For $\dim \mathcal{A} = 0$, $\dim \mathcal{A} = 1$ and $\dim \mathcal{A} = 2$.*
- *For $\dim \mathcal{A} = 3$, provided that the number of negative coordinates of v is even.*

The condition for $\dim \mathcal{A} = 3$ is also necessary.

Proof. Suppose that $v^{\otimes 4}$ is a real eigenvector of C^{M}. Let a, u, u' in $\mathbb{F}_2^4$ such that $v(a) \neq 0$, $v(a+u) \neq 0$ and $v(a+u') \neq 0$. By (6.3), it follows that

$$v(a+u+u')^2 v(a+u')v(a+u) = v(a)^2\, v(a+u)v(a+u') \neq 0.$$

Hence, $v(a+u+u') \neq 0$. This implies that $\mathcal{A}$ is an affine space. Lemma 6.1 completes the argument.

To show the converse, first consider the case with $\dim \mathcal{A} \leq 2$. It suffices to demonstrate that if $u_1, \ldots, u_4 \in \mathcal{A}$, then $\prod_{i=1}^4 v(u_i) = \prod_{i=1}^4 v(\sum_{j \neq i} u_j)$. Note that $\{u_1, \ldots, u_4\}$ and $\{\sum_{i \neq 1} u_i, \ldots, \sum_{i \neq 4} u_i\}$ generate the same affine space. Since the dimension of this space is at most two, it contains at most four elements. Hence, both products contain the same factors.

For $\dim \mathcal{A} = 3$, the previous argument no longer applies when $u_1, \ldots, u_4$ are linearly independent. In this case the left and right hand side of $\prod_{i=1}^4 v(u_i) = \prod_{i=1}^4 v(\sum_{j \neq i} u_j)$ involve different variables. Hence, since $\mathcal{A}$ contains eight elements, it is necessary and sufficient that the product of these elements is positive. □

The only symmetric rank-one invariants which are not covered by Theorem 6.1 are those having only nonzero coordinates. It would be possible to extend the result to cover this case as well, but this would have little practical value since such eigenvectors can never lead to a significant class of weak keys due to Corollary 6.1.

The search tool from Section 3.7.2 provides an alternative to Theorem 6.1, although it has the downside that it yields less insight. It was already shown in Example 3.16 that this tool can be used to find rank-one eigenvectors of C^{M}. By adding appropriate constraints, the results in the following two sections can be reproduced with little effort. Both approaches have their merits. The presentation below follows the original paper from Asiacrypt 2018 [37] and hence relies on Theorem 6.1.

6.4.2 Simultaneous eigenvectors

As discussed in Section 6.2, it is usually not possible to find the eigenvectors of C^{E_k} directly and to subsequently identify those vectors that depend only on a limited portion of the key. A more realistic approach is to find joint eigenvectors for all of the transformations in the round function. This corresponds to the strategy that was used in previous work, and it is the strategy that will be applied in this section.

The problem considered in this section is thus to find vectors v in $\mathbb{R}^{\mathbb{F}_2^{64}}$ such that $C^{\mathfrak{S}}v = \lambda v$ and $C^{\mathfrak{M}}v = \mu v$ with λ and μ in $\{-1, 1\}$. Furthermore, v must be an eigenvector of C^{P}, but if v is symmetric, we need not separately consider this requirement. For each of these vectors v, we additionally require that they are eigenvectors of $C^{l_i+\gamma_i}$ for $i = 1, \ldots, 16$. In general, this is not possible without making some assumptions on the keys l_i.

If $\{v_1, \ldots, v_{16}\}$ is a basis of eigenvectors of C^{S}, then the set of all vectors of the form $\bigotimes_{i=1}^{16} v_{j_i}$ with j_i in $\{1, \ldots, 16\}$ is a basis of eigenvectors of $C^{\mathfrak{S}} = \left(C^{\mathsf{S}}\right)^{\otimes 16}$. Suppose that $E_{+1}(C^{\mathsf{S}})$ is the eigenspace of C^{S} corresponding to eigenvalue 1, and $E_{-1}(C^{\mathsf{S}})$ likewise for eigenvalue -1. Any useful invariant must be an eigenvector of the diagonal matrices $C^{l_i+\gamma_i}$ as well. In summary, the invariants must be the fourth tensor power of an element of one of the vector spaces listed in Table 6.1.

The vectors $v^{\otimes 4}$ should additionally be eigenvectors of C^{M}. A necessary condition to this end is given by Theorem 6.1 (in fact, Lemma 6.1 is sufficient here). Using this result, only four nontrivial invariants of the form $v^{\otimes 16}$ remain. These are listed in Table 6.2. The first of these invariants is the Walsh-Hadamard transformation of a Boolean function. It corresponds to the nonlinear invariant

Table 6.1: Bases for the intersection of the eigenspaces of C^{S} and C^{γ_i}.

$\cap$	$\mathrm{Span}\{\delta_1, \delta_3, \ldots, \delta_{\mathtt{f}}\}$	$\mathrm{Span}\{\delta_0, \delta_2, \ldots, \delta_{14}\}$
$E_{+1}(C^{\mathsf{S}})$	$(1,0,0,0,0,0,0,0,\ \ 0,0,\ \ 0,0,\ \ 0,0,\ \ 0,0)$ $(0,0,1,0,1,0,1,0,-1,0,-1,0,-1,0,-1,0)$	$(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$
$E_{-1}(C^{\mathsf{S}})$	$(0,1,0,0,0,1,0,0,0,-1,0,\ \ 0,0,-1,0,-2)$ $(0,0,0,1,0,0,0,1,0,\ \ 0,0,-1,0,\ \ 0,0,\ \ 1)$	

discovered by Todo, Leander and Sasaki [266]. The eigenvector in the second row of Table 6.2 is the Fourier transformation of the indicator function of the invariant subspace obtained by Guo *et al.* [154].

Table 6.2: Invariants for Midori-64 with weak key class K. Note that the last invariant is simply the nonlinear invariant corresponding to the second invariant (which is an invariant subspace).

Eigenvector (v for $v^{\otimes 16}$)	K	$\lvert K\rvert$
$(0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,-1,\ 0,\ 0,\ 0,\ 1)$	$\kappa_1 = \kappa_2 = 0$	2^{64}
$(1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,-1,\ 0,-1,\ 0,-1,\ 0,-1,\ 0)$	$\kappa_1 = \kappa_2 = \kappa_3 = 0$	2^{32}
$(1,\ 0,-1,\ 0,-1,\ 0,-1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0)$	$\kappa_1 = \kappa_2 = \kappa_3 = 0$	2^{32}
$(0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,-1,\ 0,-1,\ 0,-1,\ 0,-1)$	$\kappa_1 = \kappa_2 = \kappa_3 = 0$	2^{32}

Note that the weak key class corresponding to a given invariant (the second column in Table 6.2) is readily determined from the vector v. For instance, consider the vector $C^{\kappa}v$, with $\kappa = (\kappa_1, \ldots, \kappa_4)$ in $\mathbb{F}_2^4$ a nibble of the round key:

$$v = (0,0,0,1,0,0,0,1,0,0,0,-1,0,0,0,1),$$

$$C^{\kappa}v = (-1)^{\kappa_3+\kappa_4}(0,0,0,1,0,0,0,(-1)^{\kappa_2},0,0,0,(-1)^{1+\kappa_1},0,0,0,(-1)^{\kappa_1+\kappa_2}).$$

Hence, v is invariant under C^{κ} provided that $\kappa_1 = \kappa_2 = 0$. Note that v is also invariant under the addition of the round constants – which has the same effect as modifying κ_4.

An alternative approach to finding invariants starts from the eigenvectors of C^{M}. Theorem 6.1 makes this method efficient. This will be the starting point to obtain more general invariants in Section 6.4.3.

6.4.3 Nonlinear invariant for "almost Midori-64"

In the previous section, a few eigenvectors of C^{R_i} were obtained by intersecting the eigenspaces of $C^{\mathfrak{M}}$, $C^{\mathfrak{S}}$ and $C^{l_i+\gamma_i}$. In general the eigenvectors of C^{R_i} are not eigenvectors of $C^{\mathfrak{M}}$ or $C^{\mathfrak{S}}$. Furthermore, the eigenvectors of C^{E_k} need not be eigenvectors of the round functions C^{R_i}. In order to find all invariants, then, it would be necessary to solve the eigenvalue problem directly. As discussed before, tackling this problem is out of the scope of this chapter, but a slightly more general type of invariant for Midori-64 is presented below.

Figure 6.3 shows the general idea: it may be possible to find a vector $u^{\otimes 16}$ which is mapped to a vector $v^{\otimes 16}$ by C^{R_i}, such that $C^{\mathsf{R}_{i+1}} v^{\otimes 16} = u^{\otimes 16}$. Such a vector $u^{\otimes 16}$ would be an eigenvector of $C^{\mathsf{R}_{i+1}} C^{\mathsf{R}_i}$, but not of C^{R_i}.

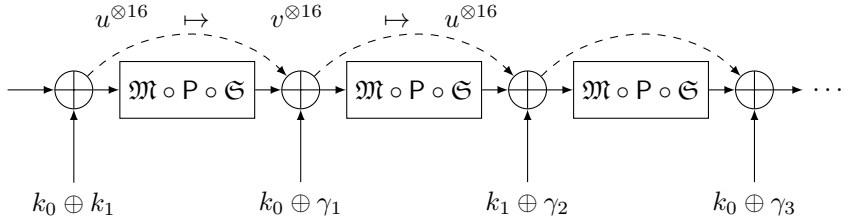


Figure 6.3: If $u \neq v$, this figure depicts an invariant for two rounds which is not invariant under one round.

To find such an invariant, it suffices to obtain vectors u and $v = C^{\mathsf{S}} u$ such that $C^{\mathsf{M}} u^{\otimes 4} = u^{\otimes 4}$ and $C^{\mathsf{M}} v^{\otimes 4} = v^{\otimes 4}$. Theorem 6.1 provides a complete list of possible choices for u and v. This approach is formalized in Algorithm 1. A Sage-implementation is available online[1]. This algorithm requires a negligible amount of time, as the inner loop is only executed 5216 times – once for each symmetric rank one invariant of $C^{\mathfrak{M}}$. Note that it also returns invariants of the conventional type.

A list of invariants produced by Algorithm 1 is given in Table 6.3. The most interesting pair of vectors u and v is given by

$$u = (0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0)$$

$$v = (0,0,0,0,0,0,0,0,0,0,-1/2,-1/2,0,0,1/2,-1/2).$$

Clearly, u is invariant under the addition of any constant. The vector v is an eigenvector of C^{κ} provided that $\kappa_2 = \kappa_4 = 0$. For the usual choice of round

[1] `http://tim.cryptanalysis.info/invariants/algorithm_1.html`

Algorithm 1 Symmetric rank-one invariants for two rounds of Midori-64.

for each affine subspace $\mathcal{A} \subseteq \mathbb{F}_2^4$ with $d := \dim \mathcal{A} \in \{0, 1, 2, 3\}$ **do**
  $S \leftarrow \{1\} \times \{1, -1\}^{2^d - 2}$
  if $d = 3$ **then**
    $S \leftarrow \{(s_1, \ldots, s_{2^d-1}, \prod_i s_i) \mid (s_1, \ldots, s_{2^d-1}) \in S\}$
  else
    $S \leftarrow S \times \{1, -1\}$
  end if
  for v in $\mathbb{R}^{\mathbb{F}_2^4}$ with $v(\mathbb{F}_2^n \setminus \mathcal{A}) = 0$ and $(v(u))_{u \in \mathcal{A}} \in S$ **do**
    $w \leftarrow C^{\mathsf{S}} v$
    $\mathcal{A}' \leftarrow \{u \in \mathbb{F}_2^4 \mid w(u) \neq 0\}$
    if $\mathcal{A}'$ is affine **and** $(\dim \mathcal{A}' \neq 3$ **or** $\prod_{u \in \mathcal{A}'} w(u) = 1)$ **then**
      yield v ▷ $v^{\otimes 16}$ is invariant for some choice of round constants
    end if
  end for
end for

constants of Midori-64, v is not invariant under the addition of the constants. However, had the round constants been chosen from $\{0, 2, 8, A\}^{16}$ rather than $\{0, 1\}^{16}$, the attack would apply. Moreover, such a restriction only applies to half of the rounds – the round constants of other rounds may be chosen arbitrarily.

The restriction $\kappa_2 = \kappa_4 = 0$ (which applies to k_0 or k_1, but not both) corresponds to a class of 2^{96} weak keys. One can verify that $v^{\otimes 16}$ corresponds to the following nonlinear invariant:

$$f(x_1, \ldots, x_{64}) = \sum_{i=1}^{16} x_{4i}x_{4i-2} + x_{4i} + x_{4i-1} + x_{4i-3}\,. \tag{6.4}$$

That is, there exists a constant c in $\mathbb{F}_2$ such that $f(\mathsf{E}_k(x)) + f(x) = c$ for all x and for any even number of rounds. One can verify that $u^{\otimes 16}$ corresponds to the following "nonlinear" invariant:

$$g(x_1, \ldots, x_{64}) = \sum_{i=1}^{16} x_{4i} + x_{4i-2}. \tag{6.5}$$

Hence, for an even number of rounds, $g(\mathsf{E}_k(x)) + g(x)$ is constant. If the number of rounds is odd, the value $f(\mathsf{E}_k(x)) + g(x)$ is constant instead.

Table 6.3: Invariants for two rounds of (modified) Midori-64, as obtained using Algorithm 1. Only invariants with at least 2^{64} weak keys are listed. Note that these invariants are not valid for all choices of the round constants. Type I refers to invariants with $u = v$, whereas type II indicates that $u \neq v$.

Invariant (v for $v^{\otimes 16}$)	Keys	Type
$(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$	2^{128}	Trivial
$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, -1, 1)$	2^{96}	II
$(0, 1, 0, 0, 1, 0, 0, 0, -1, 0, 0, 0, 0, 1, 0, 0)$	2^{80}	II
$(0, 0, 0, 1, 0, 0, -1, 0, 0, 0, 0, -1, 0, 0, -1, 0)$	2^{80}	II
$(1, -1, 0, 0, 0, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0)$	2^{64}	II
$(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1)$	2^{64}	II
$(0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0)$	2^{64}	II
$(0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)$	2^{64}	II
$(1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0)$	2^{64}	II
$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, -1, 0, 0, 1, 1)$	2^{64}	I
$(0, 0, 0, 0, 0, 0, 1, -1, 0, 0, 0, 0, 0, 0, 1, 1)$	2^{64}	I
$(0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, -1, 0, 0, 0, 1)$	2^{64}	I

6.4.4 Constructive interference in Midori-64

It is worthwhile to take a closer look at the invariant g given by (6.5) in Section 6.4.3. Since g is a linear function, it corresponds to a linear approximation with correlation ± 1 (where the sign depends on the key). Considering the fact that Midori-64 has been designed with resistance to linear cryptanalysis in mind, this is remarkable.

Result 6.1. *The correlation of any trail in "almost Midori-64" is (much) smaller than 2^{-32}, yet there is a linear approximation with correlation ± 1 for 2^{96} keys when the number of rounds is even.*

This result is an extreme example of the constructive interference phenomenon that was discussed in Section 3.6.2. From the point of view of linear trails, the linear approximation with correlation one contains 2^{16r} trails with correlation $\pm 2^{-16r}$ if the number of rounds is r. For 2^{96} keys, the signs of the correlations of all these trails agree and the correlation of the approximation becomes ± 1. This appears to be the first real-world observation of such behavior.

6.4.5 More weak keys for the invariant from Section 6.4.3

This section shows that the invariant u from Section 6.4.3 is invariant under 2^{64} additional weak keys, under the same modifications of the round constants. Although 2^{64} is small compared to 2^{96}, the result is interesting because it provides an example of an invariant over four rounds which is not necessarily invariant over two rounds.

Let u and v be as defined at the end of Section 6.4.3. For all κ in $\mathbb{F}_2^4$ with $\kappa_2 = \kappa_4 = 1$, it holds that

$$C^\kappa v = (-1)^{\kappa_1+\kappa_3}/2 \cdot (0,0,0,0,0,0,0,0,0,0,-1,1,0,0,-1,-1).$$

Let $w = (-1)^{\kappa_1+\kappa_3} C^\kappa v$. By Theorem 6.1, $w^{\otimes 4}$ is an invariant of C^{M}. Furthermore, one can check that w is an eigenvector of C^{S}.

Hence, there exist 2^{32} keys k such that $C^k\, v^{\otimes 16} = \pm w^{\otimes 16}$ with $w^{\otimes 16}$ invariant under the round function. This observation can be used to show that $u^{\otimes 16}$ defines an invariant for $2^{96}+2^{64}$ rather than 2^{96} weak keys. Figure 6.4 illustrates this. The top branch in Figure 6.4 corresponds to the discussion in Section 6.4.3 and holds assuming that $k_{0,4i-2} = k_{0,4i} = 1$ for $i = 1, \ldots, 16$. The bottom branch corresponds to a different set of weak keys for which $k_{0,4i-2} = k_{0,4i} = 1$ and $k_{1,4i-2} = k_{1,4i} = 0$ for $i = 1, \ldots, 16$. Hence, the 4-round invariant in Figure 6.4 and its full-round extension hold for $2^{96} + 2^{64}$ weak keys.

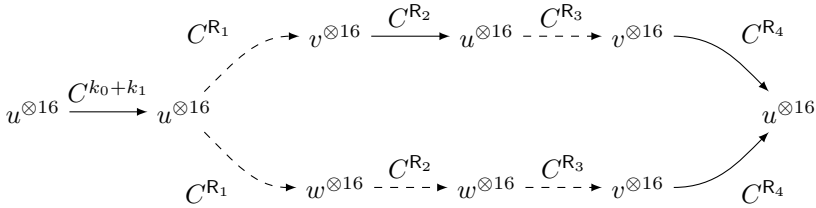


Figure 6.4: The invariant from Section 6.4.3 holds for $2^{96} + 2^{64}$ weak keys. Dashed arrows indicate transitions for which an assumption on the round keys is necessary.

6.5 Key-recovery attack on ten rounds of Midori-64

The purpose of this section is to demonstrate that the invariant for "almost Midori-64" can be used even when the round constants are not modified. In fact, the attack in this section is valid for any choice of round constants.

Specifically, it will be shown that 10 rounds of Midori-64 are subject to a key-recovery attack that requires $1.25 \cdot 2^{21}$ chosen plaintexts and has a computational cost of 2^{56} block cipher calls. The downside of this attack is that it is limited to 2^{96} out of 2^{128} keys. Note that Midori-64 has been analyzed in several prior works. Lin and Wu [201] demonstrate meet-in-the-middle attacks on 10, 11 and 12 rounds of Midori-64. Chen and Wang [89] give a 10 round impossible differential attacks. The downside of those attacks is that they cannot be executed in practice. Table 6.4 provides an overview of attacks on Midori-64.

Table 6.4: Overview of key-recovery attacks on Midori-64. Time is measured by the number of encryption operations. Memory is expressed in number of bytes.

Attack	Rounds	Time	Memory	Data	Keys	Ref.
Meet-in-the-middle	10	$2^{99.5}$	$2^{95.7}$	$2^{59.5}$	2^{128}	[201]
Meet-in-the-middle	11	2^{122}	$2^{92.2}$	2^{53}	2^{128}	[201]
Meet-in-the-middle	12	$2^{125.5}$	2^{109}	$2^{55.5}$	2^{128}	[201]
Impossible differential	10	$2^{80.8}$	$2^{68.1}$	$2^{62.4}$	2^{128}	[89]
Invariant subspace	16	2^{16}	—	2	2^{32}	[154]
Nonlinear invariant*	16	$2^{15}h$	—	$33h$	2^{64}	[266]
Integral/invariant	10	2^{56}	—	$2^{21.3}$	2^{96}	§6.5

* Attack on a mode of operation. It recovers $32h$ bits of h encrypted blocks.

The attack presented below is based on the observation that integral properties [184] and invariants can often be combined. However, because no assumptions on the round constants are made in this section, the invariant can only be used once. In this regard the nonlinear invariant that was introduced in Section 6.4.3 has an important advantage: with one assumption on the key, it covers two rounds.

6.5.1 Nonlinear property for six rounds of Midori-64

This section shows that the two-round nonlinear invariant for Midori-64 can be extended to a six round nonlinear property. When a key which does not belong to the weak key class is added to the state, the vector corresponding to a nonlinear invariant will be mapped to another vector which only depends (up to a scale factor) on key bits that are already "known", *i.e.* that had to be fixed to obtain the invariant in the first place. This holds in both the forward and backward direction, leading to a 6-round nonlinear property. This is illustrated in Figure 6.5.

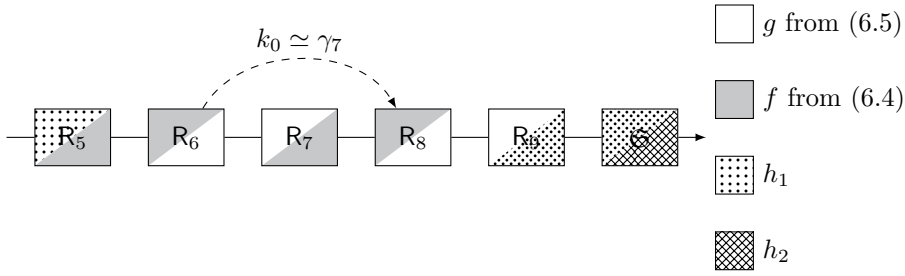


Figure 6.5: Nonlinear property for six rounds of Midori-64. The notation "$\simeq$" is used to indicate equality in the second and fourth bits of every nibble of each of its arguments.

The functions h_1 and h_2 in Figure 6.5 depend on the choice of the round constants. Specifically, h_1 depends on $\mathsf{P}^{-1}(\mathfrak{M}(\gamma_5 + \gamma_7))$ and h_2 depends on $\gamma_7 + \gamma_9$. For the purposes of this chapter, a detailed description of h_1 is not necessary. For h_2, it holds that

$$h_2(x_1, \ldots, x_{64}) = \sum_{i=1}^{16} f(\mathsf{S}(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}) + \gamma_{7,i} + \gamma_{9,i})\,.$$

In general, h_j can be written in the form

$$h_j(x_1, \ldots, x_{64}) = \sum_{i=1}^{16} h^{(\beta_{j,2i}, \beta_{j,2i+1})}(x_{4i}, x_{4i+1}, x_{4i+2}, x_{4i+3})\,, \tag{6.6}$$

where β_j is a constant in $\mathbb{F}_2^{32}$ depending on the round constants. In particular, β_2 consists of the second and fourth bits of every nibble of $\gamma_7 + \gamma_9$. For the default choice of round constants of Midori-64, $\beta_{j,2i} = 0$. Hence, only two different Boolean functions can occur as terms in (6.6):

$$h^{(00)}(x_1, x_2, x_3, x_4) = x_2 + x_4$$

$$h^{(01)}(x_1, x_2, x_3, x_4) = x_2x_3x_4 + x_1x_3x_4 + x_1x_2x_3 + x_1x_4 + x_1 + x_2\,.$$

Since the functions h_1 and h_2 are balanced *on every cell* of the state, it holds that $\sum_{x \in S} h_i(x) = 0$ with S a set of state values such that every cell has the saturated property. This makes it possible to combine integral cryptanalysis with the 6-round nonlinear property described above.

6.5.2 Integral property for four rounds of Midori-64

An integral attack on Midori-64 that is suitable for our purposes will now be given. The following standard notation will be used: saturated cells (taking all values an equal number of times) are denoted using the label "A", constant cells will be labeled by "C". Subscripts are used to denote groups of values which jointly satisfy the "A" property. Note that cells can be part of several groups, *e.g.* a cell marked "$A_{i,j}$" is contained in groups i and j. The Midori-64 designers discuss the existence of a 3.5 round integral distinguisher. In fact, one can see that a 4-round integral property[2] exists. Note that the property is nearly identical to the distinguisher discussed in Example 1.4, the difference being that the property works better than expected for Midori-64. Potential improvements using the division property or Chapter 5 are left as future work.

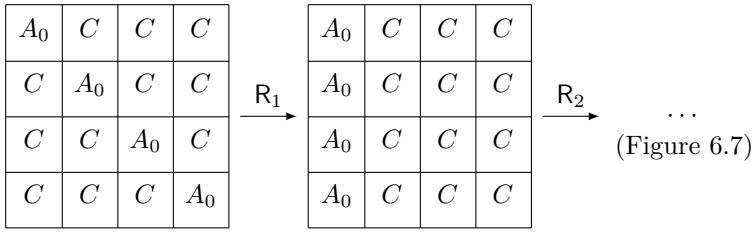


Figure 6.6: First two rounds of the integral property for four rounds of Midori.

The integral property is based on a set of chosen plaintexts such that the diagonal cells take all possible values exactly once and all other cells are constant. After one round, the same property then holds for the first column whereas all other cells are constant. This is shown in Figure 6.6.

The effect of the remaining rounds is shown in Figure 6.7. Figure 6.7 shows that, before the last application of $\mathfrak{M}$, any three distinct cells in a column jointly satisfy the "A" property. This implies that all cells can be labeled "A" after four rounds.

The derivation in Figure 6.7 starts by forming appropriate groups of cells which are independent before the third round. Four (sometimes overlapping) groups of such cells are indicated using "A_i", $i = 0, \ldots, 3$ in Figure 6.7. The maps $\mathfrak{S}$ and P preserve the groups. Furthermore, one can see that four new groups can be obtained after the application of $\mathfrak{M}$. These groups can be chosen in such a way that they are aligned in different columns of the state after P has been applied. The four-round property then follows.

[2]If the zero-sum property can be used, this actually yields a 5-round property.

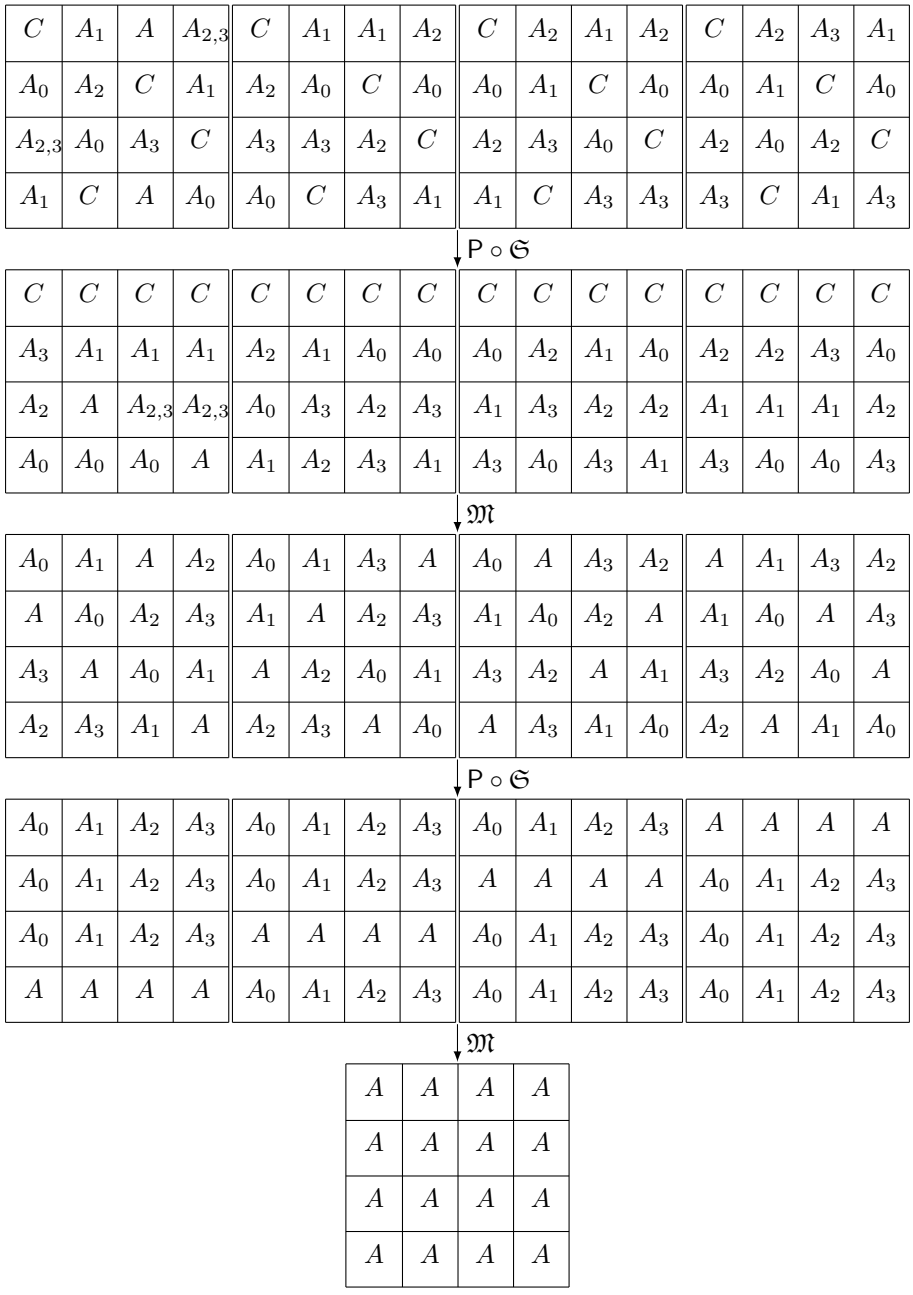


Figure 6.7: Last two rounds of the integral property for four rounds of Midori-64.

6.5.3 Combination of the nonlinear and integral properties

The final attack can now be described. Figure 6.8 provides an overview. Let $\mathcal{I}$ denote a set of plaintext/ciphertext pairs with the structure required by the integral property from Figure 6.6. Then, due to the nonlinear property from Figure 6.5, the following holds:

$$\sum_{(P,C)\in\mathcal{I}} h_2(C + k_0 + k_1) = \sum_{(P,C)\in\mathcal{I}} h_1((\mathsf{R}_4 \circ \cdots \circ \mathsf{R}_1)(P + k_0 + k_1)) = 0\,. \quad (6.7)$$

Hence, every set $\mathcal{I}$ defines a low-degree nonlinear polynomial equation in (part of) $k_0 + k_1$. Given enough such equations, one observes that a Gröbner basis for the ideal generated by these polynomials can be efficiently (within a second on a regular computer) computed. Although computing Gröbner bases is hard in general, it is easy in this case due to the fact that key bits from different cells are never multiplied together.

Note that only those key bits which are involved in h_2 in a nonconstant way can be recovered by solving the system of polynomial equations. That is, the number of key bits recovered is four times the number of nonlinear terms in (6.6). For the default Midori-64 round constants, 40 key bits can be recovered. It was observed that these bits are often uniquely determined given 40 equations. This requires $40 \cdot 2^{16} = 1.25 \cdot 2^{21}$ chosen plaintexts. A more detailed analysis of the data requirements is provided in Section 6.5.4.

The remaining 24 bits of $k_0 + k_1$ can be guessed, along with the 32 unknown bits in k_0. This requires 2^{56} block cipher calls. Note that this additional work is only necessary after it has been established that a weak key is used. Hence, an attacker in the multi-key setting has a very efficient method to identify potential targets.

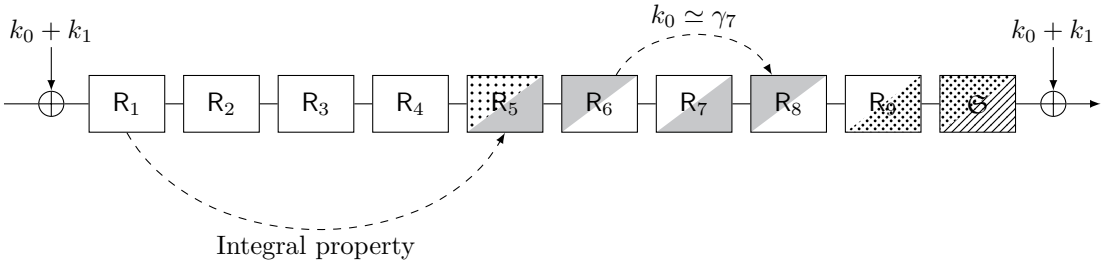


Figure 6.8: Overview of the attack on 10 rounds of Midori-64.

6.5.4 Detailed analysis of the data requirements

The data requirements of the attack are determined by the number of equations that are necessary to recover the 40 bits of $k_0 + k_1$ that can occur as indeterminates in (6.7). If the constant cells of each integral plaintext set are selected independently and uniformly at random, then the probability that the system of equations has a unique solution can be computed. Figure 6.9 provides an estimate of this probability based on a sample of 200 key-recovery experiments.

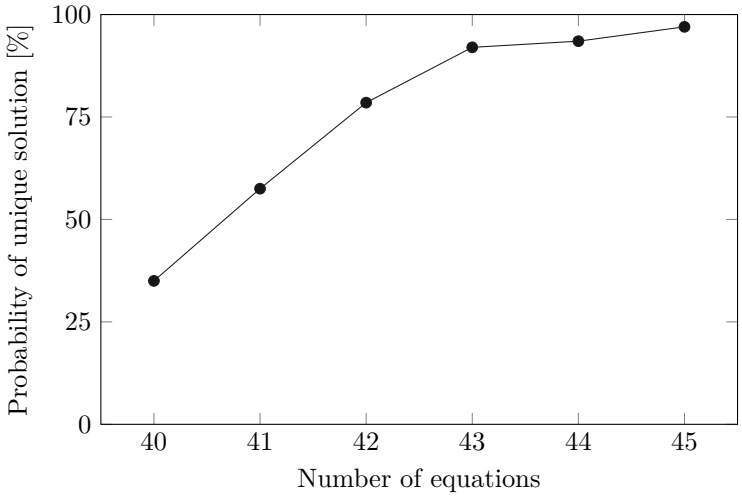


Figure 6.9: Probability that the system of equations for key-recovery has a unique solution. The equations are constructed from (6.7) by selecting the constant cells in the integral plaintext sets independently and uniformly at random.

For 40 equations – *i.e.* $1.25 \cdot 2^{21}$ chosen plaintexts – Figure 6.9 shows that the probability of recovering all 40 bits of the key is roughly 35%. With one additional equation, a probability of nearly 60% is obtained.

Note that even if the system does not have a unique solution, typically only a few additional bits of $k_0 + k_1$ will have to be guessed in the second phase of the attack. In order to minimize the required number of chosen plaintexts, additional equations may be constructed only when necessary.

6.6 Key-recovery attack on MANTIS-4

This section presents an attack on the block cipher MANTIS [29], which is closely related to Midori-64. Dobraunig, Eichlseder, Kales and Mendel give a practical attack against MANTIS-5 in the chosen tweak setting [123]. This attack has been extended to six rounds by Eichlseder and Kales [133]. The attack presented in this section is limited to MANTIS-4, but the assumptions about the capabilities of the attacker are different. The attacker is not allowed to choose the tweak, but it is assumed that a *weak tweak* is used. It will be shown that for every choice of the key, there are 2^{32} (out of 2^{64}) weak tweaks. When a weak tweak is used, the full key can be recovered from (on average) 346 chosen plaintexts and with a computational cost of approximately 2^{56} block cipher calls. If, in addition, 346 chosen ciphertexts for a single related tweak are available, the computational cost reduces to roughly 2^{18} block cipher calls. Table 6.5 contains an overview of attacks on MANTIS.

Table 6.5: Overview of key-recovery attacks on MANTIS-r. Time is measured by the number of encryption operations. 'Tweaks' is the number of weak tweaks.

Attack	r	Time	Memory	Data	Tweaks	Ref.
Truncated differential*	5	2^{28}	—	2^{38}	2^{128}	[123]
Truncated differential*	6	$2^{53.5}$	—	$2^{53.5}$	2^{128}	[133]
Zero-correlation/integral*†	3/7	$2^{66.2}$	$2^{48.4}$	$2^{53.7}$	2^{128}	[11]
Integral/invariant	4	2^{56}	—	346	2^{96}	§6.6.1
Integral/invariant*	4	2^{18}	—	692	2^{96}	§6.6.4

* These attacks rely on related tweaks.

† This attack applies to a version of MANTIS with an asymmetric number of rounds in the inbound (3) and outbound (7) direction. Such attacks are not considered in this thesis, but the techniques from this section could be used to obtain key-recovery attacks for MANTIS-6/4.

Due to the similarity between the round functions of MANTIS and Midori-64, the 2-round nonlinear invariant for Midori-64 also applies to MANTIS-4. In fact, the reflection property enables extending the 6-round nonlinear property of Midori-64 to eight rounds. Furthermore, the presence of a tweak allows mounting a weak tweak rather than a weak key attack. This corresponds to a significantly weaker adversarial model.

6.6.1 Description of the attack

An overview of the attack is shown in Figure 6.10. As in the attack on Midori-64 from Section 6.5, a few initial rounds are covered by an integral property. Since the nonlinear property extends over eight rounds for MANTIS, it suffices to use a weaker integral property. Figure 6.11 shows the property that will be used. It requires 16 chosen plaintexts.

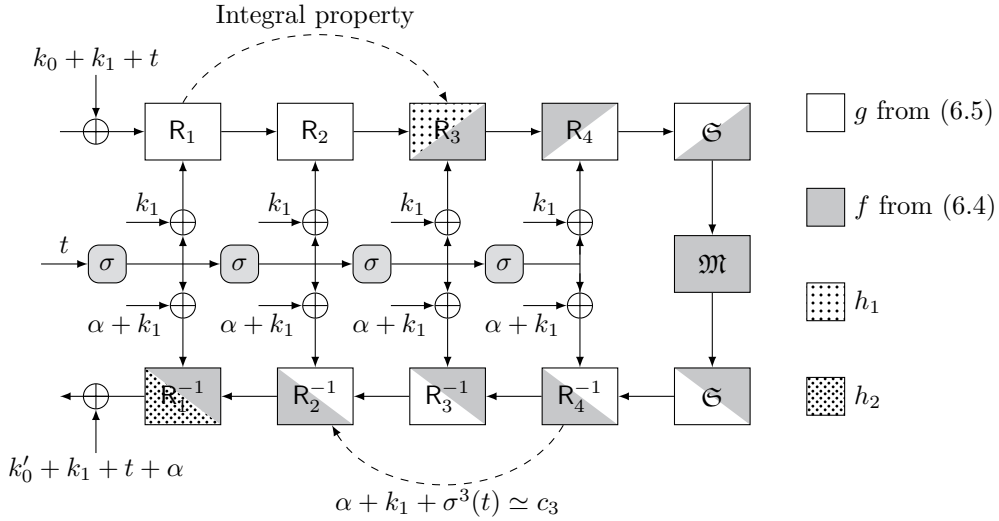


Figure 6.10: Nonlinear property over eight rounds of MANTIS-4. The notation "$\simeq$" is used to indicate equality in the second and fourth bits of every nibble of each of its arguments.

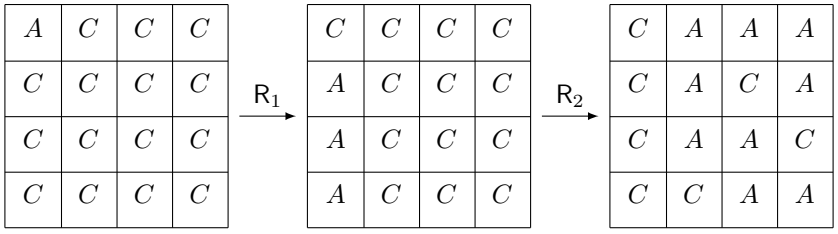


Figure 6.11: Integral property for two rounds of MANTIS.

The nonlinear property is similar to the property that was discussed in Section 6.5, but slightly more complicated. Specifically, due to the tweak-key schedule, the functions h_1 and h_2 can depend on the tweak. Like for Midori-64, h_1 and h_2 can be written in the form

$$h_j(x_1, \ldots, x_{64}) = \sum_{i=1}^{16} h^{(\beta_{j,2i}, \beta_{j,2i+1})}(x_{4i}, x_{4i+1}, x_{4i+2}, x_{4i+3}) \,, \tag{6.8}$$

where $\beta_j = (\beta_{j,1}, \ldots, \beta_{j,32})$ is a constant in $\mathbb{F}_2^{32}$ that possibly depends on the tweak and the functions $h^{(\beta_{j,2i}, \beta_{j,2i+1})}$ are given by

$$h^{(00)}(x_1, x_2, x_3, x_4) = x_2 + x_4$$

$$h^{(11)}(x_1, x_2, x_3, x_4) = x_2x_4 + x_1 + x_2 + x_3$$

$$h^{(01)}(x_1, x_2, x_3, x_4) = x_2x_3x_4 + x_1x_3x_4 + x_1x_2x_3 + x_1x_4 + x_1 + x_2$$

$$h^{(10)}(x_1, x_2, x_3, x_4) = x_1x_2x_3 + x_1x_3x_4 + x_2x_3x_4 + x_1x_4 + x_2x_4 + x_2 + x_3 + x_4 \,.$$

Note that all of these functions are balanced. The constant β_1 consists of the second and fourth bits of every nibble of α. For convenience, this will be denoted by $\beta_1 \simeq \alpha$. For β_2, we have $\beta_2 \simeq c_1 + \alpha + k_1 + \sigma(t)$. This implies that

$$\beta_2 \simeq c_1 + c_3 + \sigma(t) + \sigma^3(t) \,.$$

Let $\mathcal{I}$ denote a set of plaintext/ciphertext pairs such that the plaintexts have the structure required by the integral property, then

$$\sum_{(P,C)\in\mathcal{I}} h_2(C + k_0' + k_1 + t + \alpha) = \sum_{(P,C)\in\mathcal{I}} h_1(\mathsf{R}_1(\mathsf{R}_2(P + k_0 + k_1 + t))) = 0 \,.$$

Hence, each set $\mathcal{I}$ corresponds to a low-degree polynomial equation in (part of) the key. As in Section 6.5, a Gröbner basis for the ideal generated by these polynomials can be efficiently computed.

As in the attack on Midori-64, only the key bits which are involved in h_2 in a nonconstant way can be recovered by solving the system of polynomial equations. For simplicity, assume that the functions $h^{(00)}$, $h^{(01)}$, $h^{(10)}$ and $h^{(11)}$ all occur as terms in (6.8) in the same proportion. In this case, the average number of key bits that can be recovered by solving the system of polynomial equations is equal to 40.[3] It was observed that 40 equations are sufficient to extract 40 key bits. This requires $2^4 \cdot 40 = 640$ chosen plaintexts.

[3]For some tweaks, many more key bits can be recovered, and for others only a small number of key bits can be recovered. A detailed analysis is provided in Section 6.6.3.

The remaining bits of the whitening key $k'_0 + k_1$ (24 bits on average) can then be guessed, along with the 32 unknown bits of k_1. For each such guess, it is possible to compute k'_0 (since $k'_0 + k_1$ is already known) and hence k_0. No additional plaintext/ciphertext pairs are necessary to carry out this process. Hence, the work required for the entire key-recovery attack is then roughly 2^{56} block cipher calls.

6.6.2 Reducing data requirements by overlapping integral sets

Figure 6.11 shows one possible integral property for two rounds of MANTIS, but many alternatives exist. One example is shown in Figure 6.12. Since the input sets for the integral properties in Figures 6.11 and 6.12 overlap for an equal choice of the constant cells, the data requirements can be reduced.

For example, to obtain 40 distinct integral sets from only 316 chosen plaintexts, one proceeds as follows. First, choose 2^8 plaintexts such that the first byte of the state takes all possible values. This yields a total of 32 overlapping integral sets of size 16: half of these correspond to the integral property in Figure 6.11, the other half to that in Figure 6.12. For the eight remaining integral sets, choose one of the already queried plaintexts and build the integral set by letting the third cell take all possible values – this corresponds to yet another integral property similar to that in Figures 6.11 and 6.12. Overall, this requires $2^8 + 8 \cdot 15 = 316$ chosen plaintexts.

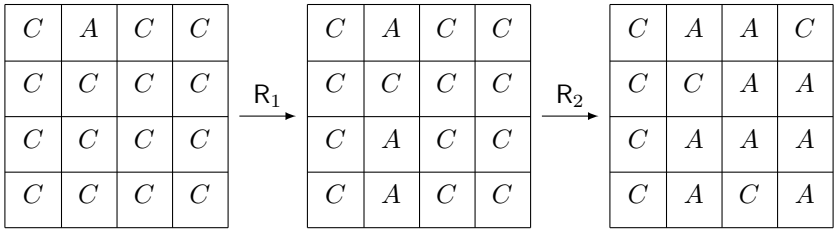


Figure 6.12: Alternative integral property for two rounds of MANTIS.

Note that the same technique can be applied to the attack on Midori-64 from Section 6.5, but it only reduces the data requirements by 40 chosen plaintexts.

6.6.3 Detailed analysis of the data requirements

As remarked in Section 6.6.1, the number of whitening key bits that can be recovered depends on the value of the tweak. Specifically, it depends on the value of β_2 in (6.8). Recall that β_2 consists of the second and forth bits of each nibble of $c_1 + c_3 + \sigma(t) + \sigma^3(t)$. Indeed, every term of the form $h^{(01)}$ or $h^{(10)}$ may contribute four unknowns to the system of equations in the key. A term of the form $h^{(11)}$ contributes at most two unknown key bits, whereas $h^{(00)}$ is linear and hence does not supply any key bits.

In Section 6.6.1, it was estimated that 40 bits of the key can be recovered. This corresponds to the average value for a uniform random choice of round constants. For a fixed choice of c_1 and c_3, the average number of recoverable key bits can be computed as follows. Clearly, $\sigma(\boldsymbol{t}) + \sigma^3(\boldsymbol{t})$ and $\boldsymbol{t} + \sigma^2(\boldsymbol{t})$ have the same probability distribution when $\boldsymbol{t}$ is a uniformly distributed random variable. Figure 6.13 illustrates the values of the nibbles of $\boldsymbol{t} + \sigma^2(\boldsymbol{t})$. The value of two cells, corresponding to fixed points of σ^2, is fixed whereas the other cells are individually – but not jointly – uniformly distributed.

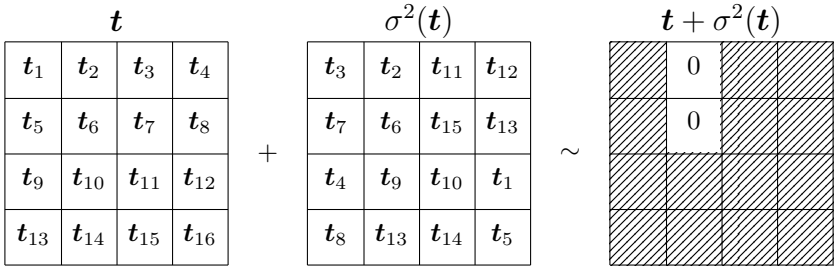


Figure 6.13: Illustration of the distribution of $\boldsymbol{t} + \sigma^2(\boldsymbol{t})$ with $\boldsymbol{t}$ uniformly distributed. The hatched cells are individually uniformly distributed, but their joint distribution is not uniform.

Hence, the average number of recoverable key bits depends only on the part of $c_1 + c_3$ corresponding to the two unhatched cells in the right part of Figure 6.13. Specifically, since these cells contribute terms of the form $h^{(01)}$ and $h^{(00)}$, it follows by linearity of expectation that the average number of key bits that can be recovered equals $4 + 14\,(2 + 1/2) = 39$. Figure 6.14 shows a histogram of the number of recovered key bits for 100000 tweaks sampled uniformly at random (with replacement). Remark that the distribution is right-skewed. In particular, while the mean number of recovered bits is 39, the median is in fact 40. The probability that at least 40 key bits can be recovered is approximately 50.4%.

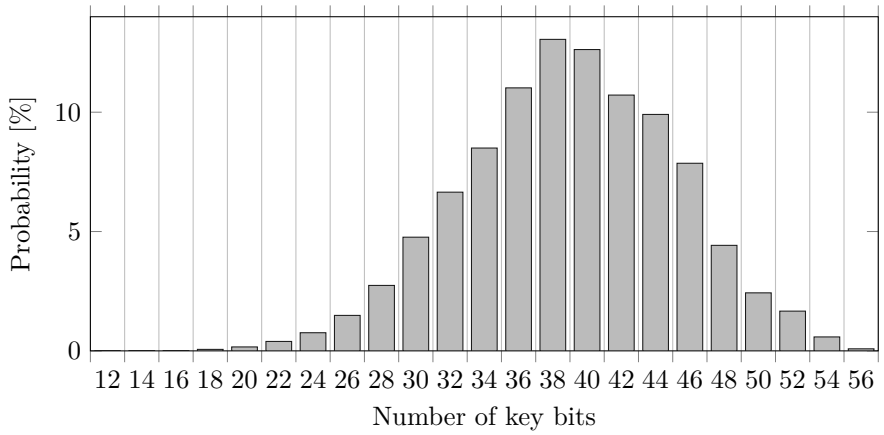


Figure 6.14: Estimated probability distribution of the number of key bits that can be recovered given a sufficiently large number of equations, for a randomly chosen tweak.

Like for the attack on Midori-64, the data requirements of the attack depend on the number of equations that are needed to uniquely recover the relevant key bits. The analysis is similar to that in Section 6.5.4. Figure 6.15 shows an estimate of the probability that the system of equations, when constructed from uniform random (overlapping) integral sets, has a unique solution. If the integral sets overlap, the probability of recovering all key bits is lower so that an additional equation is typically necessary.

To recover all 40 bits of the key with a success probability greater than 50%, 42 equations suffice. This corresponds to $2^8 + 6 \cdot 15 = 346$ chosen plaintexts.

6.6.4 Improved attack using related tweak chosen ciphertexts

If a small number of additional chosen ciphertexts under a single related tweak are available, then the computational cost of the attack can be significantly reduced. Specifically, given 346 chosen ciphertexts, the key-recovery cost can be reduced to 2^{18} block cipher calls. The basic idea is to perform the attack from Section 6.6.1 (without the brute-force phase) on the inverse cipher. An overview of the inverse attack is shown in Figure 6.16. Remark that the condition on the round key differs from that in Figure 6.10. Hence, in order to ensure that the property works for the same key k_1, a related tweak t' must be used. The only requirement on t' is that

$$t' \simeq t + \sigma^{-3}(\alpha)\,,$$

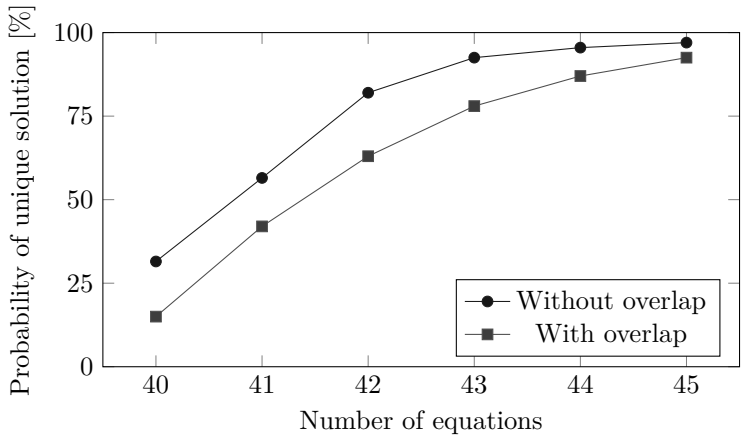


Figure 6.15: Probability that the system of equations for key-recovery on MANTIS-4 has a unique solution, estimated based on a sample of 200 key-recovery experiments.

where the symbol "$\simeq$" indicates equality in the second and fourth bits of every nibble. Hence, there are 2^{32} valid choices for the related tweak t'.

As in Section 6.6.1, an eight-round nonlinear approximation is combined with a two round integral property. Each integral set $\mathcal{I}$ defines an equation

$$\sum_{(P,C)\in\mathcal{I}} h_2'(P + k_0 + k_1 + t') = 0,$$

where h_2' is defined as in (6.8) but with a different constant $\beta_2' \simeq \beta_2 + \alpha + \sigma^{-2}(\alpha)$.

Since the bits of β_2' corresponding to the unhatched cells in Figure 6.13 are zero, the expected number of bits of $k_0 + k_1$ that can be recovered is the same as for the forward attack. Remark that, in the forward attack, one recovers bits of $k_0' + k_1$ with $k_0' = (k_0 \ggg 1) + (k_0 \gg 63)$ instead. One thus obtains a system of linear equations in k_0 and k_1. By linearity of expectation, the average number of equations is equal to $2 \cdot 39 = 78$. An estimate of the actual distribution of the number of equations is given in Figure 6.17. Since k_0 and k_0' are related by an orthomorphism, the equations in the system are linearly independent.

In conclusion, given 346 chosen plaintexts and 346 chosen ciphertexts for a related tweak, the full key can usually be recovered at a cost of 2^{18} block cipher calls. The cost of the Gröbner basis computations appears to be significantly smaller than 2^{18} encryption operations, but this may depend on the details of the implementation.

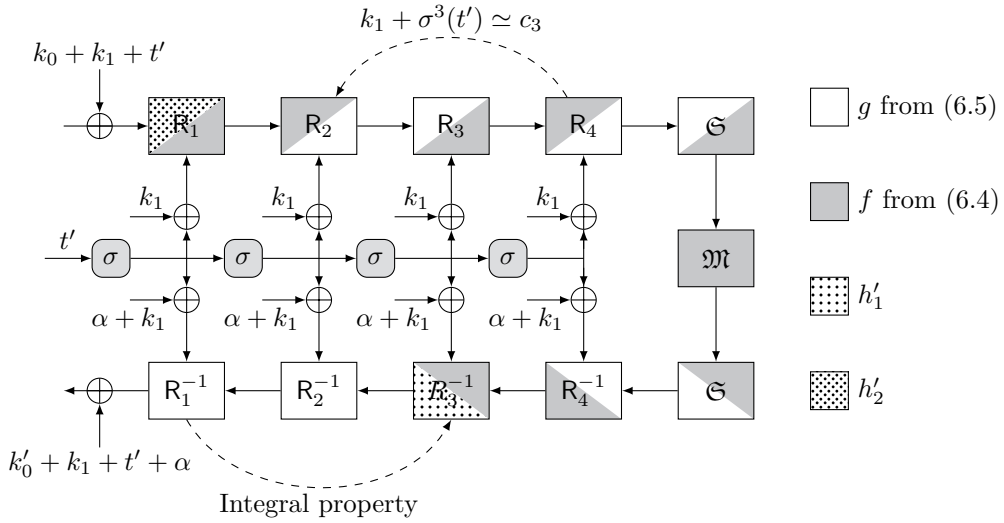


Figure 6.16: Attack on MANTIS-4 in the reverse direction. The notation "$\simeq$" is used to indicate equality in the second and fourth bits of every nibble of each of its arguments.

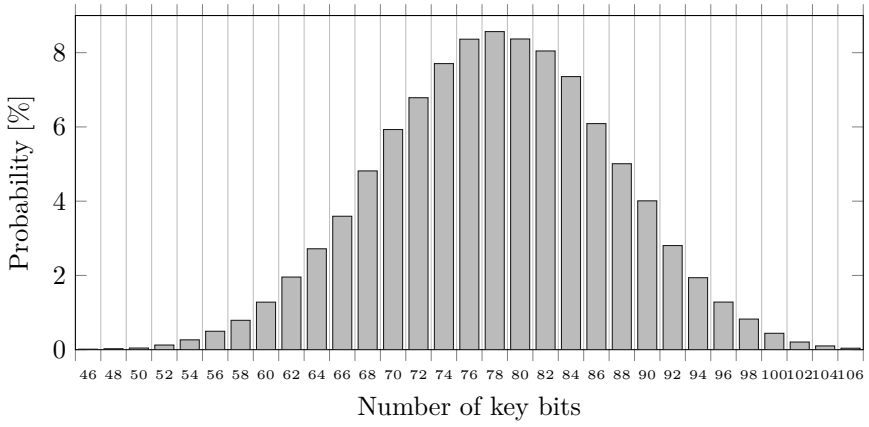


Figure 6.17: Estimated probability distribution of the number of key bits that can be recovered given a sufficiently large number of equations, for a randomly chosen tweak. Note that the distribution is more symmetric than the distribution from Figure 6.14.

7

Format-preserving encryption

This chapter develops distinguishing and message-recovery attacks on the format-preserving encryption standards FEA (South-Korea) and FF3-1 (United States). These attacks are based on multidimensional linear cryptanalysis – over $\mathbb{F}_2^n$ for FEA and over a cyclic group $\mathbb{Z}/N\mathbb{Z}$ for FF3-1. Their data- and time-complexities are low enough to be a practical concern for some applications.

The results of this chapter were published at Crypto 2021 in the paper "Linear cryptanalysis of FF3-1 and FEA" [41]. The text below is based on this paper, with adaptations where this was necessary for consistency (Section 7.4 in particular).

7.1 Introduction

Format-preserving encryption enables the encryption of plaintext with a specific format, while ensuring that the ciphertext has the same format. For example, in some applications it is convenient to be able to encrypt nine-digit integers (such as social security numbers) to nine-digit integers.

Several generic techniques such as cycle walking [32,67] can be used to transform (tweakable) block ciphers into format-preserving ciphers. However, these techniques are inefficient when there is a significant size difference between the domain of the underlying block cipher and the target domain. Consequently, a number of dedicated constructions based on small-domain tweakable Feistel ciphers were introduced. The best known examples are the United States standards FF1 and FF3-1 [129] (NIST SP800-38G rev. 1). The South-Korean standards FEA-1 and FEA-2 [198] (TTAK.KO-12.0275) follow a similar design but with lighter round functions.

Small-domain Feistel ciphers are known to be vulnerable to a number of generic attacks. In a series of papers, Patarin [227–229] analyzed the security of r-round Feistel ciphers with uniform random round functions. In particular, Patarin [229, §8] describes a distinguisher with data and time complexity $\widetilde{\mathcal{O}}(N^{r-4})$ for Feistel ciphers with domain size N^2. At CCS 2016, Bellare, Hoang and Tessaro [31] presented a message-recovery attack with a data complexity

of $\widetilde{\mathcal{O}}(N^{r-2})$ or $\widetilde{\mathcal{O}}(N^{r-3})$ (to recover the left half of the message) queries. Subsequent improvements were obtained by Hoang, Tessaro and Trieu [165].

The applicability of these attacks to FF3 in part motivated the US National Institute of Standards and Technology (NIST) to revise the FF3 standard [129]. In particular, the revised standard FF3-1 includes the requirement that the domain size must be at least one million, *i.e.* $N \geq 10^3$. Furthermore, the revision decreased the size of the tweak from 64 to 56 bits. This change was introduced to prevent a powerful slide-type attack presented by Durak and Vaudenay [127] at Crypto 2017 that was subsequently improved by Hoang *et al.* [164] and Amon *et al.* [10]. These attacks were the consequence of a weakness in the tweak-schedule of FF3 that is resolved by the changes in FF3-1.

This chapter develops new distinguishing and message-recovery attacks on small-domain Feistel ciphers with alternating round tweaks. The attacks are based on linear cryptanalysis, but go beyond standard methods in several ways. In particular, the role of the tweak input is analyzed, properties of small uniform random functions are exploited, and for FF3-1 linear cryptanalysis on the group $\mathbb{Z}/N\mathbb{Z}$ is used. Furthermore, the principle behind the message-recovery attacks is novel.

If the round tweaks alternate between two values, as in FEA-1 and FF3-1, the data and time complexity of these attacks is $\widetilde{\mathcal{O}}(N^{r/2-1.5})$. For FEA-2, which has a different tweak schedule, distinguishing and message-recovery respectively require $\widetilde{\mathcal{O}}(N^{r/3-1.5})$ and $\widetilde{\mathcal{O}}(N^{r/3-0.5})$ data and time. The new attacks are not applicable to FF1. For many instances of FF3-1, FEA-1 and FEA-2, the data and time complexity are well within the reach of real-world adversaries.

The proposed distinguishers only need weak access to the block cipher: it is sufficient to have ciphertext-only access to encryptions of an arbitrary constant message under many half-constant tweaks. In fact, access to the complete ciphertext is not necessary. The message-recovery attacks follow the security model introduced by Bellare *et al.* [31]. Specifically, given the encryption of a secret message and a known message with the same right-hand side under $\widetilde{\mathcal{O}}(N^{r/2-1.5})$ tweaks, the attack recovers the left half of the secret message. With $\widetilde{\mathcal{O}}(N^{r/2-0.5})$ queries, full messages can also be recovered. For FEA-1, the message-recovery attack can be used to set up a key-recovery attack. If q is the concrete data cost of the left-half message-recovery attack, then the key-recovery attack requires less than $16\lceil 8/\log_2 N\rceil\, q + 8q$ data and time equivalent to at most $2^{69}/N + 16\lceil 8/\log_2 N\rceil\, q + 8q$ evaluations of FEA-1.

Table 7.1 summarizes the cost of the main attacks from the literature and some of the new attacks proposed in this chapter. The advantage of a message-recovery attack is defined as the maximum value of $|P_{\mathsf{S}} - P_{\mathsf{F}}|$, where P_{S} is

Table 7.1: Summary of attacks on FEA-1, FEA-2 and FF3-1. The costs in the top half of the table are up to polylogarithmic factors in N (all of which are small in practice). Time is expressed in encryption operations. Memory requirements are small for all attacks. All of the message-recovery attacks listed in this table recover the left half of a message.

		Data	Time	Adv.	Ref.
Generic	Distinguisher	N^{r-4}	N^{r-3}	Constant	[126]
		N^{r-4}	N^{r-4}	Constant	[229]
		$N^{r/2-1}$	$N^{r/2-1}$	Constant	§7.3†
		$N^{r/2-1.5}$	$N^{r/2-1.5}$	Constant	§7.4†
		$N^{r/3-1}$	$N^{r/3-1}$	Constant	§7.3‡
		$N^{r/3-1.5}$	$N^{r/3-1.5}$	Constant	§7.4‡
	Message recovery	N^{r-3}	N^{r-3}	Constant	[31, 165]
		$N^{r/2-1.5}$	$N^{r/2-1.5}$	Constant	§7.5†
		$N^{r/3-0.5}$	$N^{r/3-0.5}$	Constant	§7.5‡
FEA-1 $N = 16, r = 12$	Distinguisher	2^{22}	2^{22}	0.1	§7.3
		2^{17}	2^{17}	0.1	§7.4
		2^{22}	2^{22}	0.6	§7.4
	Message recovery	2^{17}	2^{17}	0.1	§7.5
		2^{24}	2^{24}	0.6	§7.5
FEA-2 $N = 16, r = 18$	Distinguisher	2^{20}	2^{20}	0.1	§7.3
		2^{17}	2^{17}	0.1	§7.4
		2^{21}	2^{21}	0.6	§7.4
FF3-1 $N = 10^3, r = 8$	Distinguisher	2^{29}	2^{29}	0.1	§7.3
		2^{23}	2^{23}	0.1	§7.4
		2^{26}	2^{26}	0.6	§7.4
	Message recovery	2^{24}	2^{24}	0.1	§7.5
		2^{27}	2^{27}	0.6	§7.5

† For round tweaks that alternate between two values, as in FEA-1 and FF3-1.
‡ For round tweaks that alternate between three values, as in FEA-2.

the probability that the target message is not discarded when the number of candidates is narrowed down to a fraction P_{F} of the total number [31, 249].

The bottom part of the table reports concrete costs for the smallest instances of FEA-1, FEA-2 ($N = 16$) and FF3-1 ($N = 10^3$). Detailed cost-estimates for previous attacks on the same instances are not always available, but the improvement is substantial. For example, the attacks on FF3-1 with $N = 10^3$ require data and time comparable to previous attacks for $N = 2^5$ [31, 165] that led to the requirement $N \geq 10^3$. The numbers in Table 7.1 have been experimentally verified by performing each attack many times. Source code to reproduce this is available online[1]. Further experiments and cost calculations are given in the indicated sections.

The basic idea behind the attacks is introduced in Section 7.3: it is shown that there exists a linear trail through FEA-1 (and similarly for FEA-2) with high correlation. The novelty of this trail is the fact that it requires considering the tweak as a proper part of the input of the cipher. An analogous linear trail is then obtained for FF3-1, but using linear cryptanalysis on the cyclic groups $\mathbb{Z}/N\mathbb{Z}$ instead of $\mathbb{F}_2^n$.

Section 7.4 combines the linear approximations identified in Section 7.3 to obtain multidimensional linear approximations. These approximations are subsequently used to construct a χ^2-distinguisher. The formalism of multidimensional linear cryptanalysis is applied to justify the attack and to obtain initial estimates of the data complexity. Finally, Section 7.5 shows how the χ^2-distinguisher can be turned into a message-recovery attack. Each attack comes with a detailed analysis of the advantage and data complexity, and an experimental verification of the theoretical analysis.

Response to the attacks. In December 2020 – several months prior to the publication of the results in this chapter – both NIST (for FF3-1) and ETRI (for FEA-1 and FEA-2) were notified about these attacks. Both parties acknowledged the attacks and indicated their intention to revise their standards. Modifying the tweak schedule seems to be the most promising approach to thwart the attacks. In April 2021, I also submitted a comment to ISO SC27/WG2 as the standardization of format-preserving encryption was being considered at that time. This contributed to the decision to cancel this standardization project.

At the time of writing, NIST did not yet revise SP800-38G.

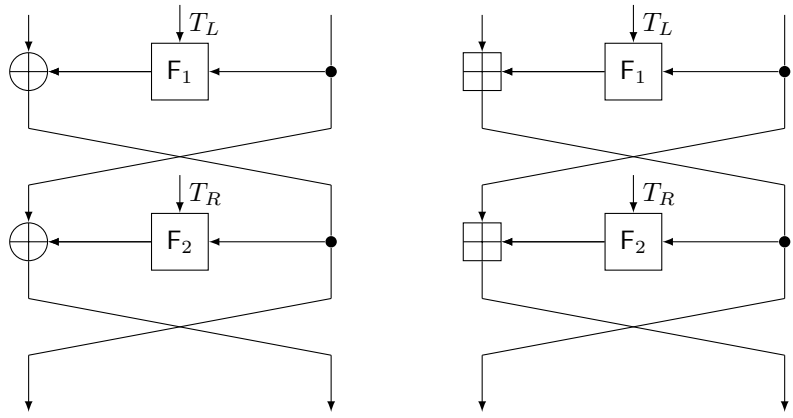
[1] `http://tim.cryptanalysis.info/fpe`

7.2 FEA and FF3-1

The attacks in this chapter are applicable to tweakable small-domain Feistel ciphers with alternating round tweaks. The South-Korean format-preserving encryption standards FEA-1 and FEA-2 [198] and the NIST standard FF3-1 [129] all follow such a design.

Figure 7.1 depicts two rounds of the overall structure of FEA-1 and FF3-1. For simplicity, it will be assumed that both branches have the same size. In both designs, the tweak is divided into two equal halves, which will be denoted by T_L and T_R for convenience. A crucial property that will be exploited by the new attacks is that the round tweak alternates between T_L and T_R. The round functions $\mathsf{F}_1, \mathsf{F}_2, \ldots$ can nevertheless be arbitrary.

As shown in Figure 7.1a, FEA-1 is a regular Feistel cipher over $\mathbb{F}_2^m \oplus \mathbb{F}_2^m$ with $m = \log_2 N$. For 128 bit keys, it has a total of 12 rounds. The tweaks T_L and T_R consist of $64 - m$ bits. The round functions F_i are truncations of a two-round SHARK-like construction (see Section 1.3.2), but can be considered to be uniform random functions for all attacks discussed in this chapter except for the key-recovery attack in Section 7.6. The necessary details of the round function will be reproduced in Section 7.6.



(a) FEA-1. (b) FF3-1.

Figure 7.1: Two rounds of a tweakable Feistel cipher with alternating round tweaks.

The design of FEA-2 is very similar to that of FEA-1. The main difference is that it uses three distinct round tweaks (repeating with period three), one of

which is constant. In addition, for FEA-2, both tweaks have a length of 64 bits and the number of rounds is 18 for 128 bit keys.

FF3-1 is an eight-round Feistel cipher over $\mathbb{Z}/N\mathbb{Z} \oplus \mathbb{Z}/N\mathbb{Z}$. The round functions $\mathsf{F}_1, \mathsf{F}_2, \ldots$ are defined as truncations of AES with the round tweak and a unique round counter as the input; the details are not important for this work as these functions will be modelled as uniform random. The tweaks T_L and T_R are bitstrings of length 28.

7.3 Linear distinguishers

In this section, linear distinguishers for FEA-1, FEA-2 and FF3-1 are introduced. Since the attacks on FEA-1 and FEA-2 are based on ordinary $\mathbb{F}_2$-linear cryptanalysis, these are described first in Section 7.3.1. Section 7.3.2 then transfers these results to Feistel ciphers defined over $\mathbb{Z}/N\mathbb{Z}$. In both cases, the analysis only relies on the theory of one-dimensional trails that was described in Section 3.3. Finally, the data complexity of the attacks is analyzed in detail and verified experimentally in Section 7.3.3.

7.3.1 FEA-1 and FEA-2

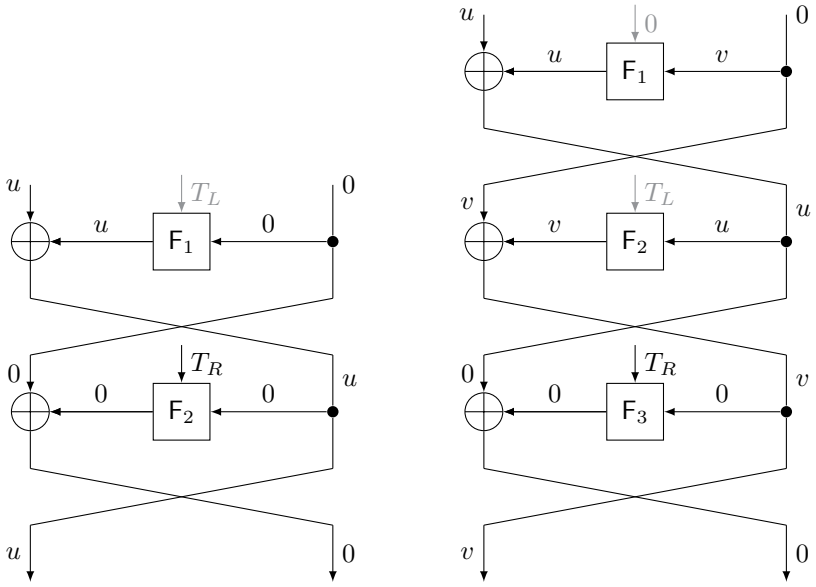
At first sight, both FEA-1 and FEA-2 seem to be robust against linear cryptanalysis, especially when their round functions $\mathsf{F}_1, \mathsf{F}_2, \ldots$ are replaced by uniform random functions. The key observation behind the attacks in this chapter is that this is not the case when (part of) the tweak is considered as a proper part of the input.

Figure 7.2 shows linear trails over two rounds of FEA-1 and three rounds of FEA-2[2]. As in Chapter 6, the group $\mathbb{F}_2^n$ is identified with its dual so that trails can be represented by sequences of masks rather than sequences of characters. In Figure 7.2, the tweak T_L is an arbitrary constant and T_R is considered to be a variable part of the input. Note that the tweak T_R is not active, so it need not be known to perform the attack. The idea behind these trails is that the absolute correlation of a linear approximation over the round function F_i (chosen uniformly at random) exceeds $1/\sqrt{N} = 2^{-m/2}$ with fairly high probability. This becomes meaningful when the tweak is included in the input, because the domain of the function which maps the tweak and the plaintext to the ciphertext is large. Indeed, the correlation of linear approximations over a random function with the same input size (including T_R of length $64 - m$) as

[2]I thank Dongyoung Roh for bringing the trails with $u \neq v$ to my attention.

FEA-1 is centered around zero with a standard deviation of $2^{-32-m/2}$. More specifically, we have the following result.

Theorem 7.1 (Daemen and Rijmen [106])**.** *Let $\boldsymbol{c}$ denote the correlation of a nontrivial linear approximation of a uniform random function $\mathbb{F}_2^n \to \mathbb{F}_2^m$. The random variable $2^{n-1}(\boldsymbol{c}+1)$ is binomially distributed with mean 2^{n-1} and variance 2^{n-2}. In particular*[3]*, as $n \to \infty$, the distribution of $2^{n/2}\boldsymbol{c}$ converges to the standard normal distribution $\mathcal{N}(0,1)$.*



(a) Two-round trail for FEA-1. (b) Three-round trail for FEA-2.

Figure 7.2: Linear trails for FEA-1 and FEA-2. The tweak T_R is considered part of the input and the value of T_L should be fixed.

Let $r \geq 2$ be an even integer. The correlation of the r-round trail from Figure 7.2a is equal to $\boldsymbol{c} = \prod_{i=1}^{r/2} \boldsymbol{c}_i$, where $\boldsymbol{c}_i \sim \mathcal{N}(0, 1/N)$ holds asymptotically due to Theorem 7.1. The random variables $\boldsymbol{c}_i$ will be assumed to be independent, which follows for instance from the strong assumption that the round functions $\mathsf{F}_1, \mathsf{F}_3 \ldots \mathsf{F}_{r-1}$ are independent. One can verify that the other trails through FEA-1 and FF3-1 have negligible correlation.

The data complexity of a constant-advantage linear distinguisher based on an approximation with correlation c is $\Theta(1/c^2)$. In this case, the correlation varies

[3]This result is a useful approximation even when n is small (for example, when $n \geq 8$).

strongly with the key so this result can not be applied directly to estimate the data complexity. A commonly used heuristic estimate is given by $1/\mathsf{E}\boldsymbol{c}^2$, where $\mathsf{E}\boldsymbol{c}^2$ is the average squared trail correlation for a uniform random key. For FEA-1, this yields $1/\mathsf{E}\boldsymbol{c}^2 = N^{r/2}$. The data complexity is analyzed in considerably more detail in Section 7.3.3.

For FEA-2 with r divisible by three, the expected squared correlation of each trail is equal to $N^{-2r/3}$. However, the number of trails for a given choice of input and output masks is $(N-1)^{r/3-1}$. Recall that the correlation of a linear approximation is equal to the sum of the correlations over all possible trails. Hence, since the trails in Figure 7.2b are indeed dominant, the sum $\boldsymbol{c}$ of the correlations of these trails is a good estimate for the correlation of the corresponding approximation. Since the covariance between the correlations of distinct trails is zero for independent uniform random round functions, it follows that

$$1/\mathsf{E}\boldsymbol{c}^2 = N^{2r/3}/(N-1)^{r/3-1} \sim N^{r/3+1}.$$

The fact that the covariance terms are zero is somewhat nontrivial, but it can be easily deduced from the definition of correlation for a uniform random function. Neglecting the covariance between the correlation of different trails is, in general, inaccurate. Finally, note that any other trail through FEA-2 necessarily has a much smaller average squared correlation.

Before continuing with the analysis of FF3-1, a simple but significant improvement to the correlation of the aforementioned linear approximation should be pointed out. If the right part of the plaintext is fixed to an arbitrary constant, then after two rounds the left branch of the state is equal to the left part of the plaintext up to addition by some constant. Consequently, the first two rounds can be effectively skipped. This decreases the data complexity by a factor N to $N^{r/2-1}$ for FEA-1. By fixing both halves of the plaintext, the first three rounds of FEA-2 can similarly be avoided. In addition, since the input mask is then no longer fixed, the number of trails within one approximation increases to $(N-1)^{r/3}$. Hence, the resulting data complexity estimate becomes $N^{r/3-1}$. A more detailed estimate of the data complexity is given in Section 7.3.3.

7.3.2 FF3-1

The analysis of FF3-1 proceeds analogously to that of FEA-1, but with linear cryptanalysis over the additive group $\mathbb{Z}/N\mathbb{Z}$ rather than $\mathbb{F}_2^m$. An iterative two-round trail is shown in Figure 7.3. In the figure, ψ denotes an arbitrary nontrivial character of $\mathbb{Z}/N\mathbb{Z}$ and $\mathbb{1}$ is the trivial character, *i.e.* $\mathbb{1}(x) = 1$ for all x in $\mathbb{Z}/N\mathbb{Z}$.

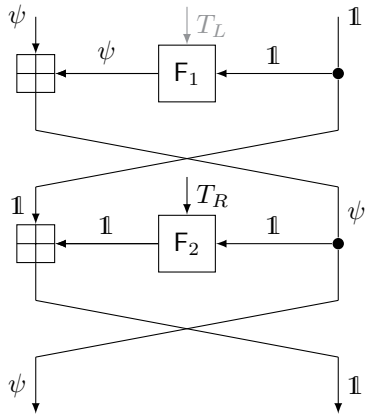


Figure 7.3: Iterative two-round trail for FF3-1. The tweak T_L is fixed.

In order to characterize the correlation of this trail, an analog of Theorem 7.1 is required. This is provided by Theorem 7.2 below. Recall that a complex-valued random variable $\boldsymbol{z}$ has a standard complex normal distribution $\mathcal{CN}(0,1)$ if its real part $\Re\{\boldsymbol{z}\} \sim \mathcal{N}(0,1/2)$ and its imaginary part $\Im\{\boldsymbol{z}\} \sim \mathcal{N}(0,1/2)$ are independent random variables.

Theorem 7.2. *Let G and H be finite commutative groups and let $\boldsymbol{c}$ denote the correlation of a nontrivial linear approximation of a uniform random function $G \to H$ corresponding to non-real characters. The correlation $\boldsymbol{c}$ has mean zero and variance $1/|G|$. Furthermore, as $|G| \to \infty$, the distribution of $\sqrt{|G|}\,\boldsymbol{c}$ converges to the standard complex normal distribution $\mathcal{CN}(0,1)$.*

Proof. Recall that a linear approximation corresponds to a pair of group characters (ψ, χ). The random variable $\boldsymbol{c}$ can be written as

$$\boldsymbol{c} = \frac{1}{|G|} \sum_{i=1}^{|G|} \psi(x_i)\, \overline{\chi(\boldsymbol{y}_i)}\,,$$

where $x_1, \ldots, x_{|G|}$ are the elements of G and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{|G|}$ are independent uniform random variables on H. The mean of $\boldsymbol{c}$ is zero, since $\mathsf{E}\chi(\boldsymbol{y}_i) = 0$ by the orthogonality relations for group characters. In addition, it follows from $\mathsf{E}|\chi(\boldsymbol{y}_i)|^2 = 1$ that $\mathsf{E}|\boldsymbol{c}|^2 = 1/|G|$. Finally, the convergence to a normal distribution follows from the central limit theorem for the sum of independent identically distributed random variables. □

By Theorem 7.2, the average squared correlation of the r-round trail from Figure 7.3 is equal to $N^{-r/2}$. As in the case of FEA-1, the right part of the

plaintext can be fixed in order to obtain a trail with average squared correlation $N^{1-r/2}$. This gives a corresponding data complexity estimate of $N^{r/2-1}$.

7.3.3 Cost analysis and experimental verification

As mentioned in Sections 7.3.1 and 7.3.2 above, the data complexity of a distinguisher based on a linear approximation with correlation c is roughly $1/|c|^2$. By *heuristically* plugging in the average squared trail correlation, the approximation $1/\mathsf{E}|\boldsymbol{c}|^2$ was obtained. This resulted in an estimated data complexity of $N^{r/2-1}$ for FEA-1 and FF3-1 and $N^{r/3-1}$ for FEA-2. This section analyzes the data complexity in more detail, along with the advantage achieved by the distinguisher. Broadly speaking, the detailed analysis confirms the heuristic estimates from Sections 7.3.1 and 7.3.2.

The distinguisher performs a hypothesis test, with null-hypothesis that the data comes from an ideal tweakable block cipher and alternative hypothesis that the data comes from the real cipher. If the absolute value of the estimated correlation exceeds a predetermined threshold, then the null-hypothesis is rejected. Like any hypothesis test, linear distinguishers allow for a trade-off between success probability P_{S} and false-positive rate P_{F}. Both probabilities are determined by the threshold parameter t. The distinguisher is successful if the estimated correlation exceeds $t\sqrt{q}$ when interacting with the true block cipher after q queries. If the estimated correlation exceeds this threshold for an ideal tweakable block cipher, then a false-positive occurs. Note that $P_{\mathsf{S}}(t)$ and $P_{\mathsf{F}}(t)$ are key-averaged quantities.

Figure 7.4 depicts the estimates of the maximum advantage $\max_t |P_{\mathsf{S}}(t) - P_{\mathsf{F}}(t)|$ which are derived below. Importantly, for large N, the curve is essentially independent of N. This will be shown below. The red dots correspond to experimental verifications of the estimates for full-round instances of FEA-1, FEA-2 and FF3-1. Each point corresponds to 1024 (FEA-1 and FF3-1) or 512 (FEA-2) evaluations of the distinguisher. For FF3-1, the experiments were performed for $N = 100 < 1000$ to limit the computational cost. The verification of the more efficient χ^2-distinguishers in Section 7.4 will be performed for $N = 1000$.

The false-positive rate is easily computed. Assume the correlation is estimated using q independent queries. If the input space is sufficiently large[4], then by Theorems 7.1 and 7.2 the variance of the ideal correlation is negligible. Hence, if the number of queries q is moderately large, then the estimated correlation $\widehat{\boldsymbol{c}}_{\mathsf{ideal}}$ is approximately distributed as $\mathcal{N}(0, 1/q)$ for FEA-1 and FEA-2

[4]Relative compared to the required number of queries q.

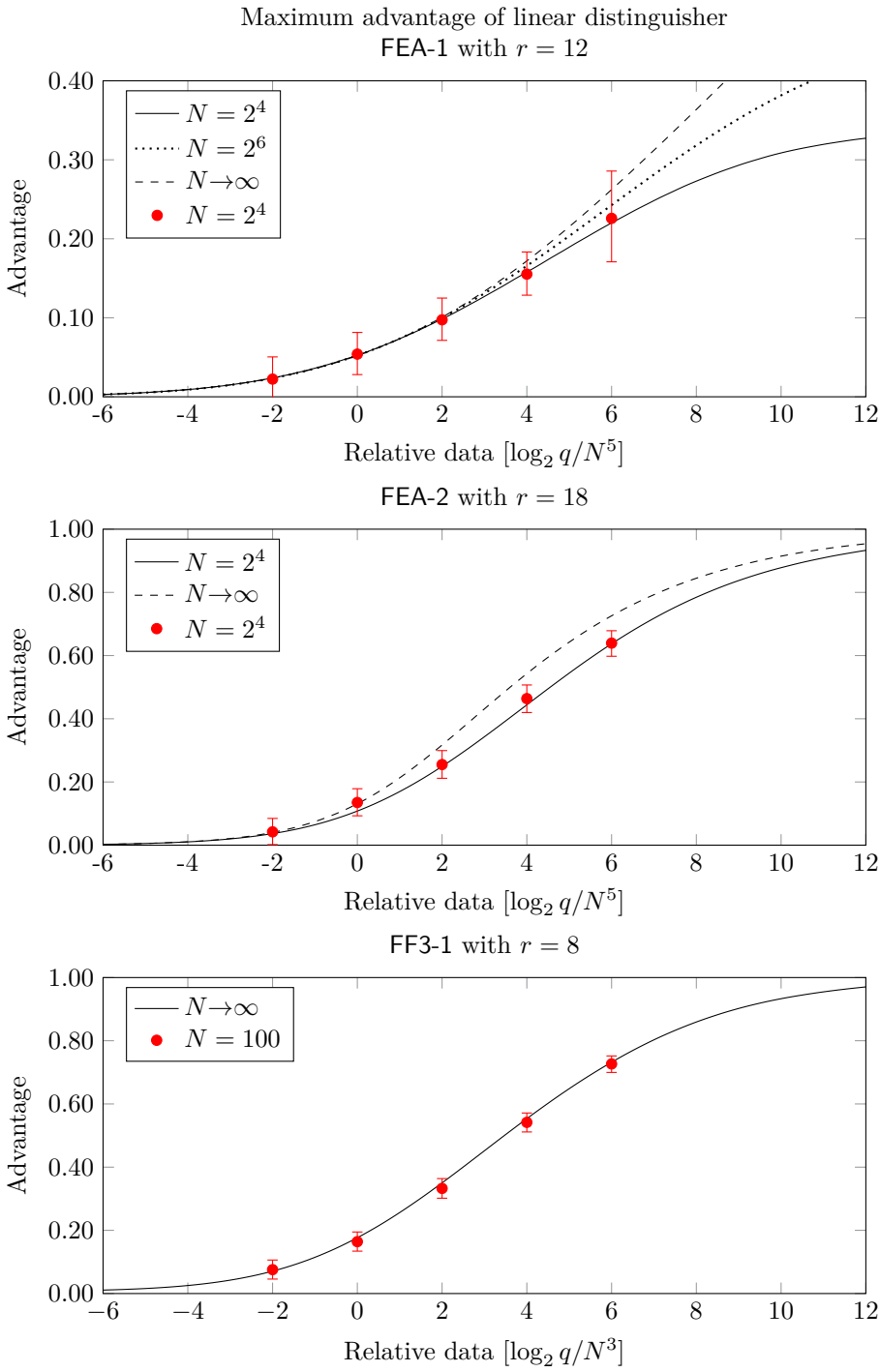


Figure 7.4: Theoretical and experimentally observed maximum advantage of the linear distinguishers for full-round FEA-1, FEA-2 and FF3-1. The error bars correspond to 95% Clopper-Pearson confidence intervals.

or $\mathcal{CN}(0, 1/q)$ for FF3-1. The false-positive rate is then

$$P_{\mathsf{F}}(t) = \Pr\left[|\widehat{\boldsymbol{c}}_{\mathsf{ideal}}| \geq t/\sqrt{q}\right] \approx 1 - \chi_\nu(\sqrt{\nu}t),$$

where χ_ν is the cumulative distribution function of the χ-distribution with ν degrees of freedom. For FEA-1 and FEA-2, $\nu = 1$ since $\boldsymbol{c}$ is real. For FF3-1, $\nu = 2$.

The calculation of the success rate P_{S} is more complicated, because the absolute correlation $|\boldsymbol{c}_{\mathsf{real}}|$ is not as strongly concentrated around its mean. Let $\widehat{\boldsymbol{c}}_{\mathsf{real}}$ denote the estimated correlation for a particular choice of the key. If the underlying correlation for this key is equal to c_{real}, then $\widehat{\boldsymbol{c}}_{\mathsf{real}}$ is approximately distributed as $\mathcal{N}(c_{\mathsf{real}}, 1/q)$ for FEA-1 and FEA-2 or $\mathcal{CN}(c_{\mathsf{real}}, 1/q)$ for FF3-1 if q is large enough and c_{real}^2 is much smaller than one. The average success probability can be approximated as

$$P_{\mathsf{S}}(t) \approx \mathop{\mathsf{E}}_{\boldsymbol{c}_{\mathsf{real}}} \Pr\left[|\boldsymbol{z}_\nu - \boldsymbol{c}_{\mathsf{real}}\sqrt{q}| \geq t\right],$$

where $\boldsymbol{c}_{\mathsf{real}}$ is the trail correlation assuming uniform random round functions and $\boldsymbol{z}_\nu$ a standard (complex if $\nu = 2$) normal random variable. To compute the average with respect to $\boldsymbol{c}_{\mathsf{real}}$, a Monte-Carlo approach was used. The implementation can be found online[5]. Importantly, the success probability curve (and consequently the maximum advantage) has essentially the same shape for all sufficiently large values of N. Indeed, by Theorems 7.1 and 7.2, the distribution of the round correlations converges to a (complex) normal distribution for large N. Hence, for $q_0 = 1/\mathsf{E}|\boldsymbol{c}_{\mathsf{real}}|^2$, the distribution of $\boldsymbol{c}_{\mathsf{real}}\sqrt{q_0}$ will be approximately the same for all large values of N. Consequently, the success probability curves tend to a constant function of q/q_0.

7.4 χ^2-distinguishers

This section introduces additional distinguishers on FEA-1, FEA-2 and FF3-1, based on Pearson's χ^2-test for goodness-of-fit between distributions. Vaudenay [274] proposed χ^2-distinguishers as a method for distinguishing non-uniform distributions in cryptanalysis when precise knowledge about these distributions is lacking.

The distinguishers in Section 7.3 are based on individual linear approximations. A natural improvement to these attacks is to exploit all approximations simultaneously. Multidimensional linear cryptanalysis provides a convenient framework to describe such attacks.

[5] `http://tim.cryptanalysis.info/fpe`

As shown in Section 3.4.2 in general and in Section 7.4.2 below for FEA-1 and FF3-1 in particular, the existence of a multidimensional linear approximation implies that a particular probability distribution related to the ciphertext is highly non-uniform. Pearson's χ^2-test can then be used to verify this property, resulting in a distinguisher.

Sections 7.4.1 and 7.4.2 explain the distinguisher in detail. The data complexity is estimated and experimentally verified in Section 7.4.3.

7.4.1 Multidimensional linear approximations

As discussed in Chapter 3 and Section 3.4.2 in particular, a multidimensional linear approximation consists of a pair of vector spaces that are spanned by subgroups of characters.

To obtain a uniform description of the attacks on FEA-1, FEA-2 and FF3-1, denote the half-domain by $\mathcal{D}$ and the space of tweaks T_R by $\mathcal{T}$. The ciphertext space is then $H = \mathcal{D} \oplus \mathcal{D}$. The input space G is either $\mathcal{D} \oplus \mathcal{T}$ or $\mathcal{T}$, depending on whether or not the left half of the plaintext is kept fixed (the right half always is). Let $\mathsf{F} : G \to H$ be the mapping from the input space to the ciphertext space corresponding to the cipher.

Any character ψ of $H \oplus G$ uniquely determines a linear approximation of the cipher. Specifically, the restriction of ψ to H corresponds to the output character of the approximation, and the restriction to G corresponds to the complex conjugate of the input character. The need for complex conjugation is due to technical reasons. Let Z^1 be the set of all such characters ψ corresponding to the linear approximations that were investigated in Section 7.3. This choice of notation hints at the fact that Z^1 is the annihilator (Definition 3.4) of a subgroup Z of $H \oplus G$. This will be motivated in Section 7.4.2. Concretely, with $\widehat{\mathcal{D}}$ the group of characters of the domain, let

$$Z^1 = \begin{cases} \{\psi : (y_L, y_R, x_L, T_R) \mapsto \overline{\chi(x_L)}\chi(y_L) \mid \chi \in \widehat{\mathcal{D}}\} & \text{for FEA-1 and FF3-1,} \\ \{\psi : (y_L, y_R, T_R) \mapsto \chi(y_L) \mid \chi \in \widehat{\mathcal{D}}\} & \text{for FEA-2.} \end{cases}$$

Note that for all three ciphers, Z^1 is a group under pointwise multiplication of functions. Hence, the Z^1 spans the output space of a multidimensional linear approximation of the function $\bar{\mathsf{F}} : x \mapsto (\mathsf{F}(x), x)$. Specifically, one has the multidimensional linear approximation (U, V) with $U = \mathrm{Span}\{\mathbb{1}\}$ and $V = \mathrm{Span}\{Z^1\}$. Finally, let $c : Z^1 \to \mathbb{C}$ be a function that assigns to a group character ψ in Z^1 the correlation $c(\psi)$ of the corresponding linear approximation.

As discussed in Examples 3.11 and 3.12, the data complexity of an optimal distinguisher based on a multidimensional linear approximation is inversely proportional to the capacity of the approximation, which is equal to

$$\|\langle V, U\rangle_{\bar{\mathsf{F}}}\|_{\mathsf{fr}}^2 - 1 = \sum_{\psi \neq \mathbb{1}} |c(\psi)|^2,$$

where the sum is over all nontrivial characters in Z^1. However, as pointed out in Section 7.3, the correlations $c(\psi)$ are heavily key-dependent and this will affect the optimal data complexity. Nevertheless, by linearity of expectation, it is easy to compute the key-averaged capacity:

$$\mathsf{E} \sum_{\psi \neq \mathbb{1}} |\boldsymbol{c}(\psi)|^2 \approx \begin{cases} N^{2-r/2} & \text{for FEA-1 and FF3-1,} \\ N^{2-r/3} & \text{for FEA-2.} \end{cases}$$

The above calculation suggests a data complexity of $N^{r/2-2}$ for FEA-1 and FF3-1 and $N^{r/3-2}$ for FEA-2. However, as will be shown below, this is too optimistic because the result that relates the capacity to the data complexity of an optimal distinguisher assumes that the correlations $c(\psi)$ are known exactly.

The multidimensional linear approximation can be turned into a distinguisher by directly estimating the capacity. It will be shown in Section 7.4.3 that the data complexity of this approach can be heuristically estimated as $\sqrt{N}/\sum_{\psi \neq \mathbb{1}} \mathsf{E}|\boldsymbol{c}(\psi)|^2$. However, there exists an equivalent but more direct distinguisher in terms of Pearson's χ^2-statistic.

7.4.2 Distinguisher based on Pearson's χ^2 statistic

Pearson's χ^2-statistic can be used as a measure of goodness-of-fit between an estimated (empirical) probability distribution $\widehat{p} : X \to [0, 1]$ and the uniform distribution on X. In this case, the χ^2-statistic with q samples satisfies

$$\chi^2/q = \left\| \widehat{p} - \frac{\mathbb{1}}{|X|} \right\|_X^2,$$

where $\|\cdot\|_X$ is the Euclidean norm scaled by $\sqrt{|X|}$ and $\mathbb{1}$ the indicator function of X. The χ^2-distinguisher succeeds in identifying the real cipher when the χ^2-statistic exceeds some threshold. Indeed, as $q \to \infty$, the estimated distribution $\widehat{p}$ tends to the true distribution p and χ^2/q tends to $\|p - \mathbb{1}/|X|\|_X^2$. In particular, if the tested distribution is uniform, then χ^2/q tends to zero as $q \to \infty$. Statistical aspects will be discussed in Section 7.4.3.

The existence of a strong multidimensional approximation implies that a probability distribution related to the plaintext and ciphertext is highly non-uniform. Specifically, as in Example 3.10, it holds that

$$V = \operatorname{Span}\left\{Z^1\right\} = \operatorname{Span}\left\{\mathbb{1}_z \mid z \in (H \oplus G)/Z\right\}.$$

Since $\mathbb{1}_z = \delta_z \circ \mathsf{P}_Z$ with $\mathsf{P}_Z(x) = x + Z$, the relevant distribution is that of $\mathsf{P}_Z((\mathsf{F}(\boldsymbol{x}), \boldsymbol{x})) = (\mathsf{F}(\boldsymbol{x}), \boldsymbol{x}) \bmod Z$ with $\boldsymbol{x}$ uniform random on G. For FEA-1 and FEA-2, Z can be taken as the orthogonal complement of the $\mathbb{F}_2$-vector space consisting of the masks in the multidimensional linear approximation. For both FEA-1 and FF3-1, the right half of the plaintext is fixed and reduction modulo Z corresponds to taking the difference of the left half of the ciphertext and the plaintext. More explicitly, if $\mathcal{D}$ is the half-domain of the cipher and $\mathcal{T}$ the space of half-tweaks T_R, then $H = \mathcal{D} \oplus \mathcal{D}$, $G = \mathcal{D} \oplus \mathcal{T}$ and

$$Z = \{(y_L, y_R, x_L, T_R) \in \mathcal{D} \oplus \mathcal{D} \oplus \mathcal{D} \oplus \mathcal{T} \mid y_L - x_L = 0\}.$$

For FEA-2, the plaintext is completely fixed, so $G = \mathcal{T}$. Consequently, reduction modulo Z amounts to truncating the ciphertext to its left half.

In Section 7.4.1, the Frobenius norm of $\langle V, U\rangle_{\bar{\mathsf{F}}}$ was computed in the Fourier basis. Relative to the basis of functions $\mathbb{1}_z/|Z^1|$ with z in $(H \oplus G)/Z$, it is given by the squared Euclidean imbalance of $\mathsf{P}_Z((\mathsf{F}(\boldsymbol{x}), \boldsymbol{x}))$ with $\boldsymbol{x}$ uniform random on G. This was worked out explicitly in Example 3.12. Combining both results shows that if $X = (H \oplus G)/Z$ and $p(z) = \Pr[(\mathsf{F}(\boldsymbol{x}), \boldsymbol{x}) \equiv z \bmod Z]$, then

$$\left\| p - \frac{\mathbb{1}}{|X|} \right\|_X^2 = \sum_{\psi \neq \mathbb{1}} |c(\psi)|^2, \tag{7.1}$$

As the number of queries q increases, the empirical distribution approaches p and the χ^2/q statistic approaches the value $\sum_{\psi \neq \mathbb{1}} |c(\psi)|^2$. This shows that the χ^2-statistic can be interpreted as an alternative method to estimate the sum of the squared correlations $|c(\psi)|^2$ for $\psi \neq \mathbb{1}$ in Z^1. As discussed in the next section, this result suggests that the data complexity of the χ^2-distinguisher can be heuristically estimated as $\sqrt{|X|}/\sum_{\psi \neq \mathbb{1}} \mathsf{E}|\boldsymbol{c}(\psi)|^2$ with $\boldsymbol{c}(\psi)$ the correlation for a uniform random key and $|X| = N$ for the choices of Z discussed above.

Using the estimates of $\sum_{\psi \neq \mathbb{1}} \mathsf{E}|\boldsymbol{c}(\psi)|^2$ from Section 7.4.1, the data complexity of the χ^2-distinguishers for r-round FEA-1 and FF3-1 can be estimated as $N^{r/2-1.5}$. For FEA-2, the data complexity estimate becomes $N^{r/3-1.5}$. This is a significant improvement over the linear attacks from Section 7.3. Furthermore, by considering smaller choices of the group Z, it is still possible to set up χ^2-distinguishers even if only part of the ciphertext is available.

Finally, it is worthwhile making the link between $p(z)$ and $c(\psi)$ explicit. This was not necessary for the above discussion, but it will be useful in Section 7.5.

The following theorem generalizes a classical result for vector spaces G and H over $\mathbb{F}_2$ [17, 162]. Up to scaling, the left-hand side contains the coordinates of $\langle V, U\rangle_{\bar{\mathsf{F}}}$ relative to the basis functions $\mathbb{1}_z$ with z in $(H \oplus G)/Z$, whereas the right-hand side contains the coordinates relative to the basis Z^1.

Theorem 7.3. *Let $\mathsf{F} : G \to H$ be a function between finite commutative groups G and H. Let Z be a subgroup of the group $H \oplus G$ and let Z^1 be the group of characters of $H \oplus G$ with kernel containing Z. If $\boldsymbol{x}$ is a uniform random variable on G, then*

$$\Pr\left[(\mathsf{F}(\boldsymbol{x}), \boldsymbol{x}) \equiv z \bmod Z\right] = \frac{1}{|Z^1|} \sum_{\psi \in Z^1} C^{\mathsf{F}}_{\psi_H, \overline{\psi}_G}\, \psi(z)\,,$$

where ψ_H is the restriction of ψ to H and ψ_G similarly for G.

Proof. Note that $\psi(z)$ is well-defined for every z in $(H \oplus G)/Z$, because $\ker \psi \supseteq Z$. The relation between $\mathbb{1}_z$ and $\mathbb{1}_{Z^1}$ is given by the Fourier transformation:

$$\widehat{\mathbb{1}}_z(\psi) = \big(\mathscr{F}_{H\oplus G}\mathbb{1}_z\big)(\psi) = |Z|\,\psi(z)\mathbb{1}_{Z^1}(\psi)\,.$$

Let $\bar{\mathsf{F}}(x) = (\mathsf{F}(x), x)$. Since $\mathscr{F}_{H\oplus G}$ is unitary up to scaling and $\mathscr{F}_G\mathbb{1} = |G|\delta_{\mathbb{1}}$,

$$\left|\left\{x \in G \mid \bar{\mathsf{F}}(x) \in z\right\}\right| = \left\langle \mathbb{1}_z, T^{\bar{\mathsf{F}}}\mathbb{1}\right\rangle = \frac{1}{|H|}\left\langle \widehat{\mathbb{1}}_z, C^{\bar{\mathsf{F}}}\delta_{\mathbb{1}}\right\rangle = \frac{|Z|}{|H|}\sum_{\psi \in Z^1} \psi(z)\, C^{\bar{\mathsf{F}}}_{\psi,\mathbb{1}}\,.$$

Dividing by $|G|$ and using $|Z^1| = |H \oplus G|/|Z|$ yields

$$\Pr\left[(\mathsf{F}(\boldsymbol{x}), \boldsymbol{x}) \equiv z \bmod Z\right] = \frac{|Z|}{|H \oplus G|}\sum_{\psi \in Z^1} \psi(z)\, C^{\bar{\mathsf{F}}}_{\psi,\mathbb{1}} = \frac{1}{|Z^1|}\sum_{\psi \in Z^1} \psi(z)\, C^{\mathsf{F}}_{\psi_H, \overline{\psi}_G}\,,$$

The result follows from $C^{\bar{\mathsf{F}}}_{\psi,\mathbb{1}} = C^{\mathsf{F}}_{\psi_H, \overline{\psi}_G}$, which is due to $\psi \circ \bar{\mathsf{F}} = (\psi_H \circ \mathsf{F})\,\psi_G$. □

With the notation from above, Theorem 7.3 shows that

$$p(z) = \frac{1}{|Z^1|}\sum_{\psi \in Z^1} c(\psi)\psi(z)\,.$$

Equivalently, p is the inverse Fourier transform of c on the group $(H \oplus G)/Z$.

7.4.3 Cost analysis and experimental verification

As in Section 7.4.2, consider the χ^2-statistic for the empirical probability distribution of $(\mathsf{F}(\boldsymbol{x}), \boldsymbol{x})$ modulo Z, where $\boldsymbol{x}$ is a uniform random input

(consisting of the tweak T_R and possibly the right half of the plaintext). Before going into detailed calculations of the advantage of the distinguisher, the heuristic estimate that was used in the previous section will be derived.

Let $\boldsymbol{\chi}^2_{\mathsf{ideal}}$ be the χ^2-statistic when the true distribution is uniform random. This is a good model for the distribution that would be observed for an ideal tweakable block cipher. Likewise, denote the χ^2-statistic for the real cipher by $\boldsymbol{\chi}^2_{\mathsf{real}}$. It is well known that $\boldsymbol{\chi}^2_{\mathsf{ideal}}$ follows a χ^2 distribution with $N-1$ degrees of freedom when the number of queries q is sufficiently large. Hence, $\mathbb{E}\boldsymbol{\chi}^2_{\mathsf{ideal}} = N-1$. For $\boldsymbol{\chi}^2_{\mathsf{real}}$, the equality (7.1) yields

$$\mathsf{E}\boldsymbol{\chi}^2_{\mathsf{real}} = q \sum_{\psi \neq \mathbb{1}} \mathsf{E}\,|\widehat{\boldsymbol{c}}(\psi)|^2$$

where the average is taken with respect to a uniform random key and the random empirical correlations $\widehat{\boldsymbol{c}}(\psi)$ based on q samples. The expected value of $|\widehat{\boldsymbol{c}}(\psi)|^2$ for a fixed key is approximately equal to $|c(\psi)|^2 + 1/q$ when $|c(\psi)|^2$ is negligible compared to one. For a uniform random key, the true correlation $\boldsymbol{c}(\psi)$ is itself a random variable and hence

$$\mathsf{E}\boldsymbol{\chi}^2_{\mathsf{real}} \approx N - 1 + q \sum_{\psi \neq \mathbb{1}} \mathsf{E}\,|\boldsymbol{c}(\psi)|^2 \approx \mathsf{E}\boldsymbol{\chi}^2_{\mathsf{ideal}} + q \sum_{\psi \neq \mathbb{1}} \mathsf{E}\,|\boldsymbol{c}(\psi)|^2 .$$

By Theorem 1.1, to obtain a low false-positive rate, the decision threshold t should be larger than the standard deviation of $\boldsymbol{\chi}^2_{\mathsf{ideal}}$. That is, $t \geq \sqrt{2(N-1)}$. Hence, a constant advantage can be expected when $\mathsf{E}\boldsymbol{\chi}^2_{\mathsf{real}} - \mathsf{E}\boldsymbol{\chi}^2_{\mathsf{ideal}} \gg \sqrt{N}$. Equivalently,

$$q \gg \sqrt{N} \big/ \textstyle\sum_{\psi \neq \mathbb{1}} \mathsf{E}|\boldsymbol{c}(\psi)|^2 .$$

Since the main purpose of this section is to obtain accurate estimates of the advantage for concrete values of N, the above heuristic reasoning will not be formalized here.

It is relatively easy to estimate the average false-positive rate $P_{\mathsf{F}}(t)$ of the χ^2-distinguisher. Indeed, as mentioned above, the statistic $\boldsymbol{\chi}^2_{\mathsf{ideal}}$ follows a χ^2 distribution with $N-1$ degrees of freedom when the number of queries q is sufficiently large. Consequently,

$$P_{\mathsf{F}}(t) = \Pr\left[\boldsymbol{\chi}^2_{\mathsf{ideal}} \geq t\right] \approx 1 - \chi^2_{N-1}(t).$$

The average success-probability $P_{\mathsf{S}}(t)$ is significantly harder to compute. If $\boldsymbol{\chi}^2_{\mathsf{real}}$ denotes the χ^2-statistic for a random sample and a random key, then

$$P_{\mathsf{S}}(t) = \Pr\left[\boldsymbol{\chi}^2_{\mathsf{real}} \geq t\right].$$

To accurately estimate this probability, a Monte-Carlo approach was used to sample from $\boldsymbol{\chi}^2_{\mathsf{real}}$. Sampling from the correlation distribution can be done efficiently, provided that the dominant trail approximation is used. A detailed exposition of the sampling strategy is beyond the goals of this chapter, but an implementation is provided online[6].

Figure 7.5 shows the estimated maximum achievable advantage for the χ^2-distinguishers for full-round FEA-1 and FEA-2 with $N = 16$ and FF3-1 with $N = 1000$. The red dots correspond to experimental verifications of the advantage by performing each attack 512 times. These figures confirm the rough data complexity estimate of $N^{r/2-1.5}$.

7.5 Message recovery attacks

In this section, it is shown how the χ^2-distinguishers from Section 7.4 can be turned into message-recovery attacks. These attacks should be situated in the message-recovery security model of Bellare *et al.* [31]. Informally, this model assumes that the adversary is allowed to (non-adaptively) query the encryption of many *distinct* tweak-message pairs related to a secret message. The distinctness requirement is sufficient to ensure that a trivial guessing attack cannot achieve a nontrivial advantage.

Section 7.5.1 shows how the left-half of a message encrypted using FEA-1 or FF3-1 can be recovered. The assumptions of the attack are similar to previous work: the attacker is given the encryption of a target message and a second message with the same right half under many tweaks. Contrary to previous work [31,165], it is not necessary that both messages are encrypted under exactly the same set of tweaks. Instead, part of each tweak (T_L) must be constant. The data complexity of the attack is computed and experimentally verified in Section 7.5.2.

With more data, it is also possible to recover the right half of messages. This is discussed in Section 7.5.3. When combined with the left-half recovery attack, this results in recovery of entire messages. The same idea is used to extend the attacks to FEA-2.

7.5.1 Left-half recovery for FEA-1 and FF3-1

Consider FEA-1 or FF3-1 with a fixed plaintext input. In this scenario, the χ^2-distinguisher from Section 7.4.2 is still applicable by using only the left part

[6] `http://tim.cryptanalysis.info/fpe`

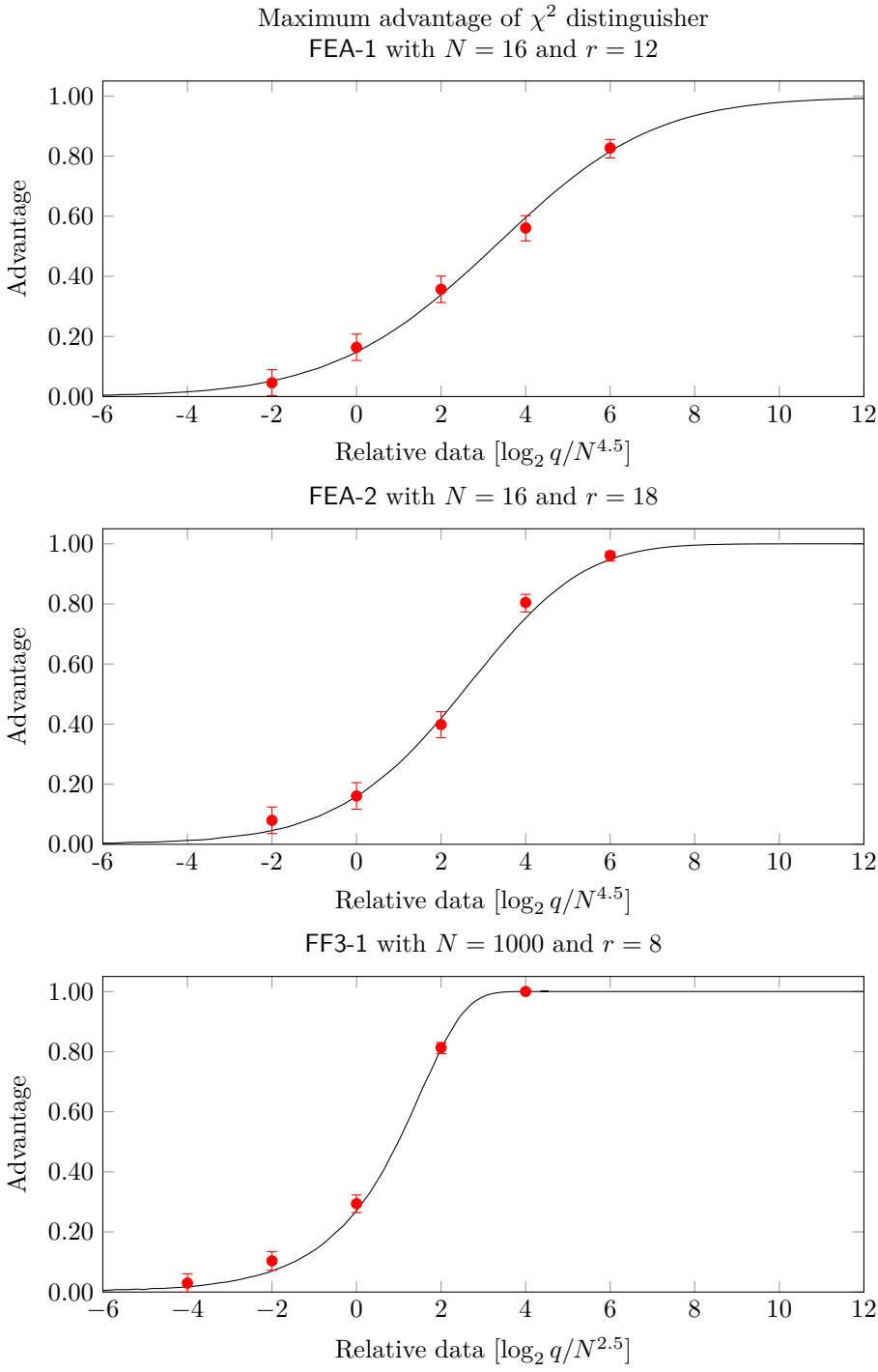


Figure 7.5: Theoretical and experimental maximum advantage of the χ^2-distinguishers for full-round FEA-1, FEA-2 and FF3-1. The error bars correspond to 95% Clopper-Pearson confidence intervals.

of the output. That is, $Z = \{(y_L, y_R, T_R) \in \mathcal{D} \oplus \mathcal{D} \oplus \mathcal{T} \mid y_L = 0\}$. The capacity of this multidimensional approximation is the same as before.

The idea behind the message-recovery attack is that a change in the plaintext affects the distribution of the left half of the ciphertext (for uniform random tweaks T_R) in a predictable way. Let $c_1(\psi)$ denote the correlation of the linear approximation corresponding to the character ψ when the plaintext is fixed to (x_L, x_R). Similarly, denote the correlation for a second plaintext (x'_L, x_R) by $c_2(\psi)$. Following the dominant trail approximation, $c_1(\psi)$ and $c_2(\psi)$ are well-approximated by the correlations of the trails given in Section 7.3. The two considered functions are the same up to the subtraction of a constant $\Delta = x_L - x'_L$ in the first round of the trail (the third round of the cipher). Hence,

$$c_2(\psi) \approx \psi_{\mathcal{D}}(\Delta) c_1(\psi)$$

with $\psi_{\mathcal{D}}$ the restriction of ψ to the half-domain $\mathcal{D}$. This approximation is accurate in practice, since the trails in Figures 7.2a and 7.3 are strongly dominant. Denote the probability distribution of the left half of the ciphertext in the first and second case by p_1 and p_2 respectively. Theorem 7.3 implies that

$$p_2(y_L) = \frac{1}{N} \sum_{\psi \in Z^1} c_2(\psi)\psi(y_L) \approx \frac{\psi_{\mathcal{D}}(\Delta)}{N} \sum_{\psi \in Z^1} c_1(\psi)\psi(y_L) = p_1(y_L + \Delta)\,.$$

In other words, the distributions p_1 and p_2 are (nearly) shifted over a distance Δ. It should be emphasized that this is a property of the ciphertext distributions and *not* of individual ciphertexts. As shown in Section 7.4.2, the distributions p_1 and p_2 are highly non-uniform. This is what makes it possible to recover Δ.

The message-recovery attack begins by estimating the probability distribution (for uniform random tweaks T_R) of the left half of the ciphertext twice: once for the secret plaintext (x_L, x_R) with fixed tweak T_L, and once for an arbitrary message (x'_L, x_R) with the same right half and for the same fixed tweak T_L. Next, for each candidate value $\Delta_{\mathtt{g}}$ for Δ, compute the statistic

$$r(\Delta_{\mathtt{g}}) = qN/4 \, \|\widehat{p}_1 - \widehat{p}_{\mathtt{g}}\|_2^2,$$

where $\widehat{p}_{\mathtt{g}}(y_L) = \widehat{p}_2(y_L - \Delta_{\mathtt{g}})$ with $\widehat{p}_1$ and $\widehat{p}_2$ the empirical estimates of p_1 and p_2 based on $q/2$ samples each. The statistics $r(\Delta_{\mathtt{g}})$ with $\Delta_{\mathtt{g}}$ in $\mathcal{D}$ can then be ranked in ascending order. If the number of samples used to obtain the empirical distributions is large enough, the values of $\Delta_{\mathtt{g}}$ corresponding to the top of the list are likely to be good candidates for Δ.

7.5.2 Cost analysis and experimental verification

The data complexity of the message-recovery attack can be estimated using a heuristic argument similar to the one that was used for the χ^2-distinguisher in Section 7.4.2. For a random sample, the statistic $\boldsymbol{r}(\Delta_{\mathrm{g}})$ satisfies

$$\boldsymbol{r}(\Delta_{\mathrm{g}}) = \frac{q}{4} \sum_{\psi \neq \mathbb{1}} |\widehat{\boldsymbol{c}}_1(\psi) - \overline{\psi_{\mathcal{D}}(\Delta_{\mathrm{g}})} \widehat{\boldsymbol{c}}_2(\psi)|^2,$$

where $\widehat{\boldsymbol{c}}_1(\psi)$ and $\widehat{\boldsymbol{c}}_2(\psi)$ are the empirical correlations and the sum is over all nontrivial ψ in Z^1. When the fixed-key correlation $|c_i(\psi)|^2$ is small, averaging over the sample gives $\mathsf{E}|\widehat{\boldsymbol{c}}_i(\psi)|^2 \approx |c_i(\psi)|^2 + 2/q$. Hence, the average of $\boldsymbol{r}(\Delta_{\mathrm{g}})$ over the sample and over a uniform random key is equal to

$$\begin{aligned}
\mathsf{E}\,\boldsymbol{r}(\Delta_{\mathrm{g}}) &= \frac{q}{4} \sum_{\psi \neq \mathbb{1}} \mathsf{E}\left(|\widehat{\boldsymbol{c}}_1(\psi)|^2 + |\widehat{\boldsymbol{c}}_2(\psi)|^2 - 2\Re\Big\{ \overline{\psi_{\mathcal{D}}(\Delta_{\mathrm{g}})}\, \widehat{\boldsymbol{c}}_1(\psi) \widehat{\boldsymbol{c}}_2(\psi) \Big\} \right) \\
&\approx \frac{q}{4} \sum_{\psi \neq \mathbb{1}} \left(\frac{4}{q} + \mathsf{E}|\boldsymbol{c}_1(\psi)|^2 + \mathsf{E}|\boldsymbol{c}_2(\psi)|^2 \right) - \frac{q}{2}\, \Re\Big\{ \sum_{\psi \neq \mathbb{1}} \overline{\psi_{\mathcal{D}}(\Delta_{\mathrm{g}})}\, \mathsf{E}\overline{\boldsymbol{c}_1(\psi)} \boldsymbol{c}_2(\psi) \Big\} \\
&\approx N - 1 + \frac{q}{2} \sum_{\psi \neq \mathbb{1}} \mathsf{E}|\boldsymbol{c}_1(\psi)|^2 - \frac{q}{2} \sum_{\psi \neq \mathbb{1}} \Re\big\{ \psi_{\mathcal{D}}(\Delta - \Delta_{\mathrm{g}}) \big\}\, \mathsf{E}|\boldsymbol{c}_1(\psi)|^2 \,.
\end{aligned}$$

where the third step follows from $c_2(\psi) \approx \psi_{\mathcal{D}}(\Delta) c_1(\psi)$. In fact, $\mathsf{E}|\boldsymbol{c}_1(\psi)|^2$ is nearly constant in ψ. If $\Delta_{\mathrm{g}} \neq \Delta$, then $\sum_{\psi \neq \mathbb{1}} \psi_{\mathcal{D}}(\Delta - \Delta_{\mathrm{g}}) = -1$ and it follows that

$$\mathsf{E}\,\boldsymbol{r}(\Delta_{\mathrm{g}}) - \mathsf{E}\,\boldsymbol{r}(\Delta) \approx q \sum_{\psi \neq \mathbb{1}} \mathsf{E}|\boldsymbol{c}_1(\psi)|^2 \,.$$

In particular, if $q \gg \sqrt{N} / \sum_{\psi \neq \mathbb{1}} \mathsf{E}|\boldsymbol{c}_1(\psi)|^2$, then $\mathsf{E}\,\boldsymbol{r}(\Delta_{\mathrm{g}}) - \mathsf{E}\,\boldsymbol{r}(\Delta) \gg \sqrt{N}$. This is sufficient to obtain a constant advantage since the standard deviation of $\boldsymbol{r}(\Delta_{\mathrm{g}})$ is of the order $\sqrt{N}$. This can be motivated by noting that, for a uniform output distribution, the distribution of $\boldsymbol{r}(\Delta_{\mathrm{g}})$ would be asymptotically χ^2 with $N - 1$ degrees of freedom. Hence, $\widetilde{\mathcal{O}}(N^{r/2-1.5})$ data should suffice to obtain a constant message-recovery advantage.

No attempt will be made here to make the above argument rigorous. Instead, accurate estimates of the message-recovery advantage for specific values of N can be computed using a Monte-Carlo approach. The main ingredient is a method to sample from the correlation distributions, which is identical to the one used for the calculations in Section 7.4.3. Results for full-round FEA-1 with $N = 16$ and FF3-1 with $N = 1000$ are shown in Figure 7.6, along with experimental estimates of the advantage.

Observe that for FF3-1 with $q = 4 \times \lfloor 2N^{2.5} \rceil \approx 2^{28}$, the theoretical advantage is an overestimate. This is due to the fact that only 2^{28} data is available for a fixed choice of the plaintext and tweak T_L. Once the variations in the ideal distribution (which was assumed to be uniform in the analysis) are of the same order as the sampling error, the advantage begins to flatten off. However, this does not imply that the advantage of the FF3-1 message-recovery attack cannot be made close to one. Indeed, one can simply perform the attack for a different choice of T_L. Of course, for even larger N, the maximum advantage that can be achieved using one choice of T_L decreases and the attack eventually becomes infeasible. Based on the estimated data complexity of the attack and Figure 7.6, this is expected to occur for $N > 2^{12}$. The right-half recovery attack from Section 7.5.3 avoids this problem and can be used for all $N < 2^{19}$, but it has a higher overall data complexity.

7.5.3 Right-half recovery and application to FEA-2

The left-half recovery attack on FEA-1 and FF3-1 could also be applied for two messages (x_L, x_R) and (x'_L, x'_R) with $x_R \neq x'_R$. However, the recovered difference would then be $\Delta = x_L - x'_L + \mathsf{F}_1(x_R) - \mathsf{F}_1(x'_R)$. If $x_L - x'_L$ is known, then the adversary can recover Δ to obtain the difference $\mathsf{F}_1(x_R) - \mathsf{F}_1(x'_R)$. This is useful because it leads to a right-half recovery attack. In addition, the output differences will be directly used in the key-recovery attack on FEA-1 that is described in Section 7.6. It is also possible to apply the same attack with a different choice of Z that includes the left half of the plaintext. In this case, the recovered difference would simply be $\mathsf{F}_1(x_R) - \mathsf{F}_1(x'_R)$ due to reduction modulo Z. The main advantage of this approach is that it increases the amount of available data per choice of the right half by a factor of N. This extends the reach of the attack to $N < 2^{19}$, compared to $N < 2^{12}$ for left-half recovery.

The right-half can be recovered by guessing x'_R until the recovered difference is zero. This does not violate the distinctness requirement of the message-recovery framework, since the tweaks T_R and the left halves of the guessed messages can be different from those of the secret message. The attack proceeds by computing the statistics $r(0)$ from Section 7.5.1 with $\widehat{p}_1$ the empirical distribution for the secret message and $\widehat{p}_2$ the empirical distribution with right-half guess x'_R. If these statistics are ranked in ascending order, the values of x'_R corresponding to the top of the list are the most promising candidates for x_R. By the analysis in Section 7.5.2, this attack requires $\widetilde{\mathcal{O}}(N^{r/2-0.5})$ data. A simulation of the maximum advantage is shown in the bottom of Figure 7.6, along with experimental results. Note that the error bars are wider than for the left-half recovery experiments because each data point was estimated using only 40 runs of the attack (to limit the time complexity of the experiment).

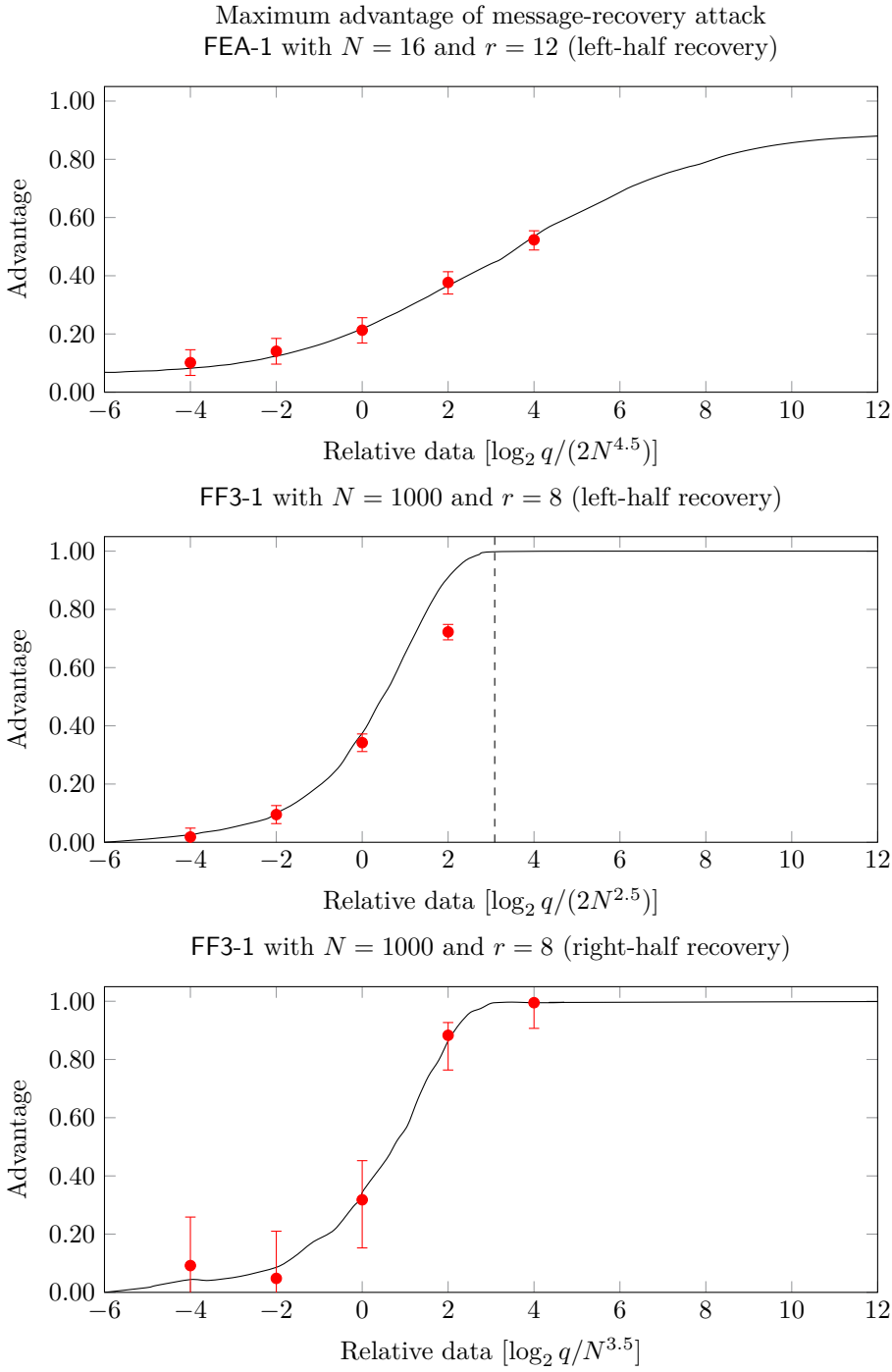


Figure 7.6: Theoretical and experimental maximum advantage of the message-recovery attacks for full-round FEA-1 and FF3-1. The error bars correspond to 95% Clopper-Pearson confidence intervals. The dashed vertical line corresponds to a data complexity of 2×2^{28}.

The same idea as above can be used to extend the message-recovery attack to FEA-2. For example, consider left-half recovery. In this case, the adversary queries the encryption of the secret message (x_L, x_R) under many tweaks with constant T_L. In addition, for each guess of x'_L, similar queries are made for (x'_L, x_R). The same process as above can be used to identify the values of x_L for which

$$\mathsf{F}_2(x_L + \mathsf{F}_1(x_R)) + x_R = \mathsf{F}_2(x'_L + \mathsf{F}_1(x_R)) + x_R\,.$$

However, there is an additional issue that must be addressed: since the approximation shown in Figure 7.2b does not have equal input and output masks, the effect of changing the plaintext input on the correlations is more complicated. Nevertheless, one can still use the same approach (with roughly the same data complexity) to check for equality between the two output distributions.

7.6 Key-recovery attack on FEA-1

This section shows how the left-half message-recovery attack on FEA-1 from Section 7.5.1 can be used for key-recovery. Naturally, the attack heavily depends on the internal details of the round function F_1. For FF3-1, key-recovery is not feasible since the round functions are truncations of the AES.

The FEA-1 round function is illustrated in Figure 7.7. It consists of two iterations of a key-addition layer, an S-box layer and a linear layer with branch number nine. Each of these layers acts on a state in a vector space $\mathbb{F}_{2^8}^8$. The round keys will be denoted by K_a and K_b. The round function F_1 is defined as the truncation of this structure to m bits.

The exact choice of the matrix representation M of the linear layer is not important. The only property of M that will be used is the fact that it has branch number nine (equivalently, is MDS). The S-box is based on inversion in $\mathbb{F}_{2^8}$, but the details are not important. However, it is important that for all nonzero Δ_1 and Δ_2, the equation $\mathsf{S}(x + \Delta_1) = \mathsf{S}(x) + \Delta_2$ has either no, two or four solutions in x. For each $\Delta_1 \neq 0$, the case with four solutions occurs for exactly one choice of Δ_2.

Recall from Section 7.5.3 that it is possible to recover output differences $\mathsf{F}_1(P) + \mathsf{F}_1(P')$ for an arbitrary choice of P and P'. The idea behind the key-recovery attack is to guess parts of the internal state of the round function and to check the validity of these guesses using such output differences. After recovering the relevant parts of the internal state, the round keys can be recovered.

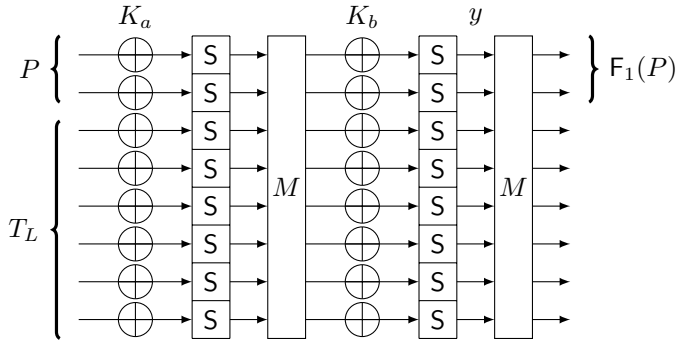


Figure 7.7: Round function of FEA-1 with round keys K_a and K_b.

Let x denote the first byte of the round function input $P\|T_L$. Observe that byte i of the internal state y (indicated in Figure 7.7) can be written as

$$y_i = \mathsf{S}(\gamma_i + M_{i,1}\,\mathsf{S}(K_{a,1} + x))\,,$$

where $\gamma_1, \ldots, \gamma_8$ in $\mathbb{F}_{2^8}$ are constants depending on the round keys K_a and K_b (but not on the first byte $K_{a,1}$) and on the tweak T_L. Importantly, $\gamma_1, \ldots, \gamma_8$ do not depend on x. Specifically,

$$\gamma_i = K_{b,i} + \sum_{j=2}^{8} M_{i,j}\mathsf{S}([P\|T_L]_j + K_{a,j})\,.$$

In Section 7.6.1, it will be shown how $K_{a,1}$ and γ_i can be recovered using a limited number of output differences. Section 7.6.2 then shows how the entire round keys K_a and K_b can be extracted from these constants and a few additional output differences.

7.6.1 Recovering $K_{a,1}$ and the internal constants γ_i

It is clear from Figure 7.7 that the output difference is a linear function of the difference between the internal states y and y' (corresponding to two inputs x and x'). Furthermore, since M is an invertible matrix, this function is of rank m. Hence, $y + y'$ can take $2^{64-m} = 2^{64}/N$ possible values. By computing an echelon form for the linear function that maps $y + y'$ to the output difference, these candidate solutions can easily be enumerated. For each guess of $y + y'$, one obtains the values

$$y_i + y'_i = \mathsf{S}(\gamma_i + M_{i,1}\mathsf{S}(K_{a,1} + x)) + \mathsf{S}(\gamma_i + M_{i,1}\mathsf{S}(K_{a,1} + x'))\,.$$

For each $i = 1, \ldots, 8$, one can determine the set of possible input differences $\mathsf{S}(K_{a,1} + x) + \mathsf{S}(K_{a,1} + x')$ that can lead to the known difference $y_i + y'_i \neq 0$. Due the properties of S, there are 127 possible input differences. Hence, each i potentially reduces the number of candidate differences by a factor $127/255 < 1/2$. Experimentally, it is found that the difference $\mathsf{S}(K_{a,1} + x) + \mathsf{S}(K_{a,1} + x')$ can be uniquely determined for over 85% of the output differences $y + y'$. Since the difference $x + x'$ is known, two candidates for $K_{a,1}$ can be computed from the difference equation. The case with four solutions is unlikely to occur and does not significantly affect the overall time and data complexity of the attack.

Once $K_{a,1}$ has been determined (as one of two possible values), the constants γ_i can also be obtained by solving a difference equation. In particular, since the case with four solutions is rare, one usually ends up with two candidates for each γ_i. To check the validity of these candidates, additional output differences will be used. To save data, one of x or x' can be reused. For each of the 2^9 candidate values, the expected output difference should then be computed and compared to the observed difference. This requires roughly 2^{12} S-box evaluations. If the candidate values are wrong, the output difference will match in roughly $1/N$ of the cases. Hence, the computational cost is dominated by the calculation of the expected output difference for the first pair.

The total number of candidates for the difference $y + y'$, the internal constants and the first byte of K_a is $2^{64+9}/N = 2^{73}/N$. Hence, $\lceil 73/m - 1 \rceil$ pairs are sufficient to obtain a unique solution. For $m = 4$, the number of available input differences is too small to obtain a unique candidate. However, this is not a major issue since the time complexity of the round key recovery procedure described in Section 7.6.2 is small enough that it can be repeated several times.

The data complexity of the above process is $(\lceil 73/m - 1 \rceil + 1)q/2$ queries, where q is the data complexity of the left-half message-recovery attack. This comes with an equal computational cost, measured in FEA-1 evaluations. The remaining computational cost is dominated by $2^{64+12}/N$ S-box evaluations. Since the cipher contains 12×16 S-boxes, one can conservatively estimate that this takes less time than $2^{68}/N$ evaluations of full-round FEA-1.

7.6.2 Recovering the round keys

Once the constants $\gamma_1, \ldots, \gamma_8$ have been recovered, obtaining the round keys K_a and K_b is relatively easy. In particular, recall that

$$\gamma_i = K_{b,i} + \sum_{j=2}^{8} M_{i,j} \mathsf{S}([P \| T_L]_j + K_{a,j}) \,.$$

Suppose $P\|T_L$ and $P'\|T'_L$ differ only in byte $j \in \{2, \ldots, 8\}$ and let γ'_i be the new value of γ_i for input $P'\|T'_L$. It is easy to see that

$$\gamma_i + \gamma'_i = M_{i,j}\mathsf{S}([P\|T_L]_j + K_{a,j}) + M_{i,j}\mathsf{S}([P'\|T'_L]_j + K_{a,j})\,.$$

Hence, after guessing $K_{a,j}$, one can compute the new constants γ'_i and the expected output differences for pairs with tweak T'_L. To obtain a unique (up to a constant) candidate for $K_{a,j}$, a total of $\lceil 8/m \rceil$ differences are sufficient. Recovering all of the bytes of K_a thus requires $7 \times \lceil 8/m \rceil$ differences. Once K_a is recovered, K_b can be computed directly.

To conclude, the data complexity of this step is $7q/2 \times (\lceil 8/m \rceil + 1)$ with q the data complexity of the left-half message-recovery attack. A few additional pairs will be required to filter spurious candidates for $K_{a,j}$, or if no unique solution for the constants $\gamma_1, \ldots, \gamma_8$ was obtained in the first step of the attack ($m = 4$). The time complexity, excluding the time required for message-recovery, is negligible compared to that of the first step.

7.6.3 Recovering all round keys

By the results in Sections 7.6.1 and 7.6.2, the round keys K_a and K_b of the first round function can be recovered using at most $\lceil 73/m - 1 \rceil + 7\lceil 8/m \rceil \leq 16\lceil 8/m \rceil$ evaluations of the left-half message-recovery attack and additional time equivalent to at most $2^{68}/N$ FEA-1 evaluations. If q is the amount of data required for the left-half recovery attack, this amounts to a total of less than $8\lceil 8/m \rceil q + 4q$ queries. However, the FEA-1 key-schedule is a Lai-Massey structure that generates two round keys per iteration. Hence, the remaining round keys can not be obtained by iterating the key-schedule without knowing the round keys for the second round. To obtain these keys, it suffices to perform the same key-recovery attack on F_2. Hence, the total cost is less than $16\lceil 8/\log_2 N \rceil q + 8q$ data for left-half recoveries and additional time equivalent to less than $2^{69}/N$ evaluations of FEA-1.

8

Reevaluation of differential attacks

Chapter 4 introduced quasidifferential trails and showed that they can be used to estimate fixed-key probabilities of differential characteristics and differentials using the dominant trail approximation. This chapter applies quasidifferential trails to the analysis of differential attacks on the block ciphers Rectangle and Speck, and the hash function KNOT. The analysis is automated and applicable to other constructions. Several attacks are shown to be invalid, most others turn out to work only for some keys but can be improved for weak keys.

Like Chapter 4, this chapter is based on the paper "Differential cryptanalysis in the fixed-key model" [56] from Crypto 2022 (joint work with Vincent Rijmen). However, unlike in Chapter 4, the focus is on applications rather than theory.

8.1 Introduction

The propagation of differences alone is not sufficient to analyze the probability of differentials. To bypass this issue, Lai, Massey and Murphy [191] introduced the hypothesis of stochastic equivalence. However, significant deviations from this hypothesis were demonstrated early on by Knudsen [180] for the characteristics used in the differential analysis of DES. Additional examples were given in the introduction of Chapter 4. Experiments such as those of Ankele and Kölbl [12] and Heys [163] further suggest that such deviations are the norm rather than the exception.

Chapter 4 used the general principles from Chapter 2 to provide a complete description of differential cryptanalysis in the form of quasidifferential trails. The sum of the correlations of all quasidifferential trails corresponding to a characteristic is equal to the probability of the characteristic. Quasidifferential trails can also be used to show that a characteristic is impossible – for instance using Theorem 4.5. Finally, it was shown that Knudsen's observations on the key-dependency of differential characteristics in DES can be explained economically using quasidifferential trails.

The purpose of this chapter is to demonstrate the applicability of quasidifferential trails as a practical tool. Section 8.2 shows how to automate the search for trails

using Satisfiability Modulo Theories (SMT), but other popular methods such as integer linear programming are also suitable. For the analysis of AndRX (such as Simon) and ARX ciphers (such as Speck), the quasidifferential transition matrices of bitwise-and and modular addition are determined explicitly.

Section 8.3 analyzes differential attacks on Rectangle [292]. The main conclusion is that the best published key-recovery attack on round-reduced Rectangle does not work, although it can be modified to obtain a valid weak key attack. In addition, the probability of 'optimal' differentials is shown to depend strongly on the key.

Several differential attacks on KNOT [293], a second-round candidate in the NIST lightweight cryptography competition, are reevaluated in Section 8.4. It is shown that the forgery and collision attacks of Zhang *et al.* [294] do not work, because the characteristics they rely on have probability zero. At the same time, it is shown that their probabilities are two orders of magnitude larger for some choices of the round constants.

Section 8.5 reevaluates the best published attacks on Speck. Most of the attacks that were analyzed only work for a subset of keys. However, for weak keys, attacks with lower data complexity can be obtained. In addition, the experimental results of Ankele and Kölbl are explained by taking into account one additional quasidifferential trail.

8.2 Modelling quasidifferential trails

As in Chapter 6, the dual group of $\mathbb{F}_2^n$ will be identified with $\mathbb{F}_2^n$ in the usual way. Hence, a quasidifferential trail is equivalent to a sequence of mask-difference pairs $(u_1, a_1), \ldots, (u_r, a_r)$.

The SMT model[1] for finding quasidifferential trails corresponding to a given differential characteristic is similar to existing models for finding linear trails. The model expresses the correlation of a trail by its negative base-2 logarithm or *weight*. In fact, since the correlation of any quasidifferential trail is at most as large as the key-averaged probability of the corresponding differential characteristic, these weights are expressed relative to the weight of the characteristic. To solve the SMT problem, Boolector [222] is used through its Python interface 'pyboolector'.

The following two sections discuss how S-boxes and modular additions can be modelled in practice. Note that linear functions are easy to model using

[1] Source code is available `https://github.com/TimBeyne/quasidifferential-trails`

Theorem 4.2 (3). That is, differences propagate as in ordinary differential cryptanalysis and masks propagate as in linear cryptanalysis.

8.2.1 S-boxes

The propagation over the S-box layer is modelled by conditions in disjunctive normal form corresponding to the relative weights of the entries of the quasidifferential transition matrix. The number of constraints can often be reduced using minimization algorithms such as Quine-McCluskey, but this was not used for the applications in this chapter.

For small (such as 4- or 8-bit) S-boxes, the quasidifferential transition matrix can be computed using the algorithm from Section 4.4.1. In particular, the quasidifferential change-of-basis transformation $\mathscr{Q}_{\mathbb{F}_2^n}$ satisfies

$$\mathscr{Q}_{\mathbb{F}_2^n} = \mathscr{Q}_{\mathbb{F}_2}^{\otimes n} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{pmatrix}^{\otimes n} .$$

As explained in Section 4.4.1, for any such matrix, there exists an efficient algorithm to compute the matrix-vector product.

8.2.2 Bitwise-and and modular addition

Several ciphers use bitwise-and or modular addition as their nonlinear components. Although these functions potentially have many input and output bits, they are highly structured. This makes it possible to express the entries of their quasidifferential transition matrix using relatively simple logical constraints.

In the following, the bitwise-and of x and y in $\mathbb{F}_2^n$ will be denoted by $x \wedge y$, the bitwise-or by $x \vee y$, and $\mathsf{and}(x \| y) = x \wedge y$. The bitwise complement of x will be written as $\bar{x}$. The addition of the integers represented by x and y modulo 2^n will be denoted by $\mathsf{add}(x \| y)$. Finally, let $\leq$ denotes the monoid order on $\mathbb{F}_2^n$.

The quasidifferential transition matrix of $\mathsf{and} : \mathbb{F}_2^{2n} \to \mathbb{F}_2^n$ is easy to compute because it acts on each pair of bits independently. Hence, Corollary 4.1 (1) can be used. This results in the following theorem.

Theorem 8.1. *Let a, b and c in $\mathbb{F}_2^n$ be differences and u, v and w in $\mathbb{F}_2^n$ masks. It holds that $D^{\mathsf{and}}_{(w,c),\,(u\|v,a\|b)} \neq 0$ if and only if $c \leq a \vee b$, $u \vee v \leq a \vee b \vee w$ and*

$a \wedge u + b \wedge v = c \wedge w$. *Furthermore, if these conditions hold, then*

$$D^{\mathsf{and}}_{(w,c),\,(u\|v,a\|b)} = 2^{-\,\mathsf{wt}(a\vee b)-\mathsf{wt}(w\wedge\bar{a}\wedge\bar{b})}\,(-1)^{u^{\mathsf{T}}(\bar{a}\wedge c)+v^{\mathsf{T}}(a\wedge c)+u^{\mathsf{T}}(a\wedge b)}\,.$$

Proof. Let $\mathsf{and}_n : \mathbb{F}_2^{2n} \to \mathbb{F}_2^n$ denote the n-bit bitwise-and function defined by

$$\mathsf{and}_n(x_1\|y_1\|x_2\|y_2\|\cdots\|x_n\|y_n) = (x_1y_1, x_2y_2, \ldots, x_ny_n)\,.$$

By Corollary 4.1 (1), it holds that $D^{\mathsf{and}_n} = \big(D^{\mathsf{and}_1}\big)^{\otimes n}$. By (4.1), it holds that

$$D^{\mathsf{and}_1}_{(w_i,c_i),(u_i\|v_i,a_i\|b_i)} = \frac{1}{4} \sum_{\substack{x,y\,\in\,\mathbb{F}_2 \\ b_ix+a_iy=a_ib_i+c_i}} (-1)^{u_ix+v_iy+w_ixy}\,.$$

The above sum can be computed case-by-case. For $a_i = b_i = 0$, the sum equals

$$\begin{aligned} D^{\mathsf{and}_1}_{(w_i,c_i),(u_i\|v_i,0\|0)} &= \delta_0(c_i)\big(1 + (-1)^{u_i} + (-1)^{v_i} + (-1)^{u_i+v_i+w_i}\big)/4 \\ &= \begin{cases} 1/2^{w_i} & \text{if } c_i = 0 \text{ and } u_i \vee v_i \le w_i\,, \\ 0 & \text{otherwise}\,. \end{cases} \end{aligned}$$

If $a_i = 1$ and $b_i = 0$, then

$$\begin{aligned} D^{\mathsf{and}_1}_{(w_i,c_i),(u_i\|v_i,1\|0)} &= \delta_0(c_i)\,\big(1 + (-1)^{u_i}\big)/4 + (-1)^{v_i}\,\delta_1(c_i)\,\big(1 + (-1)^{u_i+w_i}\big)/4 \\ &= (-1)^{v_ic_i}/2\,\delta_{c_i\wedge w_i}(u_i)\,. \end{aligned}$$

The remaining two cases are analogous and yield

$$D^{\mathsf{and}_1}_{(w_i,c_i),(u_i\|v_i,0\|1)} = (-1)^{u_ic_i}/2\,\delta_{c_i\wedge w_i}(v_i)$$

$$D^{\mathsf{and}_1}_{(w_i,c_i),(u_i\|v_i,1\|1)} = (-1)^{u_i\bar{c}_i}/2\,\delta_{c_i\wedge w_i}(u_i+v_i)\,.$$

Combining the cases above, one obtains that $D^{\mathsf{and}_1}_{(w_i,c_i),(u_i\|v_i,a_i\|b_i)} \neq 0$ if and only if $c_i \le a_i \vee b_i$, $u_i \vee v_i \le a_i \vee b_i \vee w_i$ and $a_i \wedge u_i + b_i \wedge v_i = c_i \wedge w_i$. Furthermore, under these conditions,

$$D^{\mathsf{and}_1}_{(w_i,c_i),(u_i\|v_i,a_i\|b_i)} = (-1)^{a_ic_iv_i+b_ic_iu_i+a_ib_i\bar{c}_iu_i}2^{-(a_i\vee b_i)-(w_i\wedge\bar{a}\wedge\bar{b})}\,.$$

Finally, note that $b_ic_i + a_ib_i\bar{c}_i = a_ib_i + \bar{a}_ic_i$ since $c_i \le a_i \vee b_i$. □

The quasidifferential transition matrix of $\mathsf{add} : \mathbb{F}_2^{2n} \to \mathbb{F}_2^n$ can be computed using its CCZ-equivalence to a quadratic function [248] similar to bitwise-and. This result is reproduced in Theorem 8.2 below. Two functions $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ and $\mathsf{G} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ are CCZ-equivalent if their graphs $\{(x, \mathsf{F}(x)) \mid x \in \mathbb{F}_2^n\}$ and $\{(x, \mathsf{G}(x)) \mid x \in \mathbb{F}_2^n\}$ are related by an invertible $\mathbb{F}_2$-affine transformation [81].

Theorem 8.2 (Schulte-Geers [248, Theorem 1])**.** *Let* $\mathsf{M} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ *be the linear map defined* $\mathsf{M}(x)_1 = 0$ *and* $\mathsf{M}(x)_i = \sum_{j=1}^{i-1} x_j$ *for* $i > 1$ *The function* $\mathsf{Q} : (x, y) \mapsto \mathsf{M}(x \wedge y)$ *is CCZ-equivalent to modular addition with modulus* 2^n *under the linear map* $(x, y, z) \mapsto (x + z, y + z, x + y + z)$.

In Theorem 8.3, $\mathsf{M}^\dagger$ is a near-inverse of M given by $\mathsf{M}^\dagger(x) = [x + (x \ll 1)] \gg 1$, where $\ll$ and $\gg$ denote left and right shifts respectively.

Theorem 8.3. *Let* a, b *and* c *in* $\mathbb{F}_2^n$ *be differences and* u, v *and* w *in* $\mathbb{F}_2^n$ *masks. It holds that* $D^{\mathsf{add}}_{(w,c),(u\|v,a\|b)} \neq 0$ *if and only if*

$$c'_1 = 0$$

$$\mathsf{M}^\dagger c' \leq a' \vee b'$$

$$u' \vee v' \leq a' \vee b' \vee \mathsf{M}^\mathsf{T} w'$$

$$a' \wedge u' + b' \wedge v' = c' \wedge \mathsf{M}^\mathsf{T} w'$$

$$(a'_n = b'_n = 0) \vee (a'_n u'_n + b'_n v'_n \neq w'_n) \vee (a'_n v'_n = \bar{a}'_n u'_n)\,,$$

where $(a', b', c') = (b+c, a+c, a+b+c)$ *and* $(u', v', w') = (u+w, v+w, u+v+w)$. *Furthermore, if the above conditions hold, then*

$$D^{\mathsf{add}}_{(w,c),\,(u\|v,a\|b)} = 2^{z - \mathsf{wt}(a' \vee b') - \mathsf{wt}(\mathsf{M}^\mathsf{T} w' \wedge \bar{a}' \wedge \bar{b}')}\,(-1)^{(\bar{a}' \wedge \mathsf{M}^\dagger c' + a' \wedge b')^\mathsf{T} u' + (a' \wedge \mathsf{M}^\dagger c')^\mathsf{T} v'}\,,$$

where $z = (a'_n \vee b'_n) \wedge (a'_n u'_n + b'_n v'_n = w'_n) \wedge (a'_n v'_n \neq \bar{a}'_n u'_n)$.

Proof. Let $\mathsf{Q}(x, y) = \mathsf{M}(x \wedge y)$. By Corollary 4.1 (2) and Theorem 4.2 (3), the quasidifferential transition matrix of Q satisfies

$$D^{\mathsf{Q}}_{(w,c),(u\|v,a\|b)} = \sum_{d \in \mathsf{M}^{-1}(c)} D^{\mathsf{and}}_{(\mathsf{M}^\mathsf{T} w, d),(u\|v,a\|b)}\,,$$

where the sum is over all preimages of c. If some d in $\mathbb{F}_2^n$ satisfies $\mathsf{M}(d) = c$, then $c_1 = 0$ by the definition of M. Furthermore, one can check that $d_i = c_i + c_{i+1}$ for all $i \leq n - 1$. The value of d_n is arbitrary. Hence, if $c_1 = 0$, we can write

$$D^{\mathsf{Q}}_{(w,c),(u\|v,a\|b)} = D^{\mathsf{and}}_{(\mathsf{M}^\mathsf{T} w,\, \mathsf{M}^\dagger c),(u\|v,a\|b)} + D^{\mathsf{and}}_{(\mathsf{M}^\mathsf{T} w,\, \mathsf{M}^\dagger c + e_n),(u\|v,a\|b)}\,,$$

where $e_n = (0, 0, \ldots, 0, 1)$. Theorem 8.1 can now be applied to each of the terms above. We now write the second term in terms of the first. Compared to the first term, the conditions for the second term to be non-zero additionally

include $a_n \vee b_n = 1$ and $a_n u_n + \bar{a}_n v_n = w_n$. In addition, the sign of both terms (if nonzero) differs by a factor $(-1)^{a_n v_n + \bar{a}_n u_n}$. Hence,

$$D^{\mathsf{Q}}_{(w,c),(u\|v,a\|b)} = \delta_1(c_1)\Big(1 + (-1)^{a_n v_n + \bar{a}_n u_n}\delta_1(a_n \vee b_n)\delta_{w_n}(a_n u_n + b_n v_n)\Big)$$
$$\times D^{\mathsf{and}}_{(\mathsf{M}^{\mathsf{T}}w,\, \mathsf{M}^{\dagger}c),(u\|v,a\|b)} ,$$

In order to compute D^{add}, a variant of Theorem 4.2 (3) is needed. Specifically,

$$D^{\mathsf{add}}_{(w,c),(u\|v,a\|b)} = D^{\mathsf{Q}}_{(w',c'),(u'\|v',a'\|b')} ,$$

where $w' = u + v + w$, $u' = u + w$, $v' = v + w$, $c' = a + b + c$, $a' = b + c$ and $b' = a + c$. The result follows by using the expression for the coordinates of D^{Q} that was derived above. □

8.3 Differential attacks on Rectangle

There are several reasons why Rectangle is an interesting target to illustrate the use of quasidifferential trails. The linear layer is a bit-permutation and simpler compared to similar ciphers such as PRESENT [71]. As discussed in Section 4.5.3, the self-duality of bit-permutations potentially results in quasidifferential trails with high absolute correlation relative to the probability of the corresponding differential trail. In addition, differential cryptanalysis is the dominant attack for Rectangle. The optimal differentials for Rectangle also have a limited differential effect, *i.e.* they contain few high-probability characteristics. This simplifies the analysis.

8.3.1 Specification of Rectangle

Rectangle [292] is a 64-bit substitution-permutation network, with a nonlinear layer consisting of 4-bit S-boxes and a bit-permutation as the linear layer. The state is typically represented by a 4×16 array of bits. The Rectangle round function consists of three simple operations, as illustrated in Figure 8.1.

Round-key addition. The round key bits are added to the state bits. The round keys are derived using a key-schedule based on a generalized Feistel construction. The master key is either 80 or 128 bits long. The details of this key-schedule will not be discussed here.

S-box layer. Each column of the state is transformed by a 4-bit permutation S. The S-box S is given in Table 4.1. The absolute values of the entries of

the matrix D^{S} were already illustrated in Figure 4.1. The topmost bit of each state column in Figure 8.1 corresponds to the least significant bit of the S-box input- and output values.

Linear layer. The second row (from the top) is rotated by one position to the left. The third row and the fourth row are rotated by 12 and 13 positions to the left respectively.

Rectangle repeats these steps for a total of 25 rounds, followed by a final round key addition.

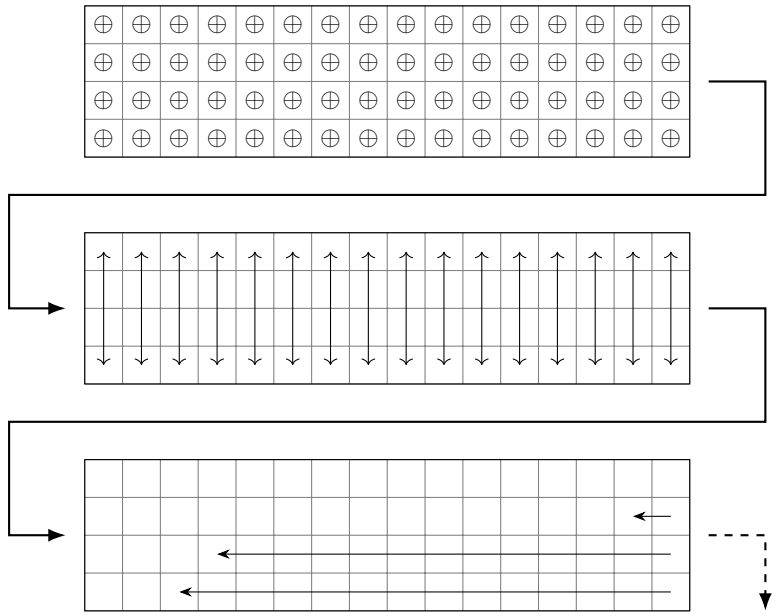


Figure 8.1: The round function of Rectangle. From top to bottom: the round key addition, the S-box layer and the linear layer consisting of a rotation of the bottom three rows.

To perform the analysis in this section, an SMT model of the propagation of quasidifferential trails in Rectangle was built as described in Section 8.2.

8.3.2 Differentials

Table 8.1 lists several differentials for Rectangle. Differential i is a 14-round differential used in the best published key-recovery attack on Rectangle [292].

Although its probability is suboptimal, its input and output differences are better suited for key-recovery. The corresponding 18-round key-recovery attack requires 2^{64} data and enough memory to hold 2^{72} counters. The time complexity amounts to $2^{78.67}$ (80-bit key) or $2^{126.66}$ (128-bit key) 18-round encryptions. A success probability of 67% is claimed.

Differential ii has a dominant characteristic with average probability[2] 2^{-61}. Based on the analysis of the designers (which included differential effects), this differential is believed to have a maximal average probability. Up to rotational equivalence, there are a total of 32 such differentials. However, as discussed below, these differentials all have similar behavior.

The average probability of differential iii is suboptimal, but the analysis in Section 8.3.3 shows that its probability is much larger for some keys.

Table 8.1: Differentials (a, b) for 14 rounds of Rectangle. The column p_{avg} gives an estimate of the average differential probability for independent round keys.

a	b	p_{avg}	Comment	№
0020000600000000	0004000000000020	$2^{-63} + 2^{-66}$	Key-recovery	i
0100007000000000	0861008400000010	$2^{-61} + 2 \cdot 2^{-64}$	'Optimal'	ii
00000000c0000600	0004000000000020	$2 \cdot 2^{-65} + 13 \cdot 2^{-68}$	'Suboptimal'	iii

8.3.3 Analysis

For completeness, the dominant characteristics for differentials i to iii are listed in Tables 8.2 to 8.4. In order to search for optimal quasidifferential trails, the propagation of the masks for fixed differences is modelled as an SMT problem.

Differential i. The two dominant characteristics for this differential are listed in Table 8.2. The first two columns of Table 8.5 list the number of quasidifferential trails of each absolute correlation for these two characteristics.

Any characteristic has at least one quasidifferential trail with correlation equal to its average probability p_{avg}, namely the trail with all-zero masks. The fact that the first characteristic has two quasidifferential trails with correlation $\pm p_{\mathsf{avg}}$ and the second four, is special. Table 8.6 shows two of these trails (one for each characteristic) with the same masks. Only rounds 9 to 12 are shown, since the masks are zero in all other rounds. Hence, these two trails describe a

[2] Average probability for independent and uniform random round keys.

Table 8.2: Characteristics in differential i.

$p_{\mathsf{avg}} = 2^{-63}$	$p_{\mathsf{avg}} = 2^{-66}$
..2....6........	..2....6........
..6....2........	..6....2........
.2....6.........	.2....6.........
.6....2.........	.6....2.........
2....6..........	2....6..........
6....2..........	6....2..........
....6.........2	....6.........2
....2.........6	....2.........6
...6.........2.	...6.........2.
...2.........6.	...2.........6.
..6.........2..	..6.........2..
..2.........6..	..2.........6..
.6.........2...	.6.........2...
.2.........6...	.2.........6...
6.........2....	6.........2....
2.........6....	2.........6....
..........2....6	..........2....6
..........6....2	..........6....2
.........2....6.	.........2....6.
.........c....2.	.........c....2.
............86..	............86..
............12..	............92..
............3...	............3..8
............8...	............8..1
..............8	9
..............1	1
..............1	1
..............6	6
...4.........2.	...4.........2.

Table 8.3: Characteristics in differential iii.

$p_{\mathsf{avg}} = 2^{-65}$	$p_{\mathsf{avg}} = 2^{-65}$
........c....6..	........c....6..
........4....2..	........4....2..
............6...	............6...
............2...	............2...
...........2....	...........2....
...........8....	...........8....
..............8.	..............8.
..............1.	..............1.
..............1.	..............1.
..............7.	..............7.
..4.........21.	..4.........21.
..3.........7e.	..3.........6e.
.e5.........23..	.e5.........22..
.38.........6c..	.38.........6c..
e5...8.....2....	e5...8.....2....
24...1.....6....	24...1.....6....
.....5....2....6	.....5....2....6
.....4....6....2	.....4....6....2
.........6....6.	.........6....6.
.........4....2.	.........4....2.
............6..	............6..
............2..	............2..
............2...	............2...
............8...	............8...
..............8	8
..............1	1
..............1	1
..............6	6
...4.........2.	...4.........2.

Table 8.4: Dominant characteristics in differential ii.

$p_{\mathsf{avg}} = 2^{-61}$	$p_{\mathsf{avg}} = 2^{-64}$	$p_{\mathsf{avg}} = 2^{-64}$
.1....7.....	.1....7.....	.1....7.....
.6....2.....	.e....2.....	.6....2.....
2....6.....	2...86.....	2....6.....
6....2.....	6...12.....	6....2.....
....6.....2	....7.....2	....6.....2
....2.....6	....2.....6	....2.....6
...6.....2.	...6.....2.	...6.....2.
...2.....6.	...2.....6.	...2.....6.
..6.....2..	..6.....2..	..6.....2..
..2.....6..	..2.....6..	..2.....6..
.6.....2...	.6.....2...	.6.....2...
.2.....6...	.2.....6...	.2.....6...
6.....2....	6.....2....	6.....2....
2.....6....	2.....6....	2.....6....
.....2....6	.....2....6	.....2....6
.....6....2	.....6....2	.....6....2
.....2....6.	.....2....6.	.....2....6.
.....c....2.	.....c....2.	.....c....2.
.....86..	.....86..	.....86..
.....12..	.....12..	.....92..
.....3...	.....3...	.....3..8
.....8...	.....8...	.....8..1
.....8	8	9
.....1	1	1
.....1	1	1
.....6	6	6
...4.....2.	...4.....2.	...4.....2.
...f.....d.	...f.....d.	...f.....d.
.861..84.....1.	.861..84.....1.	.861..84.....1.

Table 8.5: Number of quasidifferential trails for 14 rounds of Rectangle.

$\lvert c\rvert/p_{\mathsf{avg}}$	Differential i		Differential ii			Differential iii	
	2^{-63}	2^{-66}	2^{-61}	2^{-64}	2^{-64}	2^{-65}	2^{-65}
1	2	4	2	2	4	32	32
2^{-1}	2	4	2	2	4	32	32
2^{-2}	26	52	24	24	48	352	352
2^{-3}	26	60	24	24	56	480	480
2^{-4}	182	396	176	176	384	2656	2656

local, three-round effect. This is already an interesting outcome by itself, since previous techniques such as plateau characteristics are not able to describe such three-round effects.

Table 8.6: Differences and masks for two three-round quasidifferential trails with absolute correlation 2^{-13} and 2^{-19}. Both trails have the same masks.

Differences ($p_{\mathsf{trail}} = 2^{-63}$)	Differences ($p_{\mathsf{trail}} = 2^{-66}$)	Masks (both)
.....2....6.	.....2....6.	.....
.....c....2.	.....c....2.	.....c.....
.....86..	.....86..	.....84..
.....12..	.....92..	.....12..
.....3...	.....3..8	.....3...
.....8...	.....8..1	.....

Note that the propagation of the masks closely follows that of the differences. As discussed in Section 4.5.3, this is beneficial to obtain quasidifferential trails with high correlation. The correlation for the quasidifferential trail corresponding to the first characteristic in rounds 9 to 12 is equal to

$$(-1)^{\kappa_1} \times D^{\mathsf{S}}_{(\mathsf{c},\mathsf{c}),(0,2)} D^{\mathsf{S}}_{(0,2),(0,6)} \times D^{\mathsf{S}}_{(1,1),(8,8)} D^{\mathsf{S}}_{(2,2),(4,6)} \times D^{\mathsf{S}}_{(0,8),(3,3)}$$

$$= (-1)^{\kappa_1} \times \frac{-1}{8} \times \frac{1}{4} \times \frac{1}{8} \times \frac{1}{4} \times \frac{1}{8} = (-1)^{1+\kappa_1}\, 2^{-13}\,,$$

where $\kappa_1 = k_{10,10} + k_{10,15} + k_{11,12} + k_{11,13}$. Similarly, for the second characteristic, the correlation of the quasidifferential trail is equal to

$$(-1)^{\kappa_1} \times D^{\mathsf{S}}_{(\mathsf{c},\mathsf{c}),(0,2)} D^{\mathsf{S}}_{(0,2),(0,6)} \times D^{\mathsf{S}}_{(1,9),(8,8)} D^{\mathsf{S}}_{(2,2),(4,6)} \times D^{\mathsf{S}}_{(0,8),(3,3)} D^{\mathsf{S}}_{(0,1),(0,8)}$$

$$= (-1)^{\kappa_1} \times \frac{-1}{8} \times \frac{1}{4} \times \frac{-1}{8} \times \frac{1}{4} \times \frac{1}{8} \times \frac{1}{8} = \ (-1)^{\kappa_1}\, 2^{-19}\,.$$

Note the sign difference compared to the first characteristic. As shown below, it implies that the two characteristics are incompatible: for each key, one of them must have probability zero. Taking into account the first four quasidifferential trails, the probability of the first characteristic is

$$p_{i,1} \approx \big(1-(-1)^{\kappa_1}\big)\big(1+(-1)^{\lambda}/2\big)2^{-63} = \delta_1(\kappa_1)\big(1+(-1)^{\lambda}/2\big)2^{-62}\,,$$

where λ is a linear combination of round key bits. Although we did not include all quasidifferential trails in the analysis, Theorem 4.5 (2) allows concluding that the characteristic has probability zero when $\kappa_1 = 0$. Furthermore, it can be argued that lower-correlation trails are typically less significant. Although it is possible that for example the 26 trails with correlation 2^{-65} contribute a term of magnitude $2^{-63.3}$, this only happens for a small fraction of keys since it requires the signs of all these trails to point in the same direction. For the second characteristic, considering the first 8 trails results in

$$\begin{aligned} p_{i,2} &\approx \big(1+(-1)^{\kappa_1}-(-1)^{\kappa_2}-(-1)^{\kappa_1+\kappa_2}\big)\big(1+(-1)^{\lambda}/2\big)2^{-66} \\ &= \delta_0(\kappa_1)\delta_1(\kappa_2)\big(1+(-1)^{\lambda}/2\big)2^{-64}\,. \end{aligned}$$

Impact on the key-recovery attack. The time complexity of the 18-round key-recovery attack based on differential i is determined by the number of remaining pairs for the right key after filtering the data. For the maximum number of input structures, the number of remaining unordered pairs will be $p_i\,2^{63}$ on average.

If $\kappa_1 = 0$, then the number of pairs is $\delta_1(\kappa_2)(2+(-1)^{\lambda})/4$ on average over the other key bits. Since this is less than one for all values of κ_2 and λ, the key-recovery advantage will be too low to improve over brute-force.

For $\kappa_1 = 1$, the average number of unordered pairs is $2+(-1)^{\lambda}$. Using a threshold of one pair as in the original attack, this gives a time complexity of $2^{77.65}$ (80-bit key) or $2^{125.65}$ (128-bit key) assuming that the cost of evaluating the key-schedule is negligible compared to the cost of evaluating the cipher. Assuming that the number of right pairs follows a Poisson distribution within each key class, the success probability is then approximately $(1-e^{-1})/2+(1-e^{-3})/2 \approx 79\%$. Hence, the attack still marginally improves over exhaustive search. However, achieving this improvement requires filtering for weak keys using the condition $\kappa_1 = 1$ during the key-recovery phase. Otherwise, no improvement over exhaustive search is obtained. These observations can be summarized as follows.

Result 8.1. *The key-recovery attack on 18-round Rectangle from [292] using differential i does not improve over exhaustive search. For keys with* $k_{10,10} + k_{10,15} + k_{11,12} + k_{11,13} = 1$*, the attack can be modified to filter out candidate*

keys not satisfying this condition and then achieves a success probability of approximately 79% with a time complexity of $2^{77.65}$ *(80-bit key) or* $2^{125.65}$ *(128-bit key) 18-round encryptions. The attack requires* 2^{64} *data and enough memory to store* 2^{72} *counters.*

By Result 8.1, there is a rectified 18-round key-recovery attack on Rectangle with *average* success probability 39.5% and (marginally) better time complexity than exhaustive search.

Differential ii. The analysis of differential ii is similar to that of i. The three dominant characteristics are given in Tables 8.2 to 8.4. Based on the first four trails for the first two characteristics and the first eight trails for the third, the characteristic probabilities are

$$p_{ii,1} \approx \delta_1(\kappa_1)\big(1 + (-1)^{\lambda}/2\big)\, 2^{-60}$$

$$p_{ii,2} \approx \delta_1(\kappa_1)\big(1 + (-1)^{\lambda}/2\big)\, 2^{-63}$$

$$p_{ii,3} \approx \delta_0(\kappa_1)\delta_0(\kappa_2)\big(1 + (-1)^{\lambda}/2\big)\, 2^{-62}\,.$$

That is, for half of the keys, the dominant characteristic actually has no right pairs. For the other keys, its probability is roughly twice as large. The second characteristic shows similar behavior. Also note that the third characteristic is not compatible with the first two.

A similar analysis was performed for all other (up to rotational equivalence) 14-round differentials with a dominant characteristic of average probability 2^{-61}. The results were essentially the same.

Differential iii. Both characteristics with probability 2^{-65} are given in Table 8.3. Based on the 32 quasidifferential trails with correlation 2^{-65}, we find that the first characteristic has a nonzero probability if and only if 5 linearly independent equations in the round keys hold. The average probability over the keys satisfying these conditions is 2^{-60}. For the second characteristic, we find a similar effect with slightly different conditions on the round keys. Like for the first characteristic, the average probability over the weak keys is 2^{-60}. Furthermore, the conditions for the two characteristics to have nonzero probability are incompatible. Hence, the sum of the probabilities of the first two characteristics is 2^{-60} for 1/16 keys and zero for all other keys.

In addition, there are 13 characteristics with an average probability of 2^{-68}. Each of these characteristics has nonzero probability zero for only 1/64 or 1/128

keys. The conditions for this to happen may partially overlap or be inconsistent with the conditions for the first two characteristics.

8.4 Forgery and collision attacks on KNOT

In order to illustrate the relevance of quasidifferential trails to the analysis of permutations, this section analyzes several differential attacks on the KNOT family of permutations and their authenticated-encryption and hashing modes [293].

8.4.1 Specification of KNOT

KNOT is a large-state variant of Rectangle and was a second-round candidate in the NIST lightweight cryptography project. This chapter only considers the primary variant, which is a 256-bit permutation. The state is represented by a 4×64 rectangular array. The round function operations are similar to those of Rectangle, but a different S-box is used and the third and fourth row of the state are rotated by 8 and 25 positions respectively. The S-box of KNOT is given in Table 8.7.

Table 8.7: The S-box of KNOT.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S(x)	4	0	a	7	b	e	1	c	9	f	6	8	5	2	c	3

8.4.2 Differentials

At the 2020 NIST lightweight cryptography workshop, Zhang *et al.* [294] presented several differential attacks on round-reduced KNOT authenticated encryption and hashing modes. The differentials used in these attacks are listed in Table 8.8, along with their estimated probabilities (without taking into account quasidifferential trails). In this section, it will be shown that these attacks do not work because the probability of the differentials in Table 8.8 is zero. Furthermore, it will be shown that there exist round constants for which their probabilities are two orders of magnitude larger.

Table 8.8: Differentials for r rounds of KNOT-256. The column p_{avg} gives an estimate of the 'average' differential probability (for independent uniform random round constants). The differences are given in Table 8.9.

r	p_{avg}	Application	№
10	5×2^{-56}	Hash collision and AEAD forgery.	i
12	10×2^{-66}	Hash collision and AEAD forgery.	ii

Table 8.9: Input and output differences (a, b) for the differentials from Table 8.8.

i	a	0000000000000000000000000000000010000000000000001000000000000000
	b	0000000000000000000000000000000000000000100000000000000000000001
ii	a	0100000000000000001000000000000000000000000000000000000000000000
	b	0000000100000000000000000000000010000000000000000000000000000000

8.4.3 Analysis

The analysis of the differentials in Table 8.8 is similar to the analysis for Rectangle. The SMT-model for Rectangle can easily be modified to search for quasidifferential trails in KNOT.

Differential i. Based on the quasidifferential trails with correlation 2^{-56} for each of the five characteristics with $p_{\mathsf{avg}} = 2^{-56}$, one concludes that all of these characteristics have probability zero for the standard round constants of KNOT-256. Hence, the differential probability is much lower than what might be expected from the 'average'. Even if there exist other characteristics with unexpectedly large probability (a scenario considered below), this is a significant issue for the collision attack on the KNOT hash function. Indeed, the collision search consists of finding a right pair for one of the best few characteristics, since this is significantly easier than finding a right pair for the differential by random search.

Despite the observations above, it is possible that there exists a characteristic with a low 'average' probability but an unexpectedly high probability for the default round constants. The differential contains four characteristics with 'average' probability 2^{-60}. However, by analyzing the corresponding quasidifferential trails, we find that they too have probability zero. Next, there are 17 characteristics with 'average' probability 2^{-62}. Again, we find that all of them have probability zero. We also considered 24 characteristics with 'average' probabilities 2^{-63} and 2^{-65} and found that they have probability zero.

Although we did not analyze all characteristics with probability 2^{-66} or lower, they can only have a high nonzero probability for a very small fraction of round constants. Given the number of such characteristics, it is unlikely that a high probability characteristic exists.

On the flip side, there exist round constants for which one or more of the five characteristics have probability 2^{-50}. This is due to the existence of 64 quasidifferential trails with absolute correlation 2^{-56}. A careful inspection of the conditions on the round constants shows that there exist variants of KNOT with modified constants for which the probability of differential i is approximately $5 \cdot 2^{-50} = 2^{-47.7}$. Further improvements are possible by taking into account additional characteristics and quasidifferential trails.

Differential ii. The analysis of the 12-round differential is similar to the 10-round differential, and leads to similar conclusions. This is not surprising given that both characteristics follow a similar pattern up to rotational symmetry. Each of the 10 dominant characteristics has probability zero for the default round constants. In addition, we did not find any characteristics with 'average' probability 2^{-70} or higher with a nonzero probability. Hence, it is unlikely that the 12-round forgery and collision attacks presented by Zhang *et al.* are valid. Finally, there exist round constants for which one or more of the 10 characteristics have a probability of 2^{-59}.

8.5 Key-recovery attacks on Speck

This section investigates the key-dependency of several differentials for Speck from the literature. The bitvector constraints for modular addition from Theorem 8.3 are the main ingredient of the SMT-model. The same approach can be applied to any ARX block cipher or permutation.

Section 8.5.1 briefly reviews Speck. In Section 8.5.2, a simple explanation (using a single quasidifferential trail) for an experimental observation of Ankele and Kölbl [12] on Speck-64 is given. Sections 8.5.3 and 8.5.4 analyze the differentials used in the best published attacks on all variants of Speck.

8.5.1 Specification of Speck

Recall that Speck is a family of lightweight block ciphers designed at and endorsed by the United States NSA. The round function is shown in Figure 8.2. The block size is either 32, 64, 96 or 128 bits. For Speck-32, the rotation offsets

are given by $\alpha = 7$ and $\beta = 2$. For larger block sizes, $\alpha = 8$ and $\beta = 3$. The key-schedule follows a similar structure as the round function, with round keys replaced by round counters. Speck supports multiple key lengths m. These variants are denoted by Speck-n/m.

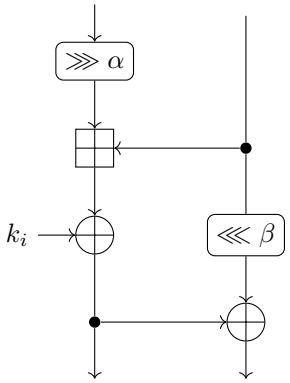


Figure 8.2: One round of Speck with round key k_i.

8.5.2 Explaining observations of Ankele and Kölbl on Speck-64

Ankele and Kölbl [12] experimentally estimated the probability of a 7-round differential for Speck-64 for 10000 random keys and found that the distribution of the number of right pairs is bimodal. Their results are reproduced in Figure 8.3, but colored to indicate two key classes that follow from the analysis below.

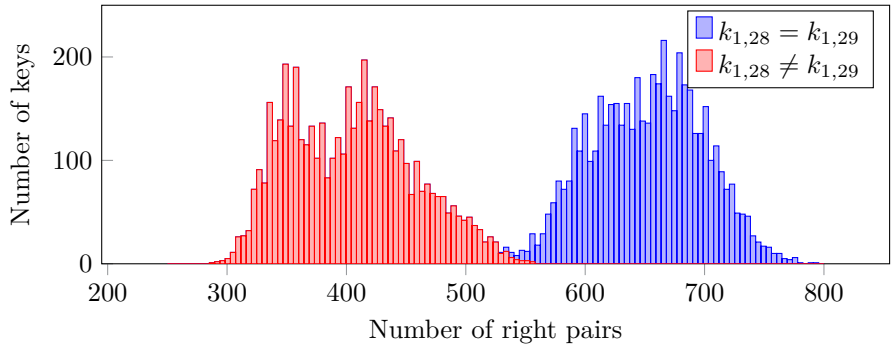


Figure 8.3: Number of right pairs for the Speck-64 differential from [12], for a total of 10000 keys. For each key, 2^{30} pairs were sampled uniformly at random.

The fact that the histogram in Figure 8.3 is bimodal already suggests the presence of an important quasidifferential trail with nonzero masks. Automatic search reveals that the best such quasidifferential trail has correlation 2^{-23}. The dominant characteristic (with probability 2^{-21}) and the masks of the quasidifferential trail with correlation 2^{-23} are shown in Table 8.10.

Differences		Masks	
4...4.92	1.42..4.	.....	.....
82.2....	..12.2..	18.....	.....
..9.....	....1...	.....	.....
....8...	.....	.....	.....
......8.	......8.	.....	.....
8.....8.	8....48.	.....	.....
..8..48.	..8.2.84	.....	.....
8.8.a.8.	8481a4a.	.....	.....

Table 8.10: Differential characteristic with key-averaged probability 2^{-21} for 7 rounds of Speck-64, and the masks of a corresponding quasidifferential trail with correlation 2^{-23}.

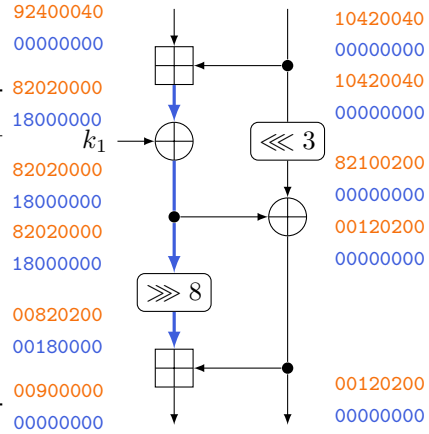


Figure 8.4: Two-round quasidifferential trail with correlation $2^{-5} \cdot 2^{-6} = 2^{-11}$, with differences in orange and masks in blue.

8.5.3 Analysis of differential attacks on Speck-32

The best published attacks on reduced-round Speck are differential attacks using the enumeration key-recovery strategy proposed by Dinur [119]. Given an r-round differential, an $r+3$ round attack is obtained by prepending one round (for free) and appending two rounds. For variants with longer key lengths, one performs the same attack for each guess of the last few round keys.

This section analyzes the best published attacks on Speck-32 reduced to 11-14 rounds. These attacks rely on the 6-9 round differentials shown in Table 8.11. Lee *et al.* [197] report on a 10-round differential with average probability $2^{-30.39}$, but it does not lead to a 15-round key-recovery attack because the time complexity would be $2^{31.39}$ for a success probability of $1 - 1/e \approx 63\%$.

Table 8.11: Differentials (a, b) for r-round Speck-32.

r	a	b	p_{avg}	Ref.	№
6	0211 0a04	850a 9520	2^{-13}	[2]	i
7	0a60 4205	850a 9520	2^{-18}	[2]	ii
8	1488 1008	850a 9520	$2^{-24} + 2^{-27}$	[2]	iii
9	8054 a900	0040 0542	$2^{-30} + 2 \cdot 2^{-33}$†	[65, 255]	iv

† $3060307 \cdot 2^{-47} \approx 2^{-29.45}$ with characteristics of average probability $\leq 2^{-49}$

Differentials i and ii. The six-round differential i is dominated by a characteristic with average probability 2^{-13}, given in Table 8.12. The next-best characteristic has average probability 2^{-23} and will be ignored in the analysis.

Table 8.12: Dominant characteristics for differentials i and ii.

$p_{\mathsf{avg}} = 2^{-13}$	$p_{\mathsf{avg}} = 2^{-18}$
	.a6. 42.5
.211 .a.4	.211 .a.4
28.. ..1.	28.. ..1.
..4.	..4. ....
8... 8...	8... 8...
81.. 81.2	81.. 81.2
8... 84.a	8... 84.a
85.a 952.	85.a 952.

There are two quasidifferential trails with correlation $\pm 2^{-15}$ and two with correlation $\pm 2^{-17}$. There also exist trails with absolute correlation 2^{-19} and lower, but their effect on the probability is limited except for a small fraction of keys. Grouping these trails appropriately, the following estimate is obtained:

$$p_i \approx (1 + (-1)^{0003^{\mathsf{T}} k_5}/4)(1 + (-1)^{0180^{\mathsf{T}} k_5}/4)2^{-13}\,,$$

where, for simplicity, only one trail of correlation $\pm 2^{-17}$ is included.

The analysis of the seven-round differential is similar. The dominant differential trail has average probability 2^{-18} and is the same as the six round trail with one additional round at the beginning. Hence,

$$p_{ii} \approx (1 + (-1)^{0003^{\mathsf{T}} k_6}/4)(1 + (-1)^{0180^{\mathsf{T}} k_6}/4)2^{-18}\,.$$

Differential iii. The differential is dominated by two characteristics, shown in Table 8.13. The first has average probability 2^{-24}. Since the last part of these

characteristics is the same as for the dominant characteristics of differentials i and ii, some of the same quasidifferential trails are obtained. However, there also exist quasidifferential trails with correlation equal to the probability of the trail. This implies that there exists keys for which these characteristics have probability zero. Specifically, for the first characteristic, we find that

$$p_{iii,1} \approx \delta_0(\mathtt{0600}^\mathsf{T} k_2)\delta_0(\mathtt{1800}^\mathsf{T} k_3)\,(1 + (-1)^{\mathtt{0003}^\mathsf{T} k_7}/4)(1 + (-1)^{\mathtt{0180}^\mathsf{T} k_7}/4)\,2^{-22}\,.$$

That is, its probability is zero for 3/4 keys, but four times larger for the other keys. For the second characteristic, we have

$$\begin{aligned} p_{iii,2} \approx{}& \delta_0(\mathtt{0600}^\mathsf{T} k_2)\delta_0(\mathtt{1800}^\mathsf{T} k_3)\delta_0(\mathtt{0a00}^\mathsf{T} k_2) \\ &\times (1 + (-1)^{\mathtt{0003}^\mathsf{T} k_7}/4)(1 + (-1)^{\mathtt{0180}^\mathsf{T} k_7}/4)\,2^{-24}\,. \end{aligned}$$

Hence, the second characteristic has nonzero probability only when the first probability is nonzero *and* $\mathtt{0a00}^\mathsf{T} k_2 = 0$.

Table 8.13: Two dominant characteristic for differential iii.

$p_{\mathsf{avg}} = 2^{-24}$	$p_{\mathsf{avg}} = 2^{-27}$
1488 1..8	1488 1..8
..21 4..1	..21 4..1
.6.1 .6.4	.e.1 .e.4
18.. ..1.	38.. ..1.
..4.	..4. ....
8... 8...	8... 8...
81.. 81.2	81.. 81.2
8... 84.a	8... 84.a
85.a 952.	85.a 952.

Differential iv. The probability is dominated by three characteristics (listed in Table 8.14). Additional characteristics only increase the overall probability, but more detailed analysis reveals that many additional characteristics have probability zero for most keys, and high probability for a relatively small fraction of keys.

The first characteristic has average probability 2^{-30}. Based on all quasidifferential trails with absolute correlation $\geq 2^{-32}$, one obtains

$$p_{iv,1} \approx \delta_0(\mathtt{000c}^\mathsf{T} k_5)\,(1 - (-1)^{\mathtt{0180}^\mathsf{T} k_1}/4)\,2^{-29}\,.$$

For the second characteristic (with average probability 2^{-33}), the quasidifferential trails with absolute correlation $\geq 2^{-34}$ yield

$$p_{iv,2} \approx \delta_1(\mathtt{6000}^\mathsf{T} k_2)\,(1 + (-1)^{\mathtt{000c}^\mathsf{T} k_5}/2 + (-1)^{\mathtt{0300}^\mathsf{T} k_4 + \mathtt{000c}^\mathsf{T} k_5}/2)\,2^{-32}\,.$$

Note that one of the two quasidifferential trails with absolute correlation 2^{-34} involves three modular additions. By Theorem 4.5, the condition $\mathtt{6000}^\mathsf{T} k_2 = 1$ is necessary to obtain a nonzero probability. However, the conditions $\mathtt{0300}^\mathsf{T} k_4 = 0$ and $\mathtt{000c}^\mathsf{T} k_5 = 1$ only imply a small but possibly nonzero correlation. For the third characteristic, we consider all quasidifferential trails with absolute correlation $\geq 2^{-35}$ and obtain

$$p_{iv,3} \approx \delta_1(\mathtt{0c00}^\mathsf{T} k_2)\delta_0(\mathtt{000c}^\mathsf{T} k_5)\,(1 - (-1)^{\mathtt{0180}^\mathsf{T} k_1}/2)\,2^{-31}\,.$$

Note that the condition $\mathtt{000c}^\mathsf{T} k_5 = 0$ is shared with the first characteristic. Since the probability of the second characteristic is too low, this implies that previous key-recovery attacks on 14 rounds of Speck-32 work for only half of the keys.

Table 8.14: Three dominant characteristics for differential iv.

$p_{\mathsf{avg}} = 2^{-30}$	$p_{\mathsf{avg}} = 2^{-33}$	$p_{\mathsf{avg}} = 2^{-33}$
8.54 a9..	8.54 a9..	8.54 a9..
.... a4.2	.... a4.2	.... a4.2
a4.2 34.8	e4.2 74.8	ac.2 3c.8
5.c. 8.e.	5.4. 8.61	7.c. 8.e.
.181 .2.3	.381 .2.7	.181 .2.3
...c .8..	..1c .8..	...c .8..
2...	2...	2... ....
..4. ..4.	..4. ..4.	..4. ..4.
8.4. 814.	8.4. 814.	8.4. 814.
..4. .542	..4. .542	..4. .542

Impact on key-recovery attacks. The above analysis allows us to reevaluate the best published attacks on reduced-round Speck-32. The attack on 13 rounds only works for one in four keys. Likewise, the attack on 14 rounds works only for half of the keys. Another way to formulate this is that the (key-averaged) success probability of these attacks is much lower than expected. For eleven and twelve rounds, the success probability is also slightly lower, but less so. Unfortunately, restoring the previous success-probability is not possible except by using alternative differentials.

However, if the results of the above analysis are taken into account, weak key attacks with lower data requirements are obtained. These attacks can be optimized either with respect to the number of weak keys, or with respect to the data complexity. To minimize the data complexity, we make assumptions on the key to maximize the probability of the differential. To maximize the number of keys for which the attack works, only conditions to ensure nonzero probabilities are imposed. Assuming that the adversary stops requesting data once the key has been found[3], these attacks require less data than what would be expected based on the average-case analysis.

Table 8.15: Rectified attacks on r-round Speck-32.

r	Time *encryptions*	Data *plaintexts*	Weak-keys *density*	Optimized
11	$2^{45.36}$	$2^{13.36}$	2^{-2}	Data
	$2^{45.88}$	$2^{13.88}$	1	Number of keys
12	$2^{50.36}$	$2^{18.36}$	2^{-2}	Data
	$2^{50.88}$	$2^{18.88}$	1	Number of keys
13	$2^{54.03}$	$2^{22.03}$	2^{-5}	Data
	$2^{56.20}$	$2^{24.20}$	2^{-2}	Number of keys
14	$2^{61.84}$	$2^{29.84}$	2^{-1}	Number of keys

The results are shown in Table 8.15. For example, the 6-round differential (11 round attack) has a probability at most $(1 + 1/4)^2\, 2^{-13} \approx 2^{-12.36}$. With early stopping, the average number of pairs required is $2^{13}(1/(1 - 1/4)^2 + 2/(1 - 1/4^2) + 1/(1 + 1/4)^2)/4 \approx 2^{12.88}$. For 14 rounds, we omit the attack optimizing the data complexity, since it requires more time than exhaustive search over a key space of size 2^{64-1} for a similar success probability.

8.5.4 Analysis of differential attacks on larger variants of Speck

The techniques from Section 8.5.3 to analyze Speck-32 carry over to the larger variants of Speck. This section reevaluates the best published attacks on these variants. They rely on the key-recovery technique of Dinur [119] and are based on the differentials shown in Table 8.16 below. For 16 rounds of Speck-96, Song *et al.* [255] also propose a differential with average probability $2^{-94.94}$. However, we do not include it as its probability is too low to improve over exhaustive search.

[3]This is possible due to the way the key-recovery attack works.

Table 8.16: Differentials for r-round Speck-n. Differences are given in Table 8.17. The average differential probability is p_{avg}, the average probability of the analyzed characteristics is p_{char}. The values p_{min} and p_{max} are the minimum and maximum value of the probability of the analyzed characteristics.

n	r	p_{avg}	p_{char}	p_{min}	p_{max}	Ref.	№
48	11	$2^{-44.31}$	$2^{-46} + 2^{-47}$	0	2^{-43}	[255]	i
64	15	$2^{-60.56}$	2^{-62}	0	2^{-59}	[255]	ii
96	15	$2^{-81.00}$	2^{-81}	0	$2^{-73.68}$	[255]	iii
128	20	$2^{-124.35}$	$4 \cdot 2^{-128}$	0	$2^{-120.36}$	[255]	iv

Table 8.17: Input and output differences (a, b) for the differentials in Table 8.16.

	a	b
i	504200 004240	202001 202000
ii	04092400 20040104	808080a0 a08481a4
iii	082020000000 000120200000	800400008124 842004008801
iv	0124000400000000 0801042004000000	8004000080000124 8420040080000801

Most of the differentials in Table 8.16 rely on a significant differential effect. Nevertheless, the analysis below will be limited to a few characteristics in each case. This is done only to simplify the analysis, since each characteristic has its own key-dependent behaviour that is not independent of other characteristics. Note that including additional characteristics can only increase the probability of the differential. In addition, it will be shown that key-dependence is much more significant than the differential effect for all differentials in Table 8.16. A detailed case-by-case analysis of the differentials in Table 8.16 now follows.

Differential i. For the 15-round Speck-48 differential, we consider two characteristics: the first has average probability 2^{-46}, the second 2^{-47}. These characteristics are shown in Table 8.18.

For the first characteristic, eight quasidifferential trails with correlation $\pm 2^{-46}$ are obtained. From these trails, it follows that the characteristic has nonzero probability if and only if $\mathtt{600000}^{\mathsf{T}} k_7 = 0$, $\mathtt{000c00}^{\mathsf{T}} k_7 = 0$ and $\mathtt{000003}^{\mathsf{T}} k_8 = 1$. In this case, the probability is 2^{-43}. There were no trails with correlation $\pm 2^{-47}$ for the same characteristic, and for simplicity we will neglect smaller trails.

The second characteristic has eight trails with correlation $\pm 2^{-47}$ and the same masks as for the first characteristic. However, the conditions for nonzero

Table 8.18: Dominant characteristics for differential i.

$p_{\mathsf{avg}} = 2^{-46}$		$p_{\mathsf{avg}} = 2^{-47}$	
5.42..	..424.	5.42..	..424.
..12.2	.2...2	..12.2	.2...2
....1.	1.....	....1.	1.....
......	8.....	......	8.....
8.....	8....4	8.....	8....4
8.8..4	8.8.2.	8.8..4	8.8.2.
84..a.	8..1a4	84..a.	8..1a4
6.8da4	6.8.8.	e.8da4	e.8.8.
.42..3	..24..	.42..7	..24..
.12.2.	....2.	.12.2.	....2.
2..1..	2.....	2..1..	2.....
2.2..1	2.2...	2.2..1	2.2...

probability are $\mathtt{600000}^{\mathsf{T}}k_7 = 1$, $\mathtt{000c00}^{\mathsf{T}}k_7 = 1$ and $\mathtt{000003}^{\mathsf{T}}k_8 = 0$. Hence, the characteristics are incompatible. If these conditions are met, then the probability is 2^{-44}.

It follows from the discussion above that for 1/8 keys, the probability is 2^{-43} up to the contributions of smaller quasidifferential trails. For 1/4 keys one characteristic has nonzero probability and the average reciprocal probability, which determines the data complexity of the attack, is $(2^{43} + 2^{44})/2 = 2^{43.58}$.

Differential ii. We consider a characteristic with average probability 2^{-62}, shown in Table 8.19. For this characteristic, there are 8 quasidifferential trails with correlation $\pm 2^{-62}$. Hence, the probability is zero for 7/8 keys and 2^{-59} otherwise.

Differential iii. The 15-round differential on Speck-96 is dominated by a single characteristic with probability 2^{-81} (see Table 8.20). However, the analysis reveals that this characteristic has nonzero probability only for 1/64 keys. Specifically, there exist 2^6 quasidifferential trails with absolute correlation 2^{-81}. This also implies that the probability of the characteristic is 2^{-75} for 1/64 weak keys.

In addition, we find $192 = 3 \cdot 2^6$ quasidifferential trails with absolute correlation 2^{-82}. The signs of the correlation of these trails are determined by independent key bits, such that for $1/2^9$ keys the probability of the characteristic becomes

Table 8.19: Characteristics for ii.

$p_{\mathsf{avg}} = 2^{-62}$	
.4.924..	2..4.1.4
2....82.	2.2....1
.......9	.1......
.8......	....
...8....	...8....
...8.8..	..48.8..
..48...8	.2.84..8
.6.8.8.8	164a.848
f24...4.	4.1.42..
..82.2..	....12.2
....9...	......1.
......8.	....
8.......	8.......
8.8.....	8.8....4
8...8..4	84..8.2.
8.8.8.a.	a.8481a4

Table 8.20: Characteristics for iii.

$p_{\mathsf{avg}} = 2^{-81}$	
.82.2.......	...12.2.....
...9........	.....1......
.....8......	....
.......8....	.......8....
.......8.8..	......48.8..
......48...8	.....2.84..8
.8..fe.8.8.8	.8..ee4a.848
...7724...4.	4.....1.42..
......82.2..	........12.2
........9...	..........1.
..........8.	....
8...........	8...........
8.8.........	8.8........4
8...8......4	84..8.....2.
8.8.8.8...2.	a.848.8..124
8..4....8124	842..4..88.1

$(1 + 3/2) \cdot 2^{-75} = 2^{-73.68}$. Based on this analysis, the average reciprocal probability is $64/27 \cdot 2^{75} \approx 2^{76.25}$ for the weak key class of density 2^{-6}.

Differential iv. The differential includes four characteristics with average probability 2^{-128}, amounting to a total average probability of 2^{-126}. These characteristics are listed in Table 8.22.

For one of these characteristics, we find 128 quasidifferential trails with absolute correlation 2^{-128}. Hence, the probability of the characteristic is actually 2^{-121} for one in 128 keys and zero otherwise.

Two characteristics each have 32 quasidifferential trails with absolute correlation 2^{-128}, implying that their probability is close to 2^{-123} for one in 64 keys and zero otherwise. The conditions to obtain a nonzero probability are a subset of those for the first characteristic. Furthermore, the conditions for both characteristics overlap in three linearly independent equations.

The remaining characteristic has eight quasidifferential trails with absolute correlation 2^{-128}. The conditions for obtaining a nonzero probability are a subset of the conditions required for each of the first three characteristics. Hence, if any of the previously discussed characteristics has a nonzero probability, then

the same is true for this characteristic.

From the above discussion, the probability is $2^{-121} + 2 \cdot 2^{-123} + 2^{-125} \approx 2^{-120.36}$ for one in 128 keys. In addition, for one in eight keys, the average reciprocal probability is $2^{125} \times 9/16 + 2^{122.68} \times 6/16 + 2^{120.36} \times 1/16 \approx 2^{124.36}$.

Impact on key-recovery attacks. The analysis above directly impacts the key-recovery attacks based on the differentials from Table 8.16. Like for Speck-32, all of these attacks have lower success probability than previously believed. Nevertheless, the analysis also leads to weak key attacks with lower data complexity. The results are summarized in Table 8.21.

For Speck-128, the analysis shows that the key-recovery attacks probably do not improve over exhaustive search over the reduced key-space. Improvements may be possible if checking the weak key conditions can be made comparatively cheap, provided that checking candidate keys dominates the cost. Since a detailed analysis of the time complexity is outside of the scope of this chapter, Table 8.21 only lists a distinguisher for this case. Although our analysis did not include all characteristics, these would only increase the *average* differential probability by $2^{-124.9}$. Further analysis shows that the probabilities of these characteristics are strongly key-dependent. Hence, the key-recovery attacks on Speck-128 from [255] most likely do not improve over exhaustive search.

Table 8.21: Rectified attacks on r-round Speck.

Variant	r	Time *encryptions*	Data *plaintexts*	Weak-keys *density*	Optimized
48/72	15	2^{68}	2^{44}	2^{-3}	Data
		$2^{68.58}$	$2^{44.58}$	2^{-2}	Number of keys
48/96	16	2^{92}	2^{44}	2^{-3}	Data
		$2^{92.58}$	$2^{44.58}$	2^{-2}	Number of keys
64/96	19	2^{92}	2^{60}	2^{-3}	——
64/128	20	2^{124}	2^{60}	2^{-3}	——
96/96	18	$2^{74.68}$	$2^{74.68}$	2^{-9}	Data
		$2^{77.25}$	$2^{77.25}$	2^{-6}	Number of keys
96/144	19	$2^{122.68}$	$2^{74.68}$	2^{-9}	Data
		$2^{125.25}$	$2^{77.25}$	2^{-6}	Number of keys
128/m	20	$2^{121.36}$	$2^{121.36}$	2^{-7}	Data†
		$2^{125.36}$	$2^{125.36}$	2^{-3}	Number of keys†

† Distinguisher only.

Table 8.22: Dominant characteristics for differential iv.

$p_{\mathsf{avg}} = 2^{-128}$	$p_{\mathsf{avg}} = 2^{-128}$
.124...4........ .8.1.42..4......	.124...4........ .8.1.42..4......
.8..2.2......... 48.8.12.2.......	.8..2.2......... 48.8.12.2.......
48...1.......... .84..8.1.......2	48...1.......... .84..8.1.......2
.8.8.8.........6 4a.848.8......16	.8.8.8.........6 4a.848.8......16
4...4.........32 1.42..4.......8.	4...4.........32 1.42..4.......8.
.2.2..........8. 8.12.2.......48.	.2.2..........8. 8.12.2.......48.
..1..........48. ..8.1.......2.84	..1..........48. ..8.1.......2.84
8.8.........2.8. 848.8......124a.	8.8.........6.8. 848.8......164a.
.4.........1244. 2..4.......8.144	.4.........324.. 2..4.......8.1.4
2..........8.22. 2.2.......48.8.1	2..........8..2. 2.2.......48.8.1
..........48...1 .1.......2.84..8	..........48...1 .1.......2.84..8
.........e.8.8.8 .8......1e4a.848	.........e.8.8.8 .8......1e4a.848
........f24...4. 4.........1.42..	........f24...4. 4.........1.42..
..........82.2.. ....12.2	..........82.2.. ....12.2
............9... ....1.	............9... ....1.
..............8.	..............8. ....
8............... 8...............	8............... 8...............
8.8............. 8.8............4	8.8............. 8.8............4
8...8..........4 84..8.........2.	8...8..........4 84..8.........2.
8.8.8.8.......2. a.848.8......124	8.8.8.8.......2. a.848.8......124
8..4....8....124 842..4..8....8.1	8..4....8....124 842..4..8....8.1

$p_{\mathsf{avg}} = 2^{-128}$	$p_{\mathsf{avg}} = 2^{-128}$
.124...4........ .8.1.42..4......	.124...4........ .8.1.42..4......
.8..2.2......... 48.8.12.2.......	.8..2.2......... 48.8.12.2.......
48...1.......... .84..8.1.......2	48...1.......... .84..8.1.......2
.8.8.8.........2 4a.848.8......12	.8.8.8.........2 4a.848.8......12
44..4.........12 1442..4.......8.	44..4.........12 1442..4.......8.
22.2..........8. 8.12.2.......48.	22.2..........8. 8.12.2.......48.
..1..........48. ..8.1.......2.84	..1..........48. ..8.1.......2.84
8.8.........6.8. 848.8......164a.	8.8.........2.8. 848.8......124a.
.4.........324.. 2..4.......8.1.4	.4.........1244. 2..4.......8.144
2..........8..2. 2.2.......48.8.1	2..........8.22. 2.2.......48.8.1
..........48...1 .1.......2.84..8	..........48...1 .1.......2.84..8
.........e.8.8.8 .8......1e4a.848	.........e.8.8.8 .8......1e4a.848
........f24...4. 4.........1.42..	........f24...4. 4.........1.42..
..........82.2.. ....12.2	..........82.2.. ....12.2
............9... ....1.	............9... ....1.
..............8.	..............8. ....
8............... 8...............	8............... 8...............
8.8............. 8.8............4	8.8............. 8.8............4
8...8..........4 84..8.........2.	8...8..........4 84..8.........2.
8.8.8.8.......2. a.848.8......124	8.8.8.8.......2. a.848.8......124
8..4....8....124 842..4..8....8.1	8..4....8....124 842..4..8....8.1

9

Generalized Feistel ciphers

This chapter presents truncated differential attacks on expanding and contracting Feistel ciphers. The attacks are generic, but lead to concrete results on GMiMC and SM4. The main focus is on the contracting case, of which the Chinese standard SM4 is the most important example. The implications for GMiMC will be discussed in Chapter 10.

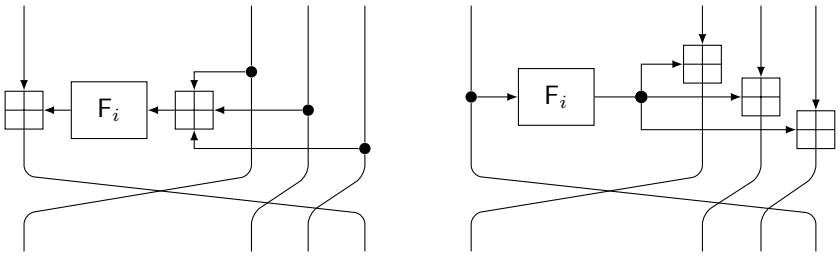
The contents of this chapter are based on the paper "Truncated differential attacks on contracting Feistel ciphers" [55] from ToSC 2022 (joint work with Yunwen Liu). The analysis of the expanding case is due to Gaëtan Leurent and appeared in our paper "Out of oddity: new cryptanalytic techniques against symmetric primitives optimized for integrity proof systems" [43] from Crypto 2020. For completeness, it is included in this chapter in slightly generalized form. My own results from [43] are presented in Chapter 10.

9.1 Introduction

Following its invention by Horst Feistel in the 1970s, the Feistel structure has become one of the most prominent architectures in modern block cipher design. One of its most eminent applications is the former American block cipher standard DES. Hence, it is not unexpected that the design and analysis of variants of the Feistel structure has become a significant research topic with valuable applications.

Following the widespread use of Feistel ciphers, many variations on the original structure were proposed. One of the main directions of this research has been the exploration of Feistel-like structures with more than two branches. Examples include *generalized Feistel ciphers* [225, 297] and the unbalanced Feistel ciphers discussed by Schneier and Kelsey [247]. The family of unbalanced Feistel structures can be further subdivided into expanding and contracting constructions. Figure 9.1 shows a single Feistel round of an *expanding Feistel cipher* and a *contracting Feistel cipher* with $t = 4$ branches.

The algebraic cipher GMiMC-erf [6] is an example of an expanding Feistel cipher. Examples of contracting Feistel ciphers include the algebraic cipher GMiMC-



(a) Contracting Feistel cipher. (b) Expanding Feistel cipher.

Figure 9.1: One round of a contracting or expanding Feistel cipher with $t = 4$ branches. The function F_i is potentially key-dependent.

crf [6] and the general-purpose block cipher SM4 [118]. The latter example is particularly important, as SM4 is the Chinese commercial block cipher standard (GB/T 32907-2016). In addition, it has been standardized by ISO/IEC under the reference number 18033-3:2010.

Given their widespread application, it is not surprising that the security analysis of Feistel ciphers has been an industrious area of research. Luby and Rackoff [208] proved the indistinguishability of three-round Feistel ciphers with uniform random round functions. Yun, Park and Lee [289] proved the birthday-bound security of t-branch[1] contracting Feistel ciphers with $2t - 1$ rounds. However, from a practical point of view, optimal security is expected and desired if the number of rounds is large enough. Hence, several works have proposed generic attacks – thereby lower bounding the number of rounds necessary for security. In particular, Guo *et al.* [155] describe meet-in-the-middle attacks on contracting Feistel ciphers. Patarin, Nachef and Berbain [230] analyze a more general contracting structure.

Differential cryptanalysis has proven to be one of the most successful tools in the security analysis of both concrete and generic Feistel structures. For example, the generic attacks of Patarin [229] on ordinary Feistel ciphers are based on differential cryptanalysis. The differential attack itself has also been extended and generalized in several ways. At FSE 1994, Knudsen [181] introduced an important extension known as *truncated differential cryptanalysis*.

In this chapter, the security of generic expanding and contracting Feistel ciphers is analyzed using truncated differentials. The motivation for doing so is twofold. On the one hand, from the viewpoint of block cipher design, it is important to know the baseline number of rounds required for security. On the other hand, new generic attacks can impact the security of concrete ciphers such as GMiMC

[1]The characteristic of the domain should not divide $t - 1$ to avoid a trivial distinguisher.

and SM4. SM4 in particular has received a significant amount of dedicated cryptanalysis and given its status as both a domestic and international standard, further advances in its analysis would be of interest.

The starting point for the analysis are two iterated truncated differentials. For the expanding case, the differential was first proposed for GMiMC-erf by Gaëtan Leurent as a part of our paper at Crypto 2020 [43]. This result is presented in Section 9.3 together with a similar truncated differential for the contracting case. In both cases, the resulting distinguisher covers $t^2 - t - 2$ rounds given $\mathcal{O}(N^{t-2})$ data for a Feistel cipher with t branches and a domain of size N^t. In the contracting case, considering truncated differential trails whose probability p_{trail} is lower than their ideal probability p_{ideal} improves this to $t^2 - 1$ rounds with $\mathcal{O}(N^{t-1})$ data.

In Section 9.4 and Section 9.4.1 in particular, improved truncated differential distinguishers are constructed. Only the contracting case is considered, since it is the most relevant for applications. The final t^2- and $(t^2 + t - 2)$-round distinguishers are based on a different iterated truncated differential that relies on several additional improvements. In particular, it takes advantage of relations between input and output differences, and optimizes the trade-off between the size of input structures and other parameters such as p_{trail} and p_{ideal}. The t^2-round distinguisher requires $\mathcal{O}(N^{t-2})$ data, for $t^2 + t - 2$ rounds $\mathcal{O}(N^{t-1})$ data is sufficient to achieve a constant advantage. The 16-round trail for $t = 4$ is shown to be optimal using SMT models in Section 9.4.2, and the distinguishers are verified experimentally in Section 9.4.3.

The t^2-round distinguisher is turned into a key-recovery attack in Section 9.5, resulting in a $(t^2 + 1)$-round attack requiring $\mathcal{O}(N^{t-2})$ data and $\mathcal{O}(N^{t-1})$ time. This is a significant improvement over the results of Guo *et al.* [155]. In particular, the key-recovery attacks of Guo *et al.* cover at most $5t - 4$ rounds (assuming the key length is equal to the block length).

As an immediate consequence of these results, an 18-round distinguisher and a 17-round key-recovery attack for SM4 are obtained. The data and time complexity of the 18-round distinguisher are approximately 2^{96}. The 17-round key-recovery attack uses 2^{70} chosen plaintexts and 2^{99} encryption operations. Although dedicated attacks on SM4 reach up to 23 rounds, their data- and time complexity is extremely large. As will be argued in Section 9.6, the new key-recovery attack is the best published attack for 17 rounds. This is remarkable given the fact that it does not use any details about the round function of SM4. The attacks also have implications for GMiMC, but these will be discussed in Chapter 10.

9.2 Preliminaries

Throughout this chapter, let U be a finite-dimensional vector space over a finite field. Furthermore, let $N = |U|$ denote the cardinality of the set U. That is, $N = q^n$ with q a prime power and n a positive integer.

9.2.1 Expanding and contracting Feistel ciphers

As illustrated in Figure 9.1 for $t = 4$, a Feistel round of a contracting Feistel cipher $\mathsf{R} : U^t \to U^t$ with t branches is defined by $\mathsf{R} : (x_1, x_2, \ldots, x_t) \mapsto (y_1, y_2, \ldots, y_t)$, with

$$y_i = \begin{cases} x_1 + \mathsf{F}(x_2 + x_3 + \ldots + x_t) & \text{if } i = t\,, \\ x_{i+1} & \text{else}\,. \end{cases}$$

Similarly, for an expanding Feistel cipher $\mathsf{R} : U^t \to U^t$, it holds that

$$y_i = \begin{cases} x_1 & \text{if } i = t\,, \\ x_{i+1} + \mathsf{F}(x_1) & \text{else}\,. \end{cases}$$

The function F is called the round function of the expanding or contracting Feistel cipher and is often key-dependent. The round function F can take various forms. For instance, the round function of SM4 has a SHARK-like structure consisting of an S-box layer followed by a multiplication with an MDS matrix [118]. For GMiMC-erf and GMiMC-crf [6], $\mathsf{F}(x) = (x + c)^3$, assuming U is a finite field and c is a constant or key. Since the attacks in this paper are generic and do not exploit the inner structure of the round function and key schedule, further details are omitted.

9.2.2 Truncated differentials

An important extension of differential cryptanalysis is the so-called truncated differential attack, first proposed by Knudsen [181]. Let A and B be subsets of U^t. The probability of the truncated differential (A, B) for $\mathsf{F} : U^t \to U^t$ with input set A and output set B is defined by

$$\Pr[A \xrightarrow{\mathsf{F}} B] = \Pr[\mathsf{F}(\boldsymbol{x}) - \mathsf{F}(\boldsymbol{y}) \in B \mid \boldsymbol{x} - \boldsymbol{y} \in A]\,,$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent uniform random variables on U^t. Equivalently,

$$\Pr[A \xrightarrow{\mathsf{F}} B] = \frac{1}{|A|} \sum_{a \in A} \Pr[\mathsf{F}(\boldsymbol{x} + a) - \mathsf{F}(\boldsymbol{x}) \in B]\,.$$

A truncated differential with $A = B$ is called iterative or iterated.

The approach to truncated differentials in this chapter will be classical. In particular, it will be assumed that probabilities in a truncated differential trail can be multiplied as if they correspond to independent events. However, since the analysis is generic, this will not lead to serious issues here. The results in Chapters 3 and 4 imply that there is an average-case[2] equivalence between multidimensional linear and truncated differential properties. Hence, the results of this chapter could also have been presented from the point of view of linear cryptanalysis. In retrospect, this might have clarified some of the heuristic assumptions that are made throughout the analysis.

When dealing with truncated differentials, it is sometimes convenient to use dependencies between input and output differences. A simple example is the property that any input difference from a set A results in *the same* output difference, rather than just any difference in the set A.

A convenient way to describe such properties without leaving the usual framework for truncated differentials from above, is to consider the input-extended cipher $\overline{\mathsf{F}} : U^t \to U^t \times U^t$ defined by $x \mapsto (x, \mathsf{F}(x))$. Indeed, if $A \subseteq U^t$ and $B \subseteq U^t \times U^t$, then

$$\Pr\left[A \xrightarrow{\overline{\mathsf{F}}} B\right] = \Pr[(\boldsymbol{x} - \boldsymbol{y}, \mathsf{F}(\boldsymbol{x}) - \mathsf{F}(\boldsymbol{y})) \in B \mid \boldsymbol{x} - \boldsymbol{y} \in A],$$

with $\boldsymbol{x}$ and $\boldsymbol{y}$ uniform random on U^t. The right-hand side above is indeed the desired probability. Ordinary truncated differentials correspond to the case $B = A \times C$ for some output difference set C.

9.3 Basic truncated differential distinguishers

In Section 9.3.1, iterated t-round truncated differentials for generic expanding and contracting Feistel ciphers are presented, and it is shown that they lead to interesting distinguishers.

When iterated too many times, the probability of the aforementioned truncated differential trails drops below the probability of the truncated differential for uniform random permutations. However, it is still possible to obtain a distinguisher as long as enough pairs are available. This observation is used in Section 9.3.2 to show that the distinguisher for contracting Feistel ciphers from Section 9.3.1 can be extended to more rounds.

[2] Average with respect to independent and uniform random round keys.

9.3.1 Iterated truncated differential trails

Figures 9.2a and 9.2b show iterated truncated differentials $A \to A$ for an expanding and a contracting Feistel cipher with $t = 4$ branches. For contracting Feistel ciphers, $A = \{(a, -a, 0, 0) \mid a \in U \setminus \{0\}\}$ and $A = \{(0, 0, -a, a) \mid a \in U \setminus \{0\}\}$ for the expanding case. The input difference is represented symbolically on each branch. For instance, the label a corresponds to an arbitrary nonzero input difference. The trail for the expanding case is due to Gaëtan Leurent, except that it is generalized to arbitrary expanding Feistel ciphers in Figure 9.2a.

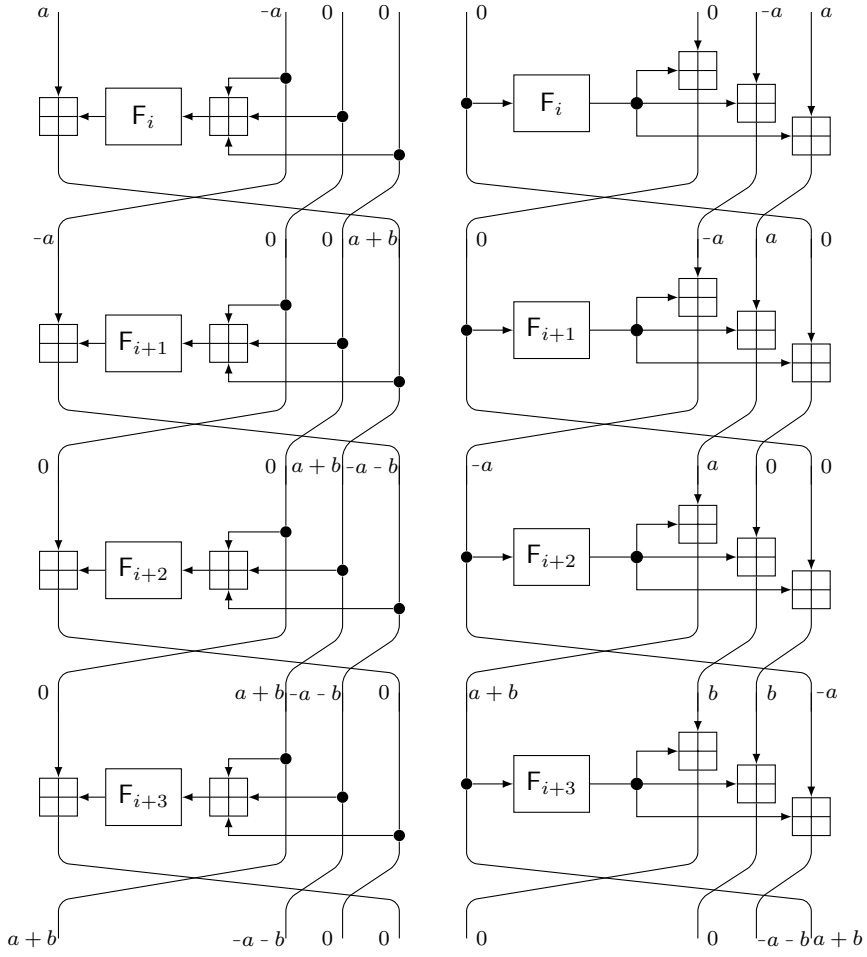
Consider the contracting case. In the first round, the probability is one since the output difference b of F_i is arbitrary. The probability for the second round is $1/(N-1) \sim 1/N$ on average, assuming that F_{i+1} is a uniform random permutation. Finally, the truncated differences in the third and fourth rounds propagate with probability one since $a + b - a - b = 0$. The analysis of the expanding case is similar. The probability in the first three rounds is one. In the last round, the probability is $1/(N-1) \sim 1/N$ on average.

Similar trails exists for any number of branches $t \geq 4$. In particular, one can simply set the rightmost $t-2$ branches to zero for the contracting case and the leftmost $t-2$ branches for the expanding case. Since the trails in Figures 9.2a and 9.2b have the same input and output sets, they can be iterated. For r divisible by t, an r round trail with probability $p_{\mathsf{trail}} = 1/(N-1)^{r/t} \sim 1/N^{r/t}$ is obtained.

For a random permutation, however, the probability of $A \to A$ is $p_{\mathsf{ideal}} = (N-1)/(N^t-1) \sim 1/N^{t-1}$. Hence, if $p_{\mathsf{ideal}} = o(p_{\mathsf{trail}})$, one obtains an r-round distinguisher using approximately $1/p_{\mathsf{trail}} = N^{r/t}$ data. It follows that a t-branch expanding or contracting Feistel cipher must have $r > t^2 - 2t$ rounds to be secure. Furthermore, the attack on $t^2 - 2t$ rounds requires N^{t-2} data.

In fact, the above can be improved by prepending $t-2$ rounds to the trail in the contracting case as shown in Figure 9.3 for $t = 4$, and similarly by appending $t-2$ rounds to the trail in the expanding case. Since this modification does not affect p_{real} or p_{ideal} in either case, one obtains a distinguisher on $t^2 - t - 2$ rounds with N^{t-2} data.

A further extension by appending at most $t-2$ rounds to the trail is possible in the contracting case. However, appending s rounds increases p_{ideal} to $(N-1)/(N^{t-s}-1) \sim 1/N^{t-s-1}$. Hence, appending rounds does not lead to an attack on more rounds. Nevertheless, for a smaller number of rounds, appending $t-2$ rounds may lead to a lower data complexity. Optimizing for the number of rounds, we obtain the following result.



(a) Contracting case. (b) Expanding case.

Figure 9.2: Truncated differential for expanding and contracting generalized Feistel ciphers with $t = 4$ branches. The probability of both trails is $1/N$. In characteristic two, the minus signs may be dropped.

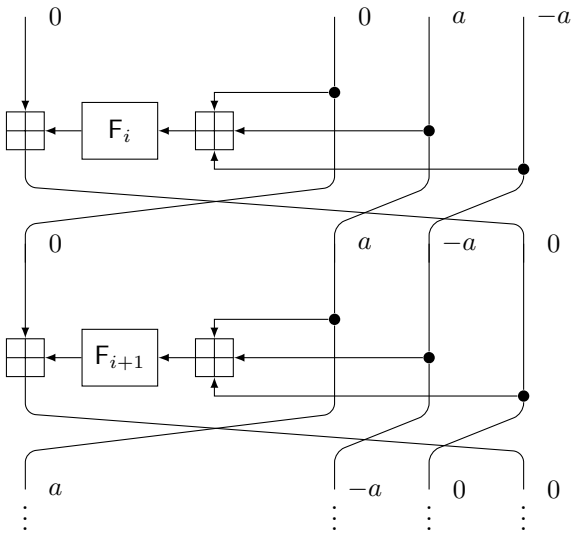


Figure 9.3: Prepending $t-2$ rounds to the trail from Figure 9.2a with probability one. In characteristic two, the minus signs may be dropped.

Result 9.1. *A generic expanding or contracting Feistel cipher with t branches and t^2-t-2 rounds can be distinguished from a uniform random permutation with advantage $\Theta(1)$ using N^{t-2} data.*

Result 9.1 implies that the number of rounds of an expanding or contracting Feistel cipher must scale quadratically with the number of branches. For a large enough number of branches, this is a significant improvement over the attacks by Patarin *et al.* [230] and Guo *et al.* [155], who showed that the number of rounds must scale linearly with the number of branches. However, note that for the most interesting applications small values of t are of particular importance. Hence, a more detailed comparison is necessary.

The distinguishers of Patarin *et al.* [230] consider a more general form of contracting Feistel ciphers, but cover at most $2t-1$ rounds. Hence, Result 9.1 improves over this for all $t \geq 4$. Guo *et al.* [155] describe key-recovery attacks up to $5t-4$ rounds when the key length is $t \log_2 N$ bits. By guessing the last round key, the distinguisher above is easily adapted to a key-recovery attack on t^2-t-1 rounds with time complexity $\tilde{\mathcal{O}}(N^{t-1})$. Hence, Result 9.1 improves over the attacks of Guo *et al.* [155] for $t \geq 6$.

Nevertheless, Result 9.1 leaves significant room for improvements. Importantly, even extensions by a number of rounds linear in t are relevant, since important

examples such as SM4 have a small number of branches. A first improvement is the use of input structures. An affine space of dimension d over U contains $N^d(N^d-1)/2$ pairs. This allows reducing the amount of data. For example, for the truncated differential used in Result 9.1, one could reduce the data complexity to $2N^{t-3}$ in this manner. However, the number of rounds that can be distinguished does not increase as this is determined by the condition $p_{\mathsf{ideal}} = o(p_{\mathsf{trail}})$.

In the remainder of this chapter, several improvements to the basic truncated differential from Section 9.3.1 will be introduced. This includes the extension of the distinguisher to the setting with $p_{\mathsf{trail}} \leq p_{\mathsf{ideal}}$ in Section 9.3.2. In Section 9.4 further improvements will be made, including taking more advantage of input structures and using dependencies between input and output differences. Such improvements can be useful for both expanding and contracting Feistel ciphers. However, due to the lack of applications beyond GMiMC-erf, only contracting Feistel ciphers are considered in the remainder of this chapter.

9.3.2 Extended distinguisher with $p_{\mathsf{trail}} \leq p_{\mathsf{ideal}}$

Even when the probability of a truncated differential trail is much lower than the ideal probability of the corresponding truncated differential, it is sometimes possible to obtain a distinguisher. Heuristically, the idea is that wrong pairs for a truncated differential trail behave as if they were encrypted under a uniform random permutation. Hence, one can argue that the true probability p_{real} of the truncated differential satisfies the folklore approximation

$$p_{\mathsf{real}} \approx p_{\mathsf{trail}} + p_{\mathsf{ideal}}(1 - p_{\mathsf{trail}}) = p_{\mathsf{ideal}} + p_{\mathsf{trail}}(1 - p_{\mathsf{ideal}}) \approx p_{\mathsf{ideal}} + p_{\mathsf{trail}}\,. \quad (9.1)$$

That is, one expects slightly more right pairs for the cipher than for a random permutation.

We now consider the data complexity of a distinguisher with $p_{\mathsf{trail}} \ll p_{\mathsf{ideal}}$, and derive a distinguisher for more than $t^2 - t - 2$ rounds based on exactly the same iterated truncated differential as in Section 9.3.1. This is possible because, as was just argued, this truncated differential has

$$p_{\mathsf{real}} - p_{\mathsf{ideal}} \approx p_{\mathsf{trail}} \sim 1/N^{r/t}\,. \quad (9.2)$$

Suppose one encrypts D plaintext pairs with differences in the input set of the truncated differential. After encrypting these pairs under the cipher, we expect to obtain an average number of Dp_{real} pairs with a difference in the output set of the truncated differential. For a random permutation, the expected number of pairs is instead Dp_{ideal}. Moreover, the distribution of the number of right pairs

under a random permutation is binomial with variance $p_{\mathsf{ideal}}(1-p_{\mathsf{ideal}})D \sim p_{\mathsf{ideal}}D$ since $p_{\mathsf{ideal}} \ll 1$. By Theorem 1.1, to obtain a distinguisher with advantage $\Theta(1)$, we require that the difference between the means of the real and ideal distribution of the number of valid pairs exceeds the standard deviation of the ideal distribution:

$$D\,(p_{\mathsf{real}} - p_{\mathsf{ideal}}) \gg \sqrt{D\,p_{\mathsf{ideal}}}\,.$$

Rewriting the above, one obtains the estimate

$$D \gg p_{\mathsf{ideal}}/(p_{\mathsf{real}} - p_{\mathsf{ideal}})^2\,. \tag{9.3}$$

For a more detailed derivation of this result including a proof that this is optimal, see for instance Blondeau and Gérard [68].

By (9.2) and (9.3), we get $D \gg p_{\mathsf{ideal}}\,N^{2r/t}$ with $p_{\mathsf{ideal}} \sim 1/N^{t-1}$. Hence, if the trail is iterated $t-1$ times (once more than in Section 9.3.1), we must have $D = N^{t-1}$ pairs. After prepending $t-2$ rounds with probability one, a distinguisher on $t(t-1)+t-2 = t^2-2$ rounds is obtained. In fact, it is possible to improve upon this by appending one round at the end. This increases the ideal probability to approximately $1/N^{t-2}$, so that t^2-1 rounds can be distinguished using N^t pairs. Using an input structure of size N, these pairs can be obtained from roughly $2N^{t-1}$ plaintexts. Note that iterating the truncated differential t times or appending one more round at the end of the trail is not worthwhile, since that would lead to a data complexity of N^t.

Result 9.2. *A generic contracting Feistel cipher with t branches and t^2-1 rounds can be distinguished from a uniform random permutation with advantage $\Theta(1)$ using N^{t-1} data.*

Compared to Result 9.1, the distinguisher with $p_{\mathsf{trail}} \ll p_{\mathsf{ideal}}$ covers $t+1$ more rounds. Unlike in Section 9.3.1, the limiting factor in further improvements is now the number of pairs that can be obtained from the input space. Indeed, provided that one has a sufficiently large input structure, it would be possible to use more than N^t pairs. However, the trail from Section 9.3.1 has an input structure of size only N. In Section 9.4.1, truncated differentials that allow for bigger input structures will be introduced.

Finally, note that for $t = 4$ (as for SM4), we now obtain a 15 round distinguisher with a data complexity of N^3. This may be compared with the 16 round key-recovery attack of Guo *et al.* [155] with a similar data complexity. The distinguisher from Result 9.1 can also be extended to a 16 round key-recovery attack, but it requires N^4 partial decryption operations and hence offers only marginal advantage over exhaustive search.

9.4 Improved truncated differential distinguishers

This section develops improved truncated differential attacks on generic contracting Feistel ciphers. In Section 9.4.1, improvements to the distinguisher from Section 9.3 are obtained by taking into account input structures and by allowing for dependencies between the input and output differences. As a result, distinguishers for $t-1$ additional rounds with the same-data complexity are obtained (Result 9.3). In Section 9.4.2, an SMT model is developed to show that (for $t=4$), these distinguishers are indeed optimal. Section 9.4.3 reports on the experimental verification of these results.

9.4.1 Input structures and input-output dependencies

As discussed at the end of Section 9.3.2, the number of rounds that can be distinguished using the truncated differential from Section 9.3 is primarily limited by the dimension of the input space. Indeed, if the dimension d of the input structure is large enough, then the number of pairs used in the attack can exceed N^t while keeping the data and time complexity below N^t. In particular, one can obtain up to $N^d(N^d-1)/2 \sim N^{2d}/2$ pairs for each structure of size N^d. A larger dimension d leads to a distinguisher for more rounds, *ceteris paribus*. In principle d can be up to $t-1$, but the trade-off with the probability p_{trail} of the trail as well as the ideal probability p_{ideal} should be kept in mind. Note that when using structures, the time complexity of the distinguisher is still equal to the data complexity. Indeed, one can count the number of occurrences of the relevant parts of the output and store them in a table. After sorting, the number of valid pairs can be determined by iterating through the table once.

Iterative truncated differential with larger d. In Figure 9.4, an iterative truncated differential for $t=4$ is shown. Whereas the truncated differential from Section 9.3 had input structures of dimension one, the truncated differential in Figure 9.4 has $d=2$. Importantly, this is achieved by allowing dependencies between the input and output differences. Recall from Section 9.2, page 239, that this can be described formally by considering the input-extended cipher. The probability of the four-round trail in Figure 9.4 is $p_{\mathsf{trail}} \sim 1/N$, and the ideal probability is $p_{\mathsf{ideal}} \sim 1/N^3$.

The trail from Figure 9.4 can be generalized to t branches by considering the following input difference structure:

$$(a_1, a_2, \ldots, a_{t-2}, b, b) \text{ such that } \sum_{i=1}^{t-2} a_i = -b\,,$$

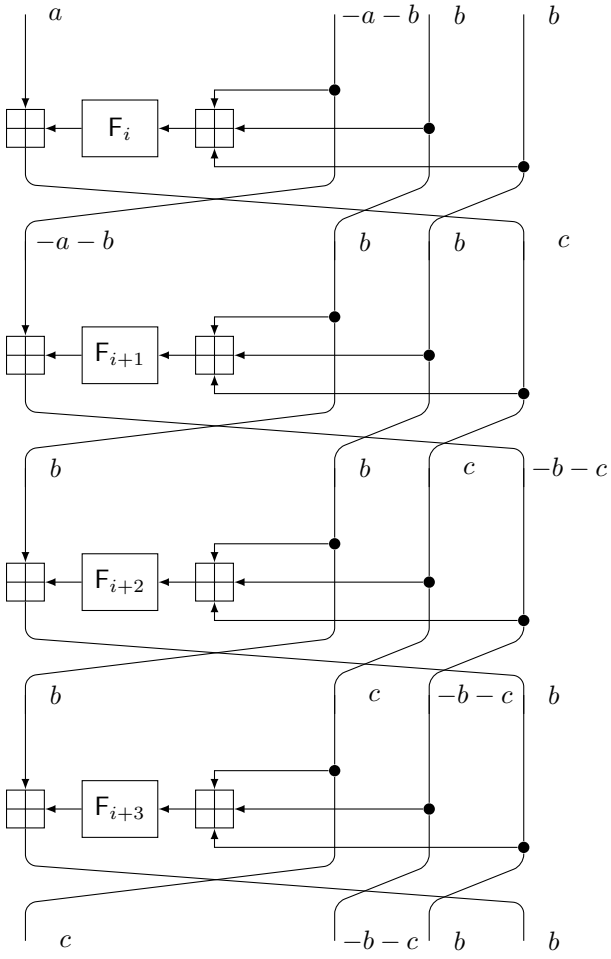


Figure 9.4: Truncated differential for a contracting generalized Feistel cipher with $t = 4$ branches. The probability of this trail is $1/N$. In characteristic two, the minus signs may be dropped.

with $a_1, \ldots, a_{t-2}, b$ in U not all zero. Like the trail from Section 9.3, this iterated trail covers r rounds with $p_{\mathsf{trail}} \sim 1/N^{r/t}$ for r a multiple of t. Furthermore, the input structure has dimension $d = t - 2$ and $p_{\mathsf{ideal}} \sim 1/N^3$.

There are several ways to extend the iterative trail from above by additional rounds, such as prepending two rounds or appending $t - 2$ rounds. However, extending the number of rounds is not necessarily optimal as it may lead to a smaller d or a higher ideal probability p_{ideal}. The next paragraph analyzes the available trade-offs in detail.

Trade-off analysis. Suppose we iterate the trail from Figure 9.4 m times, covering $m\,t$ rounds. Further assume that when the trail is deterministically extended by s rounds, the input structure dimension is d and let i be an integer such that $p_{\mathsf{ideal}} \sim 1/N^i$. As discussed in Section 9.3.2 on page 244, the number of pairs D required for the attack is then

$$D \sim p_{\mathsf{ideal}}/(p_{\mathsf{real}} - p_{\mathsf{ideal}})^2 = N^{2m-i}\,,$$

since $p_{\mathsf{real}} - p_{\mathsf{ideal}} \sim 1/N^m$. Since the maximum number of pairs that can be obtained is $N^{t-d}\,N^d(N^d - 1)/2 \sim N^{t+d}/2$, we must have $D \ll N^{t+d}$. Hence, $2m - i \leq t + d$ or equivalently $m \leq \lfloor (t + d + i)/2 \rfloor$. It follows that the number of rounds r that can be distinguished satisfies

$$r \leq t \left\lfloor \frac{t + d + i}{2} \right\rfloor + s\,. \tag{9.4}$$

This bound is tight. If $2m \geq 2d + i$, then the corresponding data complexity is $N^d\,N^{2m-i-2d} = N^{2m-d-i}$. Otherwise, the data complexity is approximately $N^{m-i/2}$.

We now consider the possible trade-offs for the iterative trail introduced above. Note that it is always possible to prepend two rounds to the trail, without affecting the trail probability or p_{ideal}. If no further rounds are appended, then $i = 3$ as discussed above. It then follows from (9.4) with $s = 2$ and $d = t - 2$ that $r = t\lfloor t + 1/2 \rfloor + 2 = t^2 + 2$ rounds can be distinguished using N^{t-1} data. If instead an additional $t - 2$ rounds are appended, then $i = 2$ and $s = t$. Hence, (9.4) yields a distinguisher on $r = t^2 + t$ rounds with N^t data. This data complexity is only marginally acceptable. If we choose $m = t - 1$ instead of $m = t$, then a distinguisher for t^2 rounds with N^{t-2} data is obtained. It is also possible to append $t - 1$ rounds, but this yields $i = 1$ and is not worthwhile.

Alternatively, it is possible to use a slightly larger input structure. Indeed, consider input differences of the following form:

$$(a_1, a_2, \ldots, a_{t-2}, b, b)\,,$$

with $a_1, \ldots, a_{t-2}, b$ in U not all zero. This is an input structure of dimension $d = t - 1$. Importantly, this can be connected to the iterative trail from above with probability $\sim 1/N$. With the input structure above, it is not possible to prepend rounds without decreasing d. If no rounds are appended, then $i = 3$ and $s = 0$. Hence, by (9.4), there is a distinguisher on $t\lfloor t + 1 \rfloor = t^2 + t$ rounds with N^t data. Again, the data complexity of this distinguisher is only marginally acceptable. Choosing $m = t$ instead, a t^2-round distinguisher with lower data complexity is obtained. However, since $2t < 2(t-1) + 3$, the data complexity is $N^{t-1.5}$ – higher than for the t^2-round distinguisher from above. Finally, suppose we append $t - 2$ rounds such that $i = 2$ and $s = t - 2$. By (9.4), one can then distinguish up to $t\lfloor t + 1/2 \rfloor + t - 2 = t^2 + t - 2$ rounds with N^{t-1} data.

Overview of the best distinguishers. Summarizing the results from the trade-off analysis yields Result 9.3. Note that these distinguishers cover more rounds than those mentioned in Results 9.1 and 9.2. More importantly, they improve over previous generic attacks on contracting Feistel ciphers for any number of branches.

Result 9.3. *For a generic contracting Feistel cipher with t branches, we have the following distinguishers from a uniform random permutation:*

- $t^2 + t - 2$ *rounds using* N^{t-1} *data,*
- t^2 *rounds using* N^{t-2} *data.*

Each of these distinguishers achieves an advantage of $\Theta(1)$.

The case $t = 4$ is of particular relevance, since the corresponding results yield a distinguisher on 16 rounds of SM4 with 2^{64} data and time and on 18 rounds with 2^{96} data and time. In Section 9.5, it will be discussed how the distinguishers in Result 9.3 can be turned into key-recovery attacks on slightly more rounds. It will be demonstrated in Section 9.6 that this leads to the best-known key-recovery attack on 17-round SM4.

9.4.2 Modelling truncated differentials using SMT

In this section, the propagation of truncated differentials through a generic contracting Feistel cipher is modelled as an SMT problem. For simplicity, the model is restricted to the case with base field $\mathbb{F}_2$. An important feature of the model is that it can be used to find distinguishers with $p_{\mathsf{trail}} \ll p_{\mathsf{ideal}}$. In addition,

relations between the input and output variables are accounted for. This is important to verify the distinguishers from Section 9.4.1. The implementation is based on Boolector [222] and is available online.[3] To automate the process of finding truncated differentials by SMT solving, we need to model the truncated differences and the corresponding transition rules by properly defined variables and constraints.

Variables. For each nonzero truncated difference in the model, it is either a new variable or a linear combination of previous variables. In order to simplify checking linear (in)dependence, a bitvector variable is used to represent the truncated difference on each branch. The zero bitvector represents the zero difference. However, nonzero bitvectors do not correspond to a specific difference and should be thought of as symbolic variables.

Specifically, a bitvector with Hamming weight one represents a free variable, *i.e.* one that is not a linear combination of other variables. Linearly independent truncated differences are represented by distinct bitvectors. Truncated differences that are linear combinations of other differences (with coefficients zero or one, as we work over $\mathbb{F}_2$) can then be represented by a bitvector with Hamming weight two or higher.

The length of the bitvectors is determined by the maximum number of free variables. Specifically, the truncated differences for an r-round t-branch contracting Feistel structure contain at most $r + t$ independent variables, including the input differences and the output differences of the round functions F_i with $i = 1, \ldots, r$. Hence, bitvectors of length $r + t$ are sufficient.

Finally, the model keeps track of the probabilities p_{trail} and p_{ideal} and represents them by their integer weights $\mathsf{wt}(p_{\mathsf{trail}})$ and $\mathsf{wt}(p_{\mathsf{ideal}})$ such that $p_{\mathsf{trail}} \sim 1/N^{\mathsf{wt}(p_{\mathsf{trail}})}$ and $p_{\mathsf{ideal}} \sim 1/N^{\mathsf{wt}(p_{\mathsf{ideal}})}$. In addition, the probability p_i of the truncated differential in round i of the trail has weight $\mathsf{wt}(p_i)$. If a probability is zero, we formally denote its weight by ∞. Within the SMT model, infinite weights are excluded by appropriate constraints.

Constraints. The average trail probability satisfies $p_{\mathsf{trail}} = \prod_{i=1}^{r} p_i$. Equivalently, the weights must satisfy the constraint

$$\mathsf{wt}(p_{\mathsf{trail}}) = \sum_{i=1}^{r} \mathsf{wt}(p_i)\,.$$

Based on the above, additional constraints for $\mathsf{wt}(p_{\mathsf{trail}}) \neq \infty$ are relatively easy to deduce. To ensure that $p_{\mathsf{trail}} \neq 0$, the first $t - 1$ branches of each

[3] `http://tim.cryptanalysis.info/contracting-feistels.zip`

output difference must equal the last $t-1$ branches of the output difference. Furthermore, since the round function is a permutation, the output difference of the round function is zero if and only if the input difference, *i.e.* the exclusive or of the bitvectors representing the rightmost $t-1$ input branches, is zero. The weight $\mathsf{wt}(p_{\mathsf{trail}})$ is then equal to the number of round function output differences that are zero or have Hamming weight at least two (a linear combination of other variables).

If $p_{\mathsf{trail}} \leq p_{\mathsf{ideal}}$, additional constraints are necessary to avoid trivially invalid trails. In particular, at least one branch of the differences in each round must be a linear combination of the differences in preceding branch differences in that round or the input-branch differences. Linear dependence is modelled recursively.

Finally, suitable constraints for $\mathsf{wt}(p_{\mathsf{ideal}})$ are added by recursively determining the number of output variables that are independent of the input variables and previous output variables.

Proving optimality using SMT. Using the SMT model introduced above, one can verify the correctness of the differential distinguishers from Section 9.4.1. To this end, we place a constraint on the trail weight for fixed values of the input structure dimension and the ideal weight and iteratively increase its value until the problem is found to be satisfiable. Alternatively, it is possible to optimize the overall weight directly, by modelling the data complexity formula from Section 9.4.1 within the SMT problem.

For $t = 4$ and $r = 16$, the best possible truncated differential distinguishers (in terms of data complexity) for all possible values of the input structure size d in $\{1, 2, 3\}$ and ideal weight i in $\{1, 2, 3\}$ are obtained within 100 minutes on a standard personal computer. The distinguisher from Result 9.3 was one of several solutions with data complexity N^2. No distinguishers with a lower data complexity were found.

9.4.3 Experimental verification

In this section the generic distinguishers from Result 9.3 are verified experimentally for $t = 4$ and $N = 2^8$. Let $\lambda = p_{\mathsf{ideal}} D$ when the distinguisher (implicitly) generates D pairs. Let $\boldsymbol{x}$ be a random variable counting the number of right pairs when the distinguisher is evaluated on a random permutation. If the distinguisher uses a threshold value $\tau\sqrt{\lambda}$, then the false-positive rate is

$$P_{\mathsf{F}} = \Pr\left[\boldsymbol{x} \geq \lambda + \tau\sqrt{\lambda}\right] = \Pr\left[\boldsymbol{x} \geq (1 + \tau/\sqrt{\lambda})\,\lambda\right].$$

Since P_{F} is the sum of D independent Bernoulli random variables with probability of success p_{ideal}, it follows from the multiplicative Chernoff bound that for $\tau \leq \sqrt{\lambda}$,

$$P_{\mathsf{F}} \leq e^{-\tau^2/3}. \tag{9.5}$$

Choosing $\tau = 2$, the false-positive rate satisfies $P_{\mathsf{F}} \leq 0.26$. For $\tau = 3/2$, one has $P_{\mathsf{F}} \leq 0.47$.

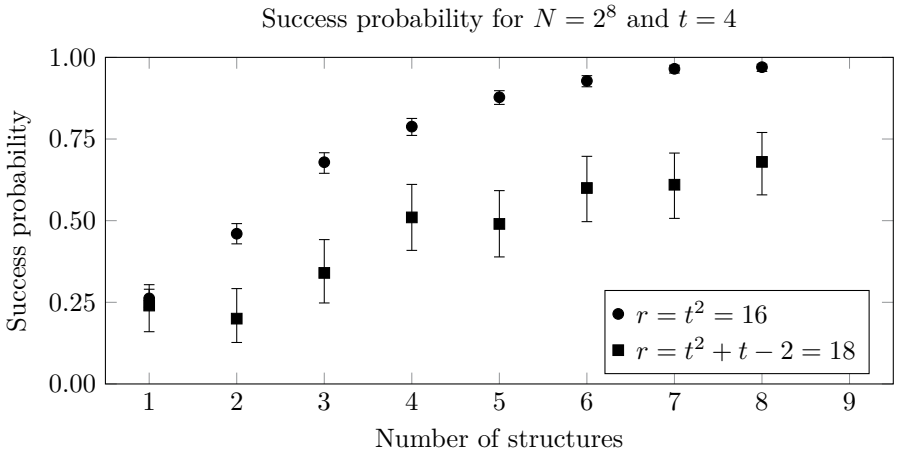


Figure 9.5: Estimates of the success probability of the distinguishers from Result 9.3 with $t = 4$ and $\tau = 2$ ($r = 16$) or $\tau = 3/2$ ($r = 18$) as a function of the data complexity (number of structures). The error bars correspond to 95% confidence intervals computed using the Clopper-Pearson method.

Figure 9.5 shows the results of the experiments for $t = 4$ and $N = 2^8$. Source code to reproduce this figure is available online.[4] The estimated success probabilities are shown for $\tau = 2$ (for $r = 16$) or $\tau = 3/2$ (for $r = 18$), *i.e.* a false-positive rate which is at most 26% or 47%. For $r = 16$, each datapoint is based on 1000 evaluations of the attack on a contracting Feistel cipher with uniform random round functions. For $r = 18$, each estimate is based on 100 experiments.

As expected, the success probability gradually increases when more structures are used. The experiments show that achieving a high success probability requires slightly more than N^2 (for $r = 16$) or N^3 (for $r = 18$) data. Note that the success probabilities shown in Figure 9.5 do not represent the maximal advantage that can be achieved using these distinguishers, since the trade-off between P_{F} and the success probability was not optimized for these experiments.

[4] `http://tim.cryptanalysis.info/contracting-feistels.zip`

9.5 Key-recovery attacks

If the rounds functions $\mathsf{F}_1, \ldots, \mathsf{F}_r$ of a contracting Feistel cipher are keyed permutations rather than random permutations, then it is of interest to consider key-recovery attacks in addition to distinguishers. For simplicity, assume that the last round key can take N possible values and the total key length is equal to the block size. This is the case for both SM4 and several instances of GMiMC-crf. The time complexity of any key-recovery attack can then be at most around N^t encryption operations.

The distinguisher on $t^2 + t - 2$ rounds from Result 9.3 could in theory be extended to a key-recovery attack on $t^2 + t - 1$ rounds with data complexity slightly larger (to ensure P_{F} is low enough) than N^{t-1} by guessing the last round key. However, the time complexity of this attack would be slightly above N^t *partial* encryptions, which is only a marginal improvement over brute-force in the most optimistic case.

More realistically, the t^2-round distinguisher from Result 9.3 can be extended to a key-recovery attack on $t^2 + 1$ rounds with data complexity close to N^{t-2} and time complexity close to N^{t-1}. Again, the attack is based on guessing the last round key and partially decrypting the set of N^{t-2} ciphertexts. Suppose that we wish to reduce the number of candidates for the last round key by a fraction $1/N^{1-\delta}$. By (9.5) in Section 9.4.3, this can be achieved by choosing the distinguisher's threshold τ such that $\exp(-\tau^2/3) \leq 1/N^{1-\delta}$. Equivalently,

$$\tau \geq \sqrt{3(1-\delta)\,\log N}\,,$$

where log denotes the natural logarithm.

By a similar reasoning as in the derivation of (9.3), the number of required pairs D must satisfy $D(p_{\mathsf{real}} - p_{\mathsf{ideal}}) \geq \tau\sqrt{D p_{\mathsf{ideal}}}$. Since $p_{\mathsf{ideal}} \sim 1/N^2$ and $p_{\mathsf{real}} - p_{\mathsf{ideal}} \sim 1/N^{t-1}$, it follows that

$$D \geq \tau^2 p_{\mathsf{ideal}}/(p_{\mathsf{real}} - p_{\mathsf{ideal}})^2 \approx \tau^2 N^{2t-4}\,.$$

Since the input structures have dimension $t-2$, the data complexity becomes $\tau^2 N^{t-2} = 3(1-\delta)\,(\log N)\,N^{t-2}$. The overall time complexity T of the attack is then

$$T \approx N^{t-1+\delta} + 3\epsilon(1-\delta)(\log N)\,N^{t-1}\,,$$

assuming partial decryption takes ϵ times the time of encryption. The first term is the remaining guessing cost and the second term is due to the partial decryption of the data. To minimize the time complexity, the parameter δ in $[0, 1)$ should be chosen to balance the terms. For instance, if $N = 2^{32}$ and $t = 4$

(the case of SM4), then with $\delta = 0.06140$ one has

$$T \approx 2^{97.96} + 2^{97.96} = 2^{98.96}\,.$$

This estimate assumes $\epsilon = 1/16$. The corresponding data complexity is $2^{69.96}$.

Alternatively, one could guess more than one round key and rely on a distinguisher for a smaller number of rounds with a lower data complexity. However, when optimizing for the number of rounds covered by the attack, this is typically not worthwhile because guessing one round key increases the time complexity by an equal amount as increasing the length of the truncated differential by t rounds. Nevertheless, this approach could be interesting in the low-data setting. Optimal trails for a smaller number of rounds can be obtained using the SMT model from Section 9.4.2, but a detailed analysis of such attacks is left as future work.

9.6 Application to SM4

From Result 9.3, truncated differential distinguishers on 16- and 18-round SM4 can be obtained using 2^{64} and 2^{96} data respectively. As discussed in Section 9.5, the generic 16-round distinguisher can be converted into a 17-round key recovery attack with $2^{69.96}$ data and $2^{98.96}$ time by guessing the last round key. The attacks on SM4 from this chapter are summarized and compared to the main attacks from the literature in Table 9.1.

In terms of the number of rounds covered, the best attacks are differential and linear type and cover up to 24-round SM4. However, those attacks require a large amount of data and time. For instance, the 24-round linear attack requires 2^{127} data and 2^{127} time (as measured in arithmetic operations), which is close to the full codebook and the cost of a brute-force key search. Previous attacks on SM4 aiming at lower data and time complexity were presented by Guo *et al.* [155], who give a 16-round key-recovery attack with data and time complexity of 2^{99} using a generic meet-in-the-middle approach. The 16-round truncated differential distinguisher proposed in this chapter only requires 2^{64} data, which significantly improves over their attack. The 17-round key-recovery attack from this chapter has a similar time complexity, but a much lower data complexity.

There appears to be no direct analysis of differential or linear attacks on 16- or 17-round SM4. To make a reasonable comparison, we consider previous differential and linear attacks with a reduced number of rounds. This leads to the conclusion that the 17-round key-recovery attack from Section 9.5 improves over reduced-round variants of previous work, for the same or similar data

Table 9.1: An overview of attacks on the SM4 block cipher. Attacks marked by † are distinguishers, the others are key-recovery attacks.

Attack Type	Rounds	Data	Time	Ref.
Differential	12	2^{67}	2^{67}	[257]†
	21	2^{118}	2^{127}	[290]
	22	2^{117}	2^{112}	[295]
	23	2^{118}	2^{127}	[257]
Multiple differential	21	2^{104}	2^{114}	[254]
	23	2^{114}	2^{127}	[296]
Linear	22	2^{117}	2^{112}	[134]
	23	2^{120}	2^{122}	[204]
	24	2^{127}	2^{127}	[204]
Multiple linear	22	2^{112}	2^{124}	[206]
	23	2^{127}	2^{127}	[91]
Multidimensional linear	23	2^{123}	2^{123}	[203]
Boomerang	18	2^{120}	2^{117}	[179]
Rectangle	16	2^{125}	2^{116}	[291]
	18	2^{124}	2^{113}	[179]
	18	2^{127}	2^{104}	[188]
Impossible differential	16	2^{105}	2^{107}	[207]
	16	2^{117}	2^{132}	[270]
	17	2^{117}	2^{132}	[282]
	18	2^{117}	2^{132}	[252]
Meet-in-the-middle	16	2^{99}	2^{99}	[155]
Truncated differential	16	2^{64}	2^{64}	§9.4.1†
	17	2^{70}	2^{99}	§9.5
	18	2^{96}	2^{96}	§9.4.1†

complexity. This claim is motivated by the analysis below. For brevity, it will be assumed that the reader is familiar with previous attacks on SM4.

The differential attack from Zhao *et al.* [296] is similar to that of Su *et al.* [257], so the latter will be used for reference below. Both attacks are based on multiple 19-round differentials with the same output difference. The key-recovery appends four rounds. If the 19-round differentials are restricted to 12 rounds, they have probabilities between 2^{-84} and 2^{-82}. As each structure of 2^{33} plaintexts contains 2^{46} pairs, the resulting data complexity would be around 2^{70}. However,

following [257, §5.1], appending five rounds for the key-recovery attack would have a time complexity larger than 2^{99}, in particular because there are few conditions that can be used to filter pairs in the last round.

More generally, we cannot use any known 13-round differentials because their probability is too low. There exist other 12-round characteristics with higher probability (optimally 2^{-67}, according to [257]), but the key-recovery heavily depends on the structure of the output differences so the analysis from [257,296] is then not directly applicable. In any case, a five-round extension by key-recovery with a time complexity below 2^{99} is unlikely.

The other attacks in Table 9.1 covering more than 18 rounds are linear attacks. Liu *et al.* [204] propose to use a three-round iterative approximation with absolute correlation 2^{-3r} for r rounds. For 19 rounds this gives an absolute correlation of 2^{-57}, and key-recovery extends this by four rounds. To set up a round-reduced variant of this attack with $\leq 2^{70}$ data, the approximation can be extended to at most 11 rounds (absolute correlation 2^{-33}). However, the key-recovery should then cover 6 rounds, which is not realistic since 80 bits already have to be guessed for just four rounds.

The work by Liu *et al.* [203] is a multidimensional linear attack, but it only uses 25 linear approximations (extended to 64 in order to apply a multidimensional analysis) and their absolute correlations are lower than those from [204]. The key-recovery appends four rounds and extending this would drive up the time complexity even more.

Cho and Nyberg [91] rely on the 5-round iterative approximations from [134]. These have absolute correlation $2^{-18.4}$ in the last two rounds. Hence, for 13 rounds, the absolute correlation would be $2^{-36.8}$. This gives a data complexity of around $2^{73.6}$. Using multiple approximations as in [91], a rough estimate suggests a data complexity similar to that of our 17-round key-recovery attack. However, this improvement will only be achieved if some internal round key bits are guessed (signs of the correlations must be guessed). Due to this, the key-recovery strategy of [91] only covers three rounds. In particular, they guess 88 key bits from the initial and final rounds as well as 34 internal round key bits. Hence, only a 16 round key-recovery attack is obtained and with a time complexity above 2^{99}.

10

Arithmetization-oriented primitives

The first part of this chapter is based on the paper "Out of oddity: new cryptanalytic techniques against symmetric primitives optimized for integrity proof systems" [43] from Crypto 2020. This paper is joint work with Anne Canteaut, Itai Dinur, Maria Eichlseder, Gregor Leander, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, Yu Sasaki, Yusoke Todo and Friedrich Wiemer. It is the result of the efforts of the StarkWare hash function evaluation committee[1] to analyze the security of the arithmetization-oriented primitives GMiMC and HadesMiMC. This paper covers a wide range of topics, but only some of these – mainly my own contributions – are discussed in this chapter. Some new observations on GMiMC are included as well. I am grateful to Nathan Keller for pointing out an error in an earlier version of the cost analysis of the preimage attack in Section 10.3.4.

The second part of this chapter presents attacks on the Legendre PRF and its variants, which led to the solution of several challenges organized by the Ethereum foundation[2]. These attacks were published in ToSC 2020 under the title "Cryptanalysis of the Legendre PRF and generalizations" [36] and are joint work with Ward Beullens, Aleksei Udovenko and Giuseppe Vitto. All authors contributed equally.

10.1 Introduction

The emergence of cryptographic protocols with advanced functionalities, such as fully homomorphic encryption, multi-party computation and new types of proof systems, has led to a demand for new symmetric primitives offering good performance in these specific applications. However, the standard criteria which govern the design of symmetric primitives are usually not appropriate in this context. For example, the cost of the homomorphic evaluation of a symmetric primitive is mainly determined by its multiplicative size and depth [8]. Similarly, the area of integrity proof systems, such as SNARKs, STARKs and Bulletproofs, is asking for symmetric primitives optimized for yet another cost metric.

[1] `https://starkware.co/hash-challenge/`
[2] `https://legendreprf.org/`

Therefore, several new ciphers and hash functions have been proposed for use in these advanced protocols. They include FHE-friendly encryption schemes such as LowMC [8], Flip [216], Kreyvium [82] and Rasta [122], MPC-friendly primitives such as MiMC [7], GMiMC [6] and the Legendre PRF [153], and some primitives dedicated to proof systems such as the functions from the Marvellous family, including Jarvis, Friday [14], Vision and Rescue [9]. In general, the security of most of these primitives is still poorly understood. For example, LowMC was broken a few weeks after its publication [120, 124, 238] and a practical attack against Jarvis was discovered not long after it was published [5].

In this chapter, the security of a few of the above-mentioned primitives is analyzed. Sections 10.2 and 10.3 are concerned with GMiMC-erf, GMiMC-crf, and HadesMiMC. Although these primitives are unusual in the sense that they are naturally defined over a large field, they are nevertheless iterative constructions. Section 10.4, on the contrary, presents attacks on the Legendre PRF. The latter is not constructed as a composition of 'simple' functions. Hence, it is not surprising that the methods used in Section 10.4 are considerably different from those used elsewhere in this thesis.

10.1.1 GMiMC and HadesMiMC

The first part of this chapter analyzes the security of GMiMC [6] and HadesMiMC [152]. As mentioned in Chapter 9, GMiMC is a family of generalized Feistel ciphers with both an expanding (GMiMC-erf) and a contracting variant (GMiMC-crf). HadesMiMC is inspired by SHARK [243], but introduces a partial S-box layer in the middle rounds. Both GMiMC and HadesMiMC can be used as block ciphers, or as cryptographic permutations in a sponge-based hash function. The analysis in this chapter has a slightly different flavor from that in the version published at Crypto 2021 [43], since the focus is less on specific instances of these primitives and more on their general security. Nevertheless, the implications for the concrete instances that were specified in the context of a public competition launched by the company StarkWare[3] will be discussed whenever they are relevant.

The results from Chapter 9 yield truncated differential attacks on GMiMC-erf and GMiMC-crf. In fact, even the unoptimized attacks from Section 9.3.1 result in full-round distinguishers and key-recovery attacks for some instances. However, the practical relevance of these attacks may be limited because most applications of GMiMC use a relatively small number of branches but a large field size. In such cases, algebraic attacks become the dominant threat vector. These results are described in detail in Section 10.2.2. Although the truncated

[3] `https://starkware.co/hash-challenge/`

differential attacks are not directly applicable in the hash function setting, it is possible to combine them with algebraic techniques to obtain reduced-round collision attacks for the concrete parameters proposed by StarkWare. This application will not be discussed in this thesis; details can be found in the Crypto paper [43].

Section 10.3 presents integral distinguishers and preimage attacks on HadesMiMC. They exploit a weakness resulting from the use of partial S-box layers, especially when these are combined with a linear layer that satisfies certain conditions. This observation is described in Section 10.3.2. It is shown that some of the MDS matrices proposed by the authors of HadesMiMC [151] meet the conditions. Section 10.3.3 sets up improved integral distinguishers on reduced-round HadesMiMC. For most concrete parameters sets, the distinguishers cover all but the first four rounds. In addition, zero-sum partitions for all but the first two rounds are presented. With some assumptions on the linear layer, the integral property extends over an arbitrary number of partial rounds. In this case, preimage attacks on several full-round instances are also obtained.

10.1.2 Legendre PRF

The Legendre symbol is the multiplicative character of $\mathbb{F}_p^\times$ that maps quadratic residues to one and non-residues to minus one. At Crypto 1988, Damgård [108] proposed a pseudorandom generator based on the Legendre symbol. In 2016, Grassi *et al.* [153] proposed a modification of Damgård's construction as a candidate pseudorandom function and showed that it is efficient in the multiparty computation setting. Their proposal is called the Legendre PRF.

Damgård additionally considered several generalizations of his pseudorandom generator that could be more efficient and/or more secure. One of these proposals is to use Jacobi symbols modulo a composite number n. Calculating Jacobi symbols is generally easier because computing them reduces to computing Legendre symbols modulo each of the smaller prime factors of n. Furthermore, Damgård argues that Jacobi symbols lead to a more secure pseudorandom generator. A second generalization proposed by Damgård is the use of higher power residue symbols. This potentially increases the throughput of the PRF, as higher residue symbols yield more than one bit of output per evaluation.

The Legendre PRF was proposed to be used in the Ethereum 2.0 proof-of-custody mechanism [141]. In this context, several cryptanalysis bounties were announced by the Ethereum foundation during the Crypto 2019 rump session [140]. The challenges include concrete instances of the Legendre PRF with expected security levels ranging from 44 to 128 bits of security. For each instance, 2^{20} sequential output bits are given and the goal is to recover the secret key.

Despite the longevity of Damgård's pseudorandomness conjecture, relatively few cryptanalytic results are available. Given quantum query access to the Legendre PRF, the key k can be recovered with a single query and in quantum polynomial time [271]. However, no subexponential attacks are known in the setting where the adversary can only query the PRF classically. The best cryptanalytic results in the classical setting are due to Khovratovich [178], who gives a memoryless birthday-bound attack. His attack recovers the key with a computational cost of $\mathcal{O}(\sqrt{p}\log p)$ Legendre symbol evaluations when given $\sqrt{p}\log p$ queries to L_k. Khovratovich also considers a higher-degree variant of the Legendre PRF. Similar to the Jacobi symbol generalization, the higher-degree Legendre PRF potentially offers security and efficiency benefits.

Section 10.4 advances the state-of-the-art in the cryptanalysis of the Legendre PRF by improving upon Khovratovich's attacks on the one hand, and by providing the first security analysis of the Jacobi and power residue symbol generalizations on the other hand. Table 10.1 provides a summary of the main results. The main improvement stems from the fact that, unlike earlier work, the new attacks exploit the multiplicativity of the Legendre symbol. The practical relevance of the attacks is demonstrated by solving the first two Legendre PRF challenges proposed by the Ethereum foundation [141]. These were expected to correspond to a security level of 44 and 54 bits, but the new attacks imply that the actual security levels for these challenges are significantly lower.

Table 10.1: Data, time and memory requirements of attacks on the Legendre PRF. The time and memory values are asymptotic ($\mathcal{O}$-notation) and assume a machine with word size $\Theta(\log p)$, ℓ and s denote the time complexity of computing a Legendre and power residue symbol respectively.

	Ref.	Data	Time	Memory
Legendre PRF	[178]	$\log p$	$\ell p \log p$	$\log p$
	[178]	$\sqrt{p}\log p$	$\ell\sqrt{p}\log p$	$\log p$
	§10.4.3	M	$M + \ell p \log p/M$	$M \log p$
	§10.4.4	M	$M^2 + \ell p \log^2 p/M^2$	M^2
	§10.4.4	M	$M^2 + p \log^2 p/M^2$	$M^2/\log p$
degree $d \geq 2$ Legendre PRF	[178]	$\log p$	$\ell p^d\, d \log p$	$d \log p$
	[178]	p	$\ell p^{d-1} d \log p$	$d \log p$
	§10.4.5	M	$M^2 + \ell p^d d^2 \log^2 p/M^2$	M^2
	§10.4.5	$d \log p$	$p^{\lceil d/2 \rceil} d \log p$	$p^{\lfloor d/2 \rfloor} d \log p$
r^{th} power-residue PRF	§10.4.8	M	$M^2 + sp \log^2 p/(M^2 \log^2 r)$	$M^2 \log r$
	§10.4.8	M	$M + sp \log^2 p/(Mr \log^2 r)$	$M \log r$

Section 10.4.4 shows how Khovratovich's attack can be improved in the low-data setting. In particular, for $M \leq \sqrt[4]{p}$ queries, the key can be recovered with a time complexity of $\mathcal{O}(p \log^2 p/M^2)$ Legendre symbol evaluations using $\mathcal{O}(M^2)$ memory. This attack is generalized to the higher-degree case in Section 10.4.5. Furthermore, a large class of weak keys for the higher-degree Legendre PRF is exhibited. For keys in this class, key-recovery requires roughly $\mathcal{O}(p^{\lceil d/2 \rceil} d \log p)$ operations with only $d\lceil \log p \rceil$ queries to the PRF. This attack requires $\mathcal{O}(p^{\lfloor d/2 \rfloor} d \log p)$ bits of memory, but trade-offs are available using Van Oorschot-Wiener golden collision search. A reduction to the unique k-XOR problem is also given, resulting in further time-memory trade-offs.

The first of Damgård's generalizations is discussed in Section 10.4.7. Specifically, it is shown that the Jacobi PRF can be broken with cost proportional to the cost of breaking the Legendre PRF for each of the prime factors of the modulus separately. The power residue symbol generalization is analyzed in Section 10.4.8. Besides a straightforward generalization of the attack from Section 10.4.4 to the r^{th} power residue symbol PRF, a more efficient attack for the case with large r is obtained.

Concurrent work. Days after the results in Section 10.4 appeared on ePrint, Kaluđerović *et al.* [174] solved the next Legendre PRF challenge. Their attack is similar, but with an improved complexity of $\mathcal{O}(M^2/\log p + p \log p \log\log p/M^2)$ operations on a machine with word size $\Theta(\log p)$.

10.2 Cryptanalysis of GMiMC

After introducing GMiMC in Section 10.2.1, Section 10.2.2 revisits the truncated differential attacks from Chapter 9 in the special case of GMiMC. This leads to full-round attacks on several block cipher instances. For the hash function case, only reduced-round distinguishers on the underlying permutation are obtained. Although these can be converted into reduced-round collision attacks, this result will not be discussed here. Section 10.2.3 contains a brief overview of the other attacks on GMiMC from the paper [43] that are not included in this section.

10.2.1 Specification of GMiMC

GMiMC is a family of block ciphers designed by Albrecht *et al.* in 2019 [6], based on different types of generalized Feistel networks with round function $x \mapsto (x + k_i + c_i)^3$ over a finite field. The round constants $c_1, \ldots, c_r$ are chosen

at random and the round keys $k_1, \ldots, k_r$ can be fixed to zero to obtain a cryptographic permutation. For the block cipher case, several possible key-schedules are discussed below.

Only the expanding and contracting variants, GMiMC-erf and GMiMC-crf respectively, are analyzed in this section. As in Chapter 9, the branches are numbered from 1 to t starting from the leftmost branch in Figure 9.1. The designers' security claims assume that the primitive is instantiated over a field of prime order p. They mention that "even if GMiMC can be instantiated over $\mathbb{F}_{2^n}$, [they] do not provide the number of rounds to guarantee security in this scenario".

For use as a block cipher, two key-schedules with different security claims are supported. The first type sets $k_1 = k_2 = \ldots = k_r$ and aims for $\log_2 p$-bit security. However, this choice is flawed because it leads to a slide attack as pointed out by Bonnetain [74]. The second type is linear and derives the round keys from a master key in $\mathbb{F}_p^t$. This construction aims at $t \log_2 p$-bit security. Since the first key-schedule is flawed, only the second option is analyzed in this section. However, the attacks below are also applicable to the first type and would be the best-known attacks if the key-schedule is modified to thwart the attack from [74].

If t is large compared to $\log_3 p$, then the authors of GMiMC argue that the best attack is a truncated differential key-recovery attack[4] and deduce the following choice for the number of rounds of GMiMC-erf and GMiMC-crf:

$$\left\lceil t(t+1)\, \frac{\log_2 p}{2(\log_2 p - 1)} \right\rceil + t + 1\,.$$

However, if t is small compared to $\log_3 p$, then interpolation attacks become dominant and the number of rounds is chosen as $2\lceil \log_3 p \rceil + 5t - 4$ for GMiMC-crf and $2\lceil \log_3 p \rceil + 3t - 2$ for GMiMC-erf.

For the hash function case, the number of rounds can be determined either as above or by attempting to match the generic security of the sponge construction. The latter approach was used for the concrete parameter sets proposed by StarkWare.

Finally, it is worth noting that $t - 1$ should not be divisible by p. This requirement is not mentioned in the specification of GMiMC [6]. Nevertheless, it is particularly important in the block cipher setting to avoid trivial invariants. In the case of GMiMC-erf, the sum of all branches is preserved under an arbitrary number of rounds if p divides $t - 1$. In the case of GMiMC-crf, every coset of

[4] Needless to say, not the same truncated differential attack as in Section 10.2.2 below.

the vector space of states with all branches equal is mapped to another coset of that same vector space.

10.2.2 Truncated differential attacks on GMiMC

Result 9.1 from Chapter 9 yields full-round attacks on some block cipher instances of GMiMC. Indeed, as mentioned above, GMiMC-erf and GMiMC-crf have roughly $t(t+3)/2+1$ rounds when t is large compared to $\log_3 p$. However, Result 9.1 covers t^2-t-2 rounds with p^{t-2} data. This is greater than or equal to $t(t+3)/2+1$ for all $t \geq 6$. Moreover, when the number of branches t is large enough, Result 9.1 shows that the number of rounds must be approximately doubled to achieve the desired security level.

As discussed in Chapter 9, Result 9.1 can be improved in several ways. A key-recovery attack on t^2-t-1 rounds with time complexity p^{t-1} is readily obtained. Furthermore, for GMiMC-crf, Result 9.3 yields a t^2+t-2 round distinguisher using p^{t-1} data. This is greater than or equal to $t(t+3)/2+1$ for all $t \geq 3$. It seems likely that similar improvements are possible for GMiMC-erf. However, GMiMC is typically instantiated with $t \ll \log_3 p$, so that algebraic attacks are dominant. For these instances, no full-round attacks are obtained in any case. Since the techniques from Chapter 9 mostly aim at maximizing the number of rounds, extending them to the expanding case is left as future work.

In the hash function case, the data complexity of the distinguishers above exceeds the cost of generic attacks on the sponge construction. Hence, there is no direct impact on the security of the hash function. Nevertheless, as shown in the following example, it is possible to obtain distinguishers on the underlying permutation. The significance of such distinguishers has been discussed in Chapter 1. In the Crypto paper [43], reduced-round collision attacks are obtained by combining these results with algebraic techniques.

Example 10.1. The most efficient instance of GMiMC-erf proposed by StarkWare is of the expanding type with $p = 2^{61} + 20 \cdot 2^{32} + 1$, $t = 12$ branches and $r = 101$ rounds. Repeating the basic iterative truncated differential for expanding Feistel ciphers from Section 9.3.1 eight times covers $8 \times 12 = 96$ rounds with probability approximately $1/p^8$. Appending five rounds yields a full-round truncated differential with the same probability. Using $2p^6$ input structures of size p, the data complexity becomes $2p^7 \approx 2^{428}$.

This can be improved by using the iterative t-round truncated differential with difference set $A = \{(0, \ldots, 0, a, b, c) \mid a, b, c \in \mathbb{F}_p\}$, which admits larger input structures. The probability over the first 96 rounds is still $1/p^8$, and appending

five rounds does not decrease the probability. Since $p_{\mathsf{trail}} = p_{\mathsf{ideal}} = 1/p^8$, only $2p^2$ structures of size p^3 are necessary and the data complexity is $2p^5 \approx 2^{306}$.

It is possible that further improvements can be obtained using the more advanced techniques from Chapter 9. ▷

10.2.3 Other attacks on GMiMC

In addition to the truncated differential attacks described above, the Crypto paper [43] presents impossible differential and integral attacks on GMiMC-erf. The impossible differential attack is not a full-round attack, but nevertheless disproves the claims of the designers [6, page 46]. The integral attacks result in full-round zero-sum partitions for the GMiMC-erf permutation, but this does not affect the security of the hash function.

10.3 Cryptanalysis of HadesMiMC

In Sections 10.3.3 and 10.3.4, two attacks against HadesMiMC are described. The first attack is an integral distinguisher covering all rounds except the first two for most sets of parameters. The second one is a full-round preimage attack that requires some assumptions on the MDS matrix defining the linear layer.

Both attacks are based on a weakness of the partial S-box layer used in HadesMiMC, which is described in Section 10.3.2. For some choices of the linear layer, this weakness extends over an arbitrary number of rounds. Although the designers of HadesMiMC do not mention any requirements on the MDS matrix, they provide several suggestions. It turns out that the classes of matrices suggested by the authors contain several weak instances.

10.3.1 Specification of HadesMiMC

HadesMiMC is a family of permutations described by Grassi *et al.* [152], following a new design strategy for block ciphers called HADES. The HADES construction aims to decrease the number of S-boxes relative to traditional AES-like designs. Reducing the number of S-boxes is important for many applications and has often been achieved using a partial S-box layer, *i.e.*, an S-box layer which does not operate on the whole internal state.

However, several attacks [20, 120, 124, 238] on constructions with a partial S-box layer have shown that the security level of such designs can be difficult to

estimate. The basic principle of HADES is to combine both aspects: the inner rounds of the cipher use a partial S-box layer to increase the resistance against algebraic attacks at a reduced implementation cost, whereas the outer rounds use a full S-box layer. The resistance against statistical attacks is analyzed by removing the inner rounds, whereas the resistance against algebraic attacks depends on the inner rounds.

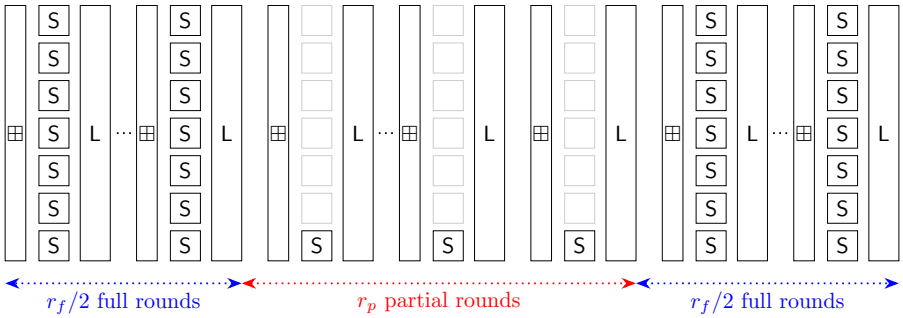


Figure 10.1: The HadesMiMC construction with $t = 7$.

HadesMiMC [152, Section 3] is a block cipher following the HADES construction dedicated to MPC applications or to STARK proof systems, where the S-box is defined by the cube mapping over a finite field and the linear layer L is given by multiplication with a $t \times t$ MDS matrix.

Two specialized instances of HadesMiMC are proposed by Grassi *et al.* in [150]: Starkad is defined over a field of even characteristic and odd absolute degree[5], whereas Poseidon is defined over a field of odd prime order p with $p \not\equiv 1 \pmod 3$. In both cases the partial rounds consist of a single S-box operating on the last coordinate of the state. The S-box is given by the cube function $x \mapsto x^3$. For the parameters proposed by StarkWare, the number of full rounds is equal to 8 and the number of partial rounds varies between 40 and 88.

10.3.2 Property of partial rounds

It is not surprising that there exists an affine subspace of $\mathbb{F}_p^t$ that is mapped to another affine subspace under one or a few partial rounds of HadesMiMC. However, it turns out that such subspaces can exist for an arbitrary number of rounds if the linear layer has a low-degree minimal polynomial. This is consequence of Theorem 10.1 below. For the following results, let $\delta_t = (0, 0, \ldots, 0, 1)$ in $\mathbb{F}_p^t$.

[5]The field degree must be odd to ensure that $x \mapsto x^3$ is a bijection.

Theorem 10.1. *Let* $\mathsf{F} : \mathbb{F}_q^t \to \mathbb{F}_q^t$ *denote a permutation obtained from* $r \geq 1$ *partial HadesMiMC rounds instantiated with linear layer* $\mathsf{L} : x \mapsto M^\mathsf{T} x$*, where* M *is a* $t \times t$ *matrix. For all* x *in* $\mathbb{F}_q^t$*, the subspace* $V = \mathrm{Span}\{\delta_t, M\delta_t, \ldots, M^{r-1}\delta_t\}^\perp$ *of* $\mathbb{F}_q^t$ *satisfies* $\mathsf{F}(x+V) \subseteq \mathsf{F}(x)+\mathsf{L}^r(V)$*. Furthermore, if the minimal polynomial of* M *has degree* h*, then* $\dim V \geq t - \min\{h, r\}$*.*

Proof. Clearly, $\dim V$ satisfies the lower bound if $M^h x = \sum_{i=1}^{h} \alpha_i M^{i-1} x$ for some coefficients $\alpha_1, \ldots, \alpha_h$ in $\mathbb{F}_q$. Let $\mathsf{F} = \mathsf{R}_r \circ \cdots \circ \mathsf{R}_1$, where R_i denotes the i^{th} partial round of HadesMiMC. Since the last coordinate of any v in V is zero, *i.e.* $v \perp \delta_t$, the image of $x + V$ by the partial S-box layer is a coset of V. It follows that $\mathsf{R}_1(x + v) = \mathsf{R}_1(x) + \mathsf{L}(v)$. Similarly, for round $i = 2, \ldots, r$, it holds that $\mathsf{R}_i(x_i + \mathsf{L}^{i-1}(v)) = \mathsf{R}_i(x_i) + \mathsf{L}^i(v)$ if $\mathsf{L}^{i-1}(v) \perp \delta_t$ or equivalently $v \perp M^{i-1}\delta_t$. □

Theorem 10.1 will be used in Section 10.3.3 to derive better integral properties over the partial rounds of HadesMiMC. After submitting the paper [43] to Crypto, but before its publication, Keller and Rosemarin independently obtained Theorem 10.1. They do not deduce attacks on HadesMiMC using this observation, but instead focus on lower bounding the dimension of V for various choices of M. Their work was published at Eurocrypt 2021 [176]. In Section 10.3.4, the following dual variant of Theorem 10.1 will be used to obtain preimage attacks.

Theorem 10.2. *Let* $\mathsf{F} : \mathbb{F}_q^t \to \mathbb{F}_q^t$ *denote a permutation obtained from* $r \geq 1$ *partial HadesMiMC rounds instantiated with round constants* $c_1, \ldots, c_r$ *and linear layer* $\mathsf{L} : x \mapsto Mx$*, where* M *is a* $t \times t$ *matrix. If* v *is an element of the subspace* $V = \mathrm{Span}\{M\delta_t, M^2\delta_t, \ldots, M^r\delta_t\}^\perp$ *of* $\mathbb{F}_q^t$*, then every* x *in* $\mathbb{F}_q^t$ *satisfies* $v^\mathsf{T}\mathsf{F}(x) = v^\mathsf{T}M^r x + \sum_{i=1}^{r} v^\mathsf{T}M^{r+1-i}c_i$*. Furthermore, if the minimal polynomial of* M *has degree* h*, then* $\dim V \geq t - \min\{h, r\}$*.*

Proof. As in Theorem 10.1, it is easy to see that $\dim V \geq t - \min\{h, r\}$. Let $\mathsf{F}_r = \mathsf{R}_r \circ \cdots \circ \mathsf{R}_1$, with R_i the i^{th} partial round of HadesMiMC. If v is orthogonal to $M\delta_t$, then the last coordinate of $v^\mathsf{T}M$ is zero because $v^\mathsf{T}M\delta_t = 0$. Hence,

$$v^\mathsf{T}\mathsf{R}_i(x) = v^\mathsf{T}M(x + c_i) = v^\mathsf{T}Mx + v^\mathsf{T}Mc_i\,.$$

Taking $i = 1$ establishes the result for $r = 1$. For $r > 1$, the result follows by induction. Indeed, if v is orthogonal to $M\delta_t$ then

$$\begin{aligned} v^\mathsf{T}(\mathsf{R}_r \circ \cdots \circ \mathsf{R}_1)(x) &= v^\mathsf{T}M(\mathsf{R}_{r-1} \circ \cdots \circ \mathsf{R}_1)(x) + v^\mathsf{T}Mc_r \\ &= v^T M^r x + v^\mathsf{T}Mc_r + \sum_{i=1}^{r-1} v^\mathsf{T}M^{r+1-i}c_i\,, \end{aligned}$$

where the second equality follows from the induction hypothesis using the assumption that $M^{\mathsf{T}}v$ belongs to $\mathrm{Span}\{M\delta_t, \ldots, M^{r-1}\delta_t\}^{\perp}$. □

The linear layers of Starkad and Poseidon are chosen such that $L_{i,j} = 1/(x_i + x_j + a)$ where a and $x_1, \ldots, x_t$ are distinct elements of $\mathbb{F}_q$ [149]. These matrices are known as *Cauchy matrices*. The following result shows that, for Starkad instances with t a power of two, there exist weak choices of $x_1, \ldots, x_t$ that enable the preimage attack from Section 10.3.4.

Theorem 10.3. *Let $G = \{x_1, \ldots, x_t\}$ be an additive subgroup of $\mathbb{F}_{2^n}$ of order t and let a in $\mathbb{F}_{2^n} \setminus G$. The $t \times t$ Cauchy matrix M defined by $M_{i,j} = 1/(x_i + x_j + a)$ satisfies $M^2 = b^2 I$ with $b = \sum_{i=1}^{t} 1/(x_i + a)$.*

Proof. The coordinates of M^2 satisfy

$$M_{i,j}^2 = \sum_{k=1}^{t} \frac{1}{x_i + x_k + a} \times \frac{1}{x_j + x_k + a} = \sum_{x \in a+G} \frac{1}{x(x + x_i + x_j)}.$$

For $i = j$, the result follows immediately. Hence, it suffices to prove that $M_{i,j}^2 = 0$ for $i \neq j$. Since $x_i \neq x_j$ for $i \neq j$, it holds that $g = x_i + x_j \in G \setminus \{0\}$. Finally, the result follows from

$$M_{i,j}^2 = \sum_{x \in a+G} \frac{1}{x(x+g)} = \frac{1}{g} \sum_{x \in a+G} \left(\frac{1}{x} + \frac{1}{x+g} \right) = 0.$$

The last equality follows from $\sum_{x \in a+G} 1/x = \sum_{x \in a+G} 1/(x+g)$. □

A special case of Theorem 10.3 is discussed by Youssef *et al.* [288, §3.2]. For an extension $\mathbb{F}_2(\zeta)$ of degree n, they show that the choice $x_i = \sum_{j=1}^{\log_2 t} d_j \, \zeta^{j-1}$ with $d_1, \ldots, d_{\log_2 t}$ the binary digits of $i - 1$ results in a Cauchy matrix M such that $M^2 = b^2 I$.

Alternatively, the HadesMiMC authors propose [151, Appendix B] the use of a matrix of the form AB^{-1} where both A and B are Vandermonde matrices with generating elements a_i and b_i. In this case, if $a_i = b_i + r$ for some r in $\mathbb{F}_q$, then the resulting MDS matrix will be an involution if $\mathbb{F}_q$ has characteristic two [246]. In odd characteristic, one obtains an involution whenever $a_i = -b_i$.

10.3.3 Integral distinguishers

In HadesMiMC, the number of rounds has been chosen such that the algebraic degree of each output coordinate is close to $t(q-1)$, similar to the behaviour of

most permutations. Since the degree is upper bounded by 3^r after r rounds, at least $\lceil \log_3(t(q-1)) \rceil$ rounds are necessary to reach total degree $t(q-1)$. For example, if $t = 12$ and $q = 2^{61} + 20 \cdot 2^{32} + 1$ (a Poseidon instance proposed by StarkWare), then at least 41 rounds out of 48 in total are necessary. For $t = 12$ and $q = 2^{63}$ (a Starkad instance proposed by StarkWare), 43 rounds out of 51 in total are necessary.

It is worth pointing out that the theory from Chapter 5 implies that 3^r cannot be a tight bound on the degree of an iterated partial layer, except for the first few iterations (when the degree is below $\log_3(q-1)$). More accurate bounds, which yield full-round distinguishers on some instances, can be obtained using a variant of the reasoning in Example 5.9. However, this chapter uses a different approach based on Theorem 10.1 that is more useful when the data complexity is low. Improving this using the results from Chapter 5 is left as future work.

The basic idea is to improve upon the trivial bound by choosing a specific subspace of inputs. The following distinguisher applies to all partial rounds and the last four full rounds. It is meaningful in both the block-cipher setting and the permutation setting. By Theorem 10.1, there exists a one-dimensional subspace V of $\mathbb{F}_q^t$ such that V is mapped to a coset $\gamma + W$ of $W = \mathsf{L}^{t-1}(V)$ after $t-1$ partial rounds. The spaces V and $\gamma + W$ are indicated in Figure 10.2. For x in V, let $f(x)$ be the i^{th} coordinate of the output of HadesMiMC as shown in Figure 10.2. If $W = \text{Span}\{w\}$, then

$$\sum_{x \in V} f(x) = \sum_{x \in \gamma + W} (\mathsf{R}_r \circ \cdots \circ \mathsf{R}_1)(x)_i = \sum_{z \in \mathbb{F}_q} (\mathsf{R}_r \circ \cdots \circ \mathsf{R}_1)(\gamma + zw)_i = 0\,,$$

where $\mathsf{R}_1, \ldots, \mathsf{R}_r$ are $r \leq \lfloor \log_3(q-2) \rfloor$ full or partial HadesMiMC rounds. The last equality is due to the fact that $\mathsf{R}_r \circ \cdots \circ \mathsf{R}_1$ is a function of degree at most $q-2$.

The above yields a distinguisher on $\lfloor \log_3(q-2) \rfloor + t - 1$ rounds, starting after the initial full rounds. For most sets of concrete parameters, this actually exceeds the recommended number of rounds (except the first $r_f/2$ full rounds) for both Poseidon and Starkad. Furthermore, if the degree of the minimal polynomial of the linear layer L is less than $t-1$, then by Theorem 10.1 the distinguisher covers an arbitrary number of partial rounds.

By extending the above approach in the backward direction, a zero-sum partition for a (slightly) larger number of rounds can be obtained – although zero-sum partitions are only meaningful in the known-key or permutation setting. The problem is that contrary to GMiMC, the inverse round function of HadesMiMC has a much higher degree than the round function itself. Indeed, the inverse of the cube function over $\mathbb{F}_q$ is given by $x \mapsto x^{(2q-1)/3}$. Using classical degree

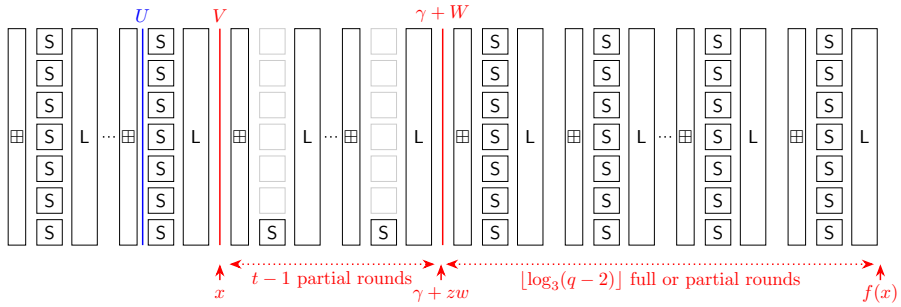


Figure 10.2: Integral and zero-sum partition distinguishers against HadesMiMC.

bounds, a degree lower than $(q-2)$ can not be guaranteed for more than a single inverse round.

However, because $\mathsf{L}^{-1}(V)$ is one-dimensional, an additional S-box layer can be overcome. Specifically, there exists a vector $v = (v_1, \ldots, v_t)$ such that

$$\mathsf{L}^{-1}(V) = \left\{(xv_1, xv_2, \ldots, xv_t) \mid x \in \mathbb{F}_q\right\}.$$

The image of $\mathsf{L}^{-1}(V)$ under the inverse of the full S-box layer is then equal to

$$U = \left\{(x^{1/3}v_1^{1/3}, x^{1/3}v_2^{1/3}, \ldots, x^{1/3}v_t^{1/3}) \mid x \in \mathbb{F}_q\right\}$$

Hence, this image is again a one-dimensional vector space. That is, $U = \mathrm{Span}\{u\}$ with $u_i = v_i^{1/3}$ for $i = 1, \ldots, t$. This particular property does not extend to more rounds because of the addition of a round constant. Prepending one more round yields a zero-sum partition, since the degree of the inverse S-box layer does not exceed $q-2$.

Table 10.2 summarizes the implications of the results above for the instances of Poseidon and Starkad proposed by StarkWare. For many parameter choices, there is an integral distinguisher with data complexity q on all except the initial four rounds. In the known-key or permutation setting, there is a zero-sum partition that additionally covers two of the initial four rounds. It is worth reiterating that for instances of HadesMiMC with a linear layer with minimal polynomial of degree less than $t-1$, these results can be extended to an arbitrary number of partial rounds.

Table 10.2: Number of full and partial rounds (r_f and r_p) of HadesMiMC covered by the distinguishers in this section. The initial two full rounds are only included for the zero-sum partitions.

Security level	t	Poseidon			Starkad		
		$\lceil \log_2 q \rfloor$	Proposed r_f, r_p	Covered r_f, r_p	$\lceil \log_2 q \rfloor$	Proposed r_f, r_p	Covered r_f, r_p
128 bit	12	61	8, 40	*2*+4, 45	63	8, 43	*2*+4, 46
	4	125	8, 81	*2*+4, 77	125	8, 85	*2*+4, 77
	12	125	8, 83	*2*+4, 85	125	8, 86	*2*+4, 85
	3	253	8, 83	*2*+4, 157	255	8, 85	*2*+4, 158
	12	253	8, 85	*2*+4, 165	255	8, 88	*2*+4, 166
256 bit	8	125	8, 82	*2*+4, 81	125	8, 86	*2*+4, 81
	14	125	8, 83	*2*+4, 87	125	8, 83	*2*+4, 87

10.3.4 Preimage attacks

Theorem 10.2 shows that there are linear relations between the inputs and outputs of an arbitrary number of partial rounds of HadesMiMC when the degree of the minimal polynomial of the linear layer is lower than $t-1$. This can be used to setup a simplified system of equations for finding preimages, leading to a full-round preimage attack for some choices of the rate and capacity parameters of the sponge construction.

Suppose that L is such that the vector space V from Theorem 10.2 is of dimension d. In the worst case, $d = t-2$. By Theorem 10.2, there exists a matrix U_1 in $\mathbb{F}_q^{d\times t}$ such that $U_1\mathsf{F}(x) = U_1(\mathsf{L}^r(x) + a)$ for a known constant a and F the composition of the partial rounds. Indeed, let the rows of U_1 be a basis for V. Furthermore, let U_2 in $\mathbb{F}_q^{(t-d)\times t}$ be a matrix with row space complementary to the row space of U_1. Given a value y for the output of the partial rounds, one has the following equations in the input x:

$$\begin{aligned} U_1 y &= U_1(\mathsf{L}^r(x) + \textstyle\sum_{i=1}^{r} \mathsf{L}^{r+1-i}(c_i)) \\ U_2 y &= U_2\mathsf{F}(x)\,. \end{aligned} \tag{10.1}$$

Consider a HadesMiMC permutation in a sponge construction with rate k and capacity $c = t-k$. Computing preimages of a one-block digest $(h_1, \ldots, h_k)$ in $\mathbb{F}_q^k$ then corresponds to solving the system of equations $[\mathsf{HadesMiMC}(m\|\mathsf{iv})]_i = h_i$ for $i = 1, \ldots, k$ in the unknowns $m_1, \ldots, m_k$.

The idea of the attack is simple: for each guess of $U_2\mathsf{F}(x)$ in $\mathbb{F}_q^{t-d}$, replace the equations for the partial rounds by the linear equations (10.1) and solve the resulting system of equations by computing a Gröbner basis.

Below, it is shown that the time complexity of the attack is approximately

$$2\gamma\,(2\pi)^{-\omega/2}\,k^{2-\omega/2}\,e^{\omega k}\,3^{(\omega k+1)(r_f-1)}\,q^{t-d}\,, \tag{10.2}$$

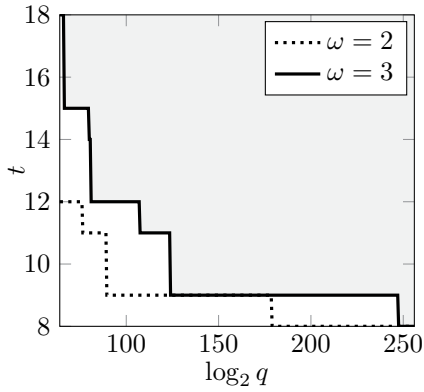
with ω and γ such that the cost of computing the row-reduced echelon form of an $n \times n$ matrix is γn^ω. The analysis below focuses on the case where the number of output elements is equal to the rate. This is the most challenging setting. Indeed, if the output size is smaller than the rate – as in some of the StarkWare challenges – then the preimage problem typically has many solutions. This allows the attacker to partially or completely avoid the guessing phase. If further degrees of freedom remain after fixing $U_2\mathsf{F}(x)$ completely, then one or more input elements can be fixed to an arbitrary value.

For example, for a linear layer with quadratic minimal polynomial, $r_f = 8$ and r_p arbitrary, Figure 10.3a shows for which choices of q and t an improvement over the generic security of the sponge construction is obtained. The insecure instances are shaded in grey. This area assumes conservative values for the cost of row-reduction, *i.e.* $\omega = 3$ and $\gamma = 3/2$. The cost itself is shown in Figure 10.3b. One should keep in mind that these figures correspond to the most challenging case, *i.e.* assuming that the hash output is of length k and no shorter.

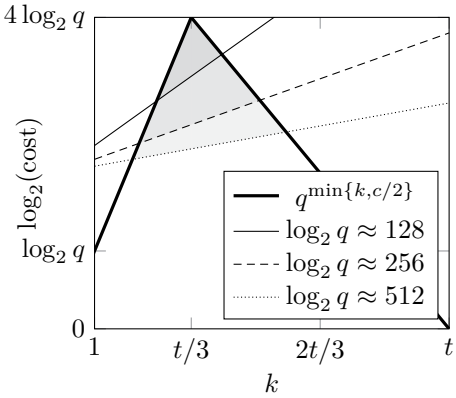
For the concrete parameters proposed by StarkWare, better-than-generic attacks on some variants are obtained assuming that the hash output has length $c \leq k$. Indeed, if $c \leq d/2 = t/2 - 1$, then a sufficiently large number of preimages is likely to exist so that it is no longer necessary to guess $U_2\mathsf{F}(x)$. In addition, input variables may be fixed until only $c+t-d = c+2$ free variables remain. This leads to a computational cost of $2\gamma\,(2\pi)^{-\omega/2}\,(c+2)^{2-\omega/2}\,e^{\omega(c+2)}\,3^{(\omega(c+2)+1)(r_f-1)}$. An overview of the concrete results is given in Table 10.3.

Table 10.3: Computational cost ($\mathbb{F}_q$-operations) of preimage attacks on different instances of HadesMiMC with digest length c, assuming a weak linear layer.

Security level	Variant	$\lceil \log_2 q \rfloor$	t	c	Computational cost $\omega \approx 2.8$	Computational cost $\omega = 3$
128 bit	Poseidon	253	11	1	2^{115}	2^{122}
	Starkad	255	11	1	2^{115}	2^{122}
256 bit	Poseidon	125	14	4	2^{221}	2^{236}
	Starkad	125	14	4	2^{221}	2^{236}



(a) Minimum t such that the cost is better than generic for some choice of k.



(b) Cost for different values of the rate k with $t = 12$ and $\omega = 3$.

Figure 10.3: Cost analysis of the preimage attack on HadesMiMC with a weak linear layer and $r_f = 8$. The shaded areas correspond to parameters for which the attack improves over the $q^{\min\{k,c/2\}}$ security level.

To conclude this section, the estimate (10.2) will be derived. The cost of solving a system of equations using Gröbner bases is dominated by two steps:

1. Computing a Gröbner basis with respect to a total degree term order such as the degree reverse lexicographic (degrevlex) order. For standard reduction algorithms such as Faugère's F4 and F5, the time required for this step can be upper bounded by [21]

$$T_{\mathsf{gb}} = \widetilde{\mathcal{O}}\left(\binom{D+k}{D}^{\omega}\right),$$

   for k variables, D the maximum degree of the Gröbner basis elements and ω the matrix-multiplication exponent.

2. Converting the degrevlex Gröbner basis to a Gröbner basis with respect to a lexicographic order. For the FGLM algorithm, the cost of this step can be estimated as [135]

$$T_{\mathsf{fglm}} = \widetilde{\mathcal{O}}(k \dim(\mathbb{F}_q[m_1, \ldots, m_k]/I)^{\omega}),$$

   where I is the ideal generated by the polynomials that define the system.

The time required to factor the univariate polynomials in the lexicographic Gröbner basis can be assumed to be negligible. Hence, the time complexity of the attack is dominated by $q^{t-d}\,(T_{\mathsf{gb}} + T_{\mathsf{fglm}})$.

To set up a system of preimage equations for HadesMiMC, two diametrical approaches can be considered. In the first strategy, one attempts to minimize the number of variables by setting up a system of high-degree polynomials relating the input and output of the permutation. In the second approach, intermediate variables are introduced at every round, leading to a system of many low-degree equations. The latter strategy is usually preferred, as it leads to a lower degree D. However, a routine calculation shows that reducing the number of variables is more important for the proposed attack. Hence, the former approach is used below.

Clearly, the S-box layer of the first round may be ignored in the analysis. Furthermore, since the HadesMiMC specification states that the last linear layer can be omitted, the last round could also be ignored. Nevertheless, this is not the case for Starkad and Poseidon, so this will not be taken into account in the analysis.

For each guess of $U_2\mathsf{F}(x)$, the digest coordinates $h_1, \ldots, h_k$ can be expressed as polynomials in the input (after the first S-box layer) of degree 3^{r_f-1}. In general, bounding D is highly nontrivial. However, for regular systems, Macaulay's bound [21,212] yields $D \leq (3^{r_f-1}-1)k+1$. Furthermore, small-scale experiments suggest that this bound is tight for this particular system of equations. It is hard to obtain theoretical estimates of $\dim(\mathbb{F}_q[m_1, \ldots, m_k]/I)$, but small-scale experiments suggest that it scales as $3^{k(r_f-1)}$, which is consistent with results obtained by Faugère and Perret [137]. Since the FGLM algorithm is able to exploit sparse linear algebra methods [136], it is reasonable to assume that T_{gb} is dominant compared to T_{fglm}.

Suppose that 3^{r_f-1} is much larger than k. Following [21, §1.3], it holds that

$$T_{\mathsf{gb}} \leq \gamma\, k\, (D - 3^{r_f-1} + 1) \binom{k+D-1}{D}^{\omega} \lessapprox \gamma\, k^2\, 3^{r_f-1} \binom{k+D-1}{D}^{\omega}.$$

In the above, the parameters γ and ω are such that the computational cost of computing the row-reduced echelon form of an $n \times n$ matrix is γn^{ω}. By Stirling's approximation,

$$\log \binom{k+D-1}{D} = \log \binom{k3^{r_f-1}}{k} \approx k + k(r_f-1)\log 3 - \log \sqrt{2\pi k}.$$

If computing the reduced row-echelon form of an $n \times n$ matrix takes time γn^{ω}, then it follows that

$$T_{\mathrm{gb}} \lessapprox \gamma\, (2\pi)^{-\omega/2}\, k^{2-\omega/2}\, e^{\omega k}\, 3^{(\omega k+1)(r_f-1)}\,.$$

Multiplying by q^{t-d} yields (10.2).

10.4 Cryptanalysis of the Legendre PRF

This section presents attacks on the Legendre PRF and its generalizations. The Legendre PRF and its higher-order variant are introduced in Section 10.4.1. Section 10.4.2 briefly discusses Khovratovich's attack, and a table-based variant of this attack is presented in Section 10.4.3. This variant is more suitable for the concrete challenges that were proposed by the Ethereum foundation, as only limited data (2^{20} consecutive outputs) are provided.

The new attack is presented in Section 10.4.4. Section 10.4.5 generalizes the attack to the degree-d Legendre PRF, and Section 10.4.6 exhibits weak keys for this extension. Attacks on the Jacobi and power residue PRF are given in Section 10.4.7 and Section 10.4.8 respectively. Some final comments on the solution to the concrete challenges are given in Section 10.4.9.

10.4.1 Specification of the Legendre PRF

For an odd prime p, the Legendre symbol of an element a of $\mathbb{F}_p$ is defined as

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a = b^2 \text{ for some } b \text{ in } \mathbb{F}_p^\times \ , \\ 0 & \text{if } a = 0 \ , \\ -1 & \text{otherwise .} \end{cases}$$

The distribution of Legendre symbols has been a subject of study for number theorists at least since the early 1900s [3,109,110,168,278]. The Weil bound [285] implies that the number of occurrences of a fixed pattern of l nonzero Legendre symbols among the integers $1, 2, \ldots, p-1$ modulo p is $p/2^l + \mathcal{O}(\sqrt{p})$ as $p \to \infty$. In 1988, Damgård [108] conjectured pseudorandom properties of the sequence

$$\left(\frac{k}{p}\right), \left(\frac{k+1}{p}\right), \left(\frac{k+2}{p}\right), \ldots,$$

where k has been sampled from $\mathbb{F}_p$ uniformly at random. He proposed to use this construction as a pseudorandom number generator. In 2016, Grassi *et al.* [153] proposed the same construction as a candidate pseudorandom function and showed that it can be evaluated efficiently in the multiparty computation setting. Concretely, the *Legendre pseudorandom function* $L_k(x)$ is defined by mapping the Legendre symbol with a secret shift k to $\{0, 1\}$:

$$L_k(x) = \left\lfloor \frac{1}{2}\left(1 - \left(\frac{k+x}{p}\right)\right)\right\rfloor , \tag{10.3}$$

where p is a public prime number. The following definition will be convenient when dealing with expressions such as (10.3).

Definition 10.1 (Legendre function)**.** For an odd prime p, the Legendre function $l : \mathbb{F}_p \to \mathbb{F}_2$ is defined as

$$l(x) = \left\lfloor \frac{1}{2}\left(1 - \left(\frac{x}{p}\right)\right)\right\rfloor .$$

It maps quadratic residues to 0 in $\mathbb{F}_2$ and quadratic non-residues to 1 in $\mathbb{F}_2$.

Khovratovich considers the following extension of the Legendre PRF.

Definition 10.2 (Degree-d Legendre PRF)**.** Let p be an odd prime and d a positive integer. The degree d-Legendre PRF over $\mathbb{F}_p$ is a family of functions $L_k : \mathbb{F}_p \to \mathbb{F}_2$ such that for each k in $\mathbb{F}_p^d$,

$$L_k(x) = l\big(x^d + \textstyle\sum_{i=0}^{d-1} k_{i+1}\, x^i\big).$$

For $d = 1$, this reduces to (10.3) and L_k is called *the* Legendre PRF over $\mathbb{F}_p$.

Remark 10.1. The Legendre symbol is *multiplicative*, *i.e.* for all a and b in $\mathbb{F}_p$,

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) .$$

In terms of the Legendre function l, one has $l(ab) = l(a) + l(b)$ if $ab \neq 0$. ▷

The analysis below often considers sequential evaluations of L_k starting from a point a with an additive step b. This leads to Definition 10.3.

Definition 10.3 (L-sequences)**.** Let p be an odd prime, m a positive integer and a, b elements of $\mathbb{F}_p$. An *arithmetic L-sequence of length m with starting point a and stride b* is an $\mathbb{F}_2^m$-vector

$$L_k(a + b\,[m]) = (L_k(a), L_k(a + b), \ldots, L_k(a + (m-1)b)) .$$

To justify the correctness of the attacks, the following assumption will be used.

Assumption 10.1. Let p be an odd prime and d a positive integer. Let $m = d\lceil \log p \rceil$. For all k in $\mathbb{F}_p^d$, then as $p \to \infty$, there exist at most $\mathcal{O}(1)$ keys k' in $\mathbb{F}_p^d$ such that $L_{k'}([m]) = L_k([m])$.

Legendre symbols, and hence the Legendre function, can be efficiently computed using the law of quadratic reciprocity. That is, for distinct odd primes p and q,

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\,\frac{q-1}{2}} .$$

This leads to an algorithm requiring $\mathcal{O}(\log p)$ arithmetic operations, or $\mathcal{O}(\log^2 p \log\log p)$ bit operations. Brent and Zimmerman [80] give an asymptotically better algorithm with complexity $\mathcal{O}(\log p \log^2 \log p)$. In the remainder of this section, the cost of an attacks will often be expressed in terms of the number of Legendre symbol computations.

10.4.2 Previous attacks

Khovratovich [178] describes a chosen plaintext attack on the Legendre PRF L_k that recovers k with $\mathcal{O}(\sqrt{p}\log p)$ queries to L_k. It is based on a memoryless collision search between two functions and can be summarized as follows.

Let $m = \lceil \log p \rceil$ and consider the functions $x \mapsto L_k(x+[m])$ and $x \mapsto L_0(x+[m])$. The L-sequence $L_k(x+[m])$ is available by querying the Legendre PRF, whereas $L_0(x+[m])$ does not depend on k. By Assumption 10.1, a collision between $x \mapsto L_k(x+[m])$ and $x \mapsto L_0(x+[m])$ yields k with high probability. Indeed, let a and b in $\mathbb{F}_p$ such that $L_k(a+[m]) = L_0(b+[m])$. Equivalently,

$$L_0(a+k+[m]) = L_0(b+[m])\,.$$

By Assumption 10.1, the number of keys k satisfying the above equality is $\mathcal{O}(1)$.

Collisions between $x \mapsto L_k(x+[m])$ and $x \mapsto L_0(x+[m])$ can be found with a generic memoryless collision search method [220, 272] in $\mathcal{O}(\sqrt{p})$ evaluations of both functions. Since computing each L-sequence requires $m = \mathcal{O}(\log p)$ calls to L_k, the overall complexity sums up to $\mathcal{O}(\sqrt{p}\log p)$ queries to L_k and L_0.

Note that Khovratovich's original attack builds sequences of length m using arbitrary evaluations of the Legendre function L_k rather than consecutive ones. This difference does not affect the overall attack complexity, but by using L-sequences it will be possible to reduce the data complexity in Section 10.4.4.

Khovratovich [178] also presents a generalization of the above attack to the quadratic case and, ultimately, to arbitrary degrees. It recovers the key using $\mathcal{O}(p^{d-1}d\log p)$ Legendre symbol evaluations, given $\mathcal{O}(p)$ queries to L_k.

10.4.3 Table-based collision search

Before introducing the new attack in Section 10.4.4, Khovratovich's attack will be converted into a table-based collision attack. This makes it possible to trade off the data- and the time complexity of the attack.

Let M be the allowed number of queries to the oracle L_k, where $\log p \ll M < \sqrt{p}$. Let $m = \lceil \log p \rceil$ and let $M' = M - m + 1$. The attack proceeds as follows:

1. Store in a table $\mathcal{T}$ the pairs $(L_k(a + [m]), a)$ for all a in $\{0, \ldots, M' - 1\}$.
2. Sample b uniformly at random from $\mathbb{F}_p$ until $(L_0(b + [m]), a) \in \mathcal{T}$ for some a in $\{0, \ldots, M' - 1\}$. For each a corresponding to such a collision, a candidate key k' is recovered as $k' = b - a$. By Assumption 10.1, the number of candidate keys is at most $\mathcal{O}(1)$. Candidate keys k' can be tested by comparing one or more entries of $\mathcal{T}$ with the corresponding arithmetic L-sequences with starting point k'.

The first step requires M queries to L_k, from which one obtains M' arithmetic L-sequences that are stored using $\mathcal{O}(M \log p)$ memory. The second step requires $\mathcal{O}(p \log p/M)$ evaluations of the Legendre symbol and no additional memory is needed. Hence, the overall computational cost of the attack is $\mathcal{O}(M + p \log p/M)$.

Note that this variant of the attack reduces the query and time complexities by a $\log p$ factor compared to the memoryless collision search, although a significant amount of memory is employed.

Remark 10.2. The above attack can be made deterministic by choosing b in $\{0, \ldots, \lfloor p/M' \rfloor\}$ and considering the sequences $v = L_0(bM' + [m])$ in the second step of the attack. Indeed, for every k in $\mathbb{F}_p$, the arithmetic L-sequence at offset $M' \lceil k/M' \rceil$ will be computed in both steps of the attack and the correct key is guaranteed to be recovered after at most $\mathcal{O}(M + p \log p/M)$ Legendre symbol evaluations. ▷

10.4.4 Improved attack on the Legendre PRF

This section shows how Khovratovich's attack (Section 10.4.2) on the Legendre PRF can be improved when the total number of available queries is less than $\sqrt{p}$. Although, in its simplest form, the improved method requires additional memory, several techniques to reduce memory requirements while keeping the same overall time complexity will be discussed.

The attack is based on expanding the table $\mathcal{T}$ from Section 10.4.3 without increasing the number of queries M. The key idea is to exploit the multiplicative property of the Legendre symbol.

Lemma 10.1. *Let m be a positive integer and k in $\mathbb{F}_p$. For all a in $\mathbb{F}_p$ and b in $\mathbb{F}_p^\times$, it holds that*

$$L_{k/b}(a/b + [m]) = (l(b), \ldots, l(b)) + L_k(a + b[m]),$$

if none of the involved Legendre symbols evaluate to zero.

Proof. Immediate by the multiplicative property of the Legendre symbol. □

Theorem 10.4. *Let k be an element of $\mathbb{F}_p$ and $m \leq M$ positive integers. From the arithmetic L-sequence $L_k([M])$, one can efficiently extract $\sim M^2/m$ arithmetic L-sequences of the form $L_{k/b}(a/b + [m])$ for distinct (a, b) in $\mathbb{F}_p \times \mathbb{F}_p^\times$.*

Proof. Let b a positive integer such that $b \leq \lfloor M/m \rfloor$. By Lemma 10.1, we get

$$L_k(a + b[m]) = (l(b), \ldots, l(b)) + L_{k/b}(a/b + [m])$$

for any non-negative $a < M - bm + 1$. Hence, each b yields a total of $M - bm + 1$ L-sequences of length m. Moreover, since $L_k(a - b[m])$ is equal to the sequence $L_k(a - b(m-1) + b[m]) = L_k(a' + b[m])$ written in reverse order, we can consider negative values for b too, thus doubling the total number of sequences. Hence, the total number of arithmetic L-sequences of length m that can be extracted from $L_k([M])$ equals

$$2 \sum_{b=1}^{\lfloor M/m \rfloor} (M - bm + 1) \sim \frac{2M^2}{m} - m \sum_{b=1}^{M/m} b \sim \frac{2M^2}{m} - \frac{M^2}{m} = \frac{M^2}{m}\,. \qquad \square$$

Theorem 10.4 can be used to improve the table-based collision search from Section 10.4.3 as follows. As before, let M be the allowed number of queries, where $\log p \ll M < \sqrt{p}$. Let $m = \lceil \log p \rceil$. The attack proceeds as follows:

1. Query the sequence $L_k([M])$ and extract $\sim M^2/m$ sequences of the form $L_{k/b}(a/b + [m])$ from it. This is possible by Theorem 10.4. Store all of the triples $(L_{k/b}(a/b + [m]), a, b)$ in a table $\mathcal{T}$.
2. Sample c uniformly at random from $\mathbb{F}_p$ until $(L_0(c + [m]), a, b) \in \mathcal{T}$ for some a and b. For each pair (a, b) corresponding to such a collision, a candidate key k' is recovered as $k' = bc - a$. By Assumption 10.1, the number of candidate keys is at most $\mathcal{O}(1)$. As before, the correctness of candidate keys k' can easily be verified.

The first step of the attack requires M queries to L_k and $\sim M/m$ Legendre symbol evaluations. Storing the table $\mathcal{T}$ requires $\mathcal{O}(M^2)$ memory. In the second phase, an average of $\sim mp/M^2$ samples must be tested before a collision is found. Hence, the computational cost of this step is dominated by $\mathcal{O}(pm^2/M^2)$ Legendre symbol evaluations.

It follows that the overall cost of the attack is dominated by the extraction of $\mathcal{O}(M^2/m)$ sequences, the evaluation of $\mathcal{O}(M/m + p \log^2 p/M^2)$ Legendre symbols and a memory requirement of $\mathcal{O}(M^2)$. For $M < \sqrt{p}$, this is always an improvement over the attack from Section 10.4.3 – possibly after discarding some of the data.

The following paragraphs describe three optimizations that further reduce both the time and the memory complexity of the attack by a factor $\Omega(\log p)$.

Using consecutive values of c. The second step of the attack from Section 10.4.4 can be optimized by choosing consecutive values of c rather than uniform random samples. This approach allows reusing most of the Legendre symbol computations since, for example, $L_0(c+[m])$ and $L_0(c+1+[m])$ overlap almost completely.

A priori, this allows reducing the number of Legendre symbol computations by a factor of $\Omega(m)$. However, there is an important caveat: since the guesses for c are not independent, the expected number of iterations of the second step is no longer pm/M^2. To see why this is the case, recall that for any c, the algorithm will output the correct key k if there exists $(\cdot, a, b) \in \mathcal{T}$ such that $k = bc - a$. Since the table contains an entry $(\cdot, a, b)$ for all sufficiently small values of a and b, it is clear that if the table contains $(\cdot, a, b)$ such that $k = bc - a$ then it is also likely to contain $(\cdot, a + b, b)$ since $k = b(c + 1) - (a + b)$. Therefore, if c is a good guess, then $c + 1$ is also likely to be a good guess. Since the "good" values of c are clustered together in groups of size $\mathcal{O}(m)$, the required number of iterations will be $\mathcal{O}(pm^2/M^2)$, which means that the factor $\Omega(m)$ that was saved by using consecutive guesses for c is lost again.

However, this idea can still be used to reduce the memory complexity of the algorithm by only storing one entry $(\cdot, a, b)$ for each cluster of good c's. By storing only the triples $(\cdot, a, b)$ such that $|a| < |b|$, the size of the table can be reduced by a factor of $\Omega(m)$ without impacting the time complexity of the attack.

Expanding the number of L-sequences in the second step. Theorem 10.4 can be used to create new L-sequences from those computed during the second step of the attack. Indeed, after computing a large number of $w = \Omega(m)$ consecutive Legendre symbols $L_0(c + [w])$, it is possible to extract $\Omega(w^2/m^2)$ arithmetic subsequences of the form $L_0(c + c' + d[m])$ such that $|c'| < |d|$, with no need to compute additional Legendre symbols. Using the property that

$$L_0(c + c' + d[m]) = L_0((c + c')/d + [m]) + L_0(d)\,,$$

we can then do $\Omega(w^2/m^2)$ table lookups. Asymptotically, this allows to amortize away the cost of computing Legendre symbols. That is, the time complexity is dominated by the extraction of $\mathcal{O}(pm^2/M^2)$ subsequences rather than by the computation of $\mathcal{O}(pm^2/M^2)$ Legendre symbols.

Not storing reverse sequences. Since the sequence $a + b[m]$ is just the reverse of the sequence $a+b(m-1)-b[m]$, there is some redundancy in the table $\mathcal{T}$. Indeed, for every entry (s, a, b) in $\mathcal{T}$, the reverse sequence corresponding to the entry $(s', a + b(m-1), -b)$ is also stored.

If, instead, only the lexicographically smallest sequence is stored, then the memory requirements are reduced by a factor of two without affecting the overall time complexity just by looking up either the sequence $L_0(c + [m])$ *or* its reverse in $\mathcal{T}$, depending which comes first lexicographically.

10.4.5 Improved attack on the degree-d Legendre PRF

This section generalizes the attack from Section 10.4.4 to the degree-d Legendre PRF. The attack proceeds in essentially the same way as described in Section 10.4.4 for the linear case. The main difficulty is in extending Theorem 10.4 to the higher-degree case.

Lemma 10.2. *Let m be a positive integer and k in $\mathbb{F}_p$. For all a in $\mathbb{F}_p$ and b in $\mathbb{F}_p^\times$, there exists an invertible affine transformation $T_{a,b} : \mathbb{F}_p^d \to \mathbb{F}_p^d$ such that for all k in $\mathbb{F}_p^d$,*

$$L_{T_{a,b}(k)}([m]) = (l(b^d), \ldots, l(b^d)) + L_k(a + b[m]),$$

if none of the involved Legendre symbols evaluate to zero. Moreover, for any choice of (a, b) in $\mathbb{F}_p \times \mathbb{F}_p^\times$, the transformation $T_{a,b}$ can be efficiently computed.

Proof. Let f be the monic degree d polynomial with coefficient vector k, and let $T_{a,b}(k)$ be the coefficient vector of the monic polynomial $f(a + bx)/b^d$. It follows from the multiplicative property of the Legendre symbol that

$$L_{T_{a,b}(k)}([m]) = (l(b^d), \ldots, l(b^d)) + L_k(a + b[m]).$$

Furthermore, it is not hard to see that $T_{a,b}$ is invertible, affine and that it can be computed efficiently. □

Theorem 10.5. *Let k be an element of $\mathbb{F}_p^d$ and $m \leq M$ positive integers. From the arithmetic L-sequence $L_k([M])$, one can efficiently extract $\sim M^2/m$ arithmetic L-sequences of the form $L_{T_{a,b}(k)}([m])$ with $T_{a,b}$ as defined in Lemma 10.2 for distinct pairs (a, b) in $\mathbb{F}_p \times \mathbb{F}_p^\times$.*

Proof. The proof is analogous to that of Theorem 10.4. □

The table-based collision search can be modified as follows. Let M be the allowed number of consecutive queries to the oracle L_k and $m = d\lceil \log p \rceil$. The attack comprises the following steps:

1. Query the sequence $L_k([M])$ and extract $\sim M^2/m$ sequences of the form $L_{T_{a,b}(k)}([m])$ from it. This is possible by Theorem 10.5. Store all of the triples $(L_{T_{a,b}(k)}([m]), a, b)$ in a table $\mathcal{T}$.
2. Sample $\tilde{k}$ uniformly at random from $\mathbb{F}_p^d$ until $(L_{\tilde{k}}([m]), a, b) \in \mathcal{T}$ for some a and b. For each pair (a, b) corresponding to such a collision, one recovers a candidate key $k' = T_{a,b}^{-1}(\tilde{k})$. By Assumption 10.1, the number of candidate keys is at most $\mathcal{O}(1)$. As before, the correctness of candidate keys can easily be verified.

The computational cost of the first step is dominated by the extraction of $\mathcal{O}(M^2/m)$ sequences. For the second step, at most $\mathcal{O}(p^d\, m^2/M^2)$ Legendre symbols are expected to be evaluated. Hence, the total computational cost of the attack consists of $\mathcal{O}(M^2/m)$ sequence extractions and $\mathcal{O}(p^d\, d^2 \log^2 p/M^2)$ Legendre symbol evaluations. The attack requires $\mathcal{O}(M^2)$ memory.

For $d \geq 3$, the time complexity is minimized for $M = p$. The time complexity is then $\mathcal{O}(p^{d-2} d^2 \log^2 p)$ Legendre symbol computations. Hence, this method improves a factor of p in time over the attacks by Khovratovich [178].

10.4.6 Weak keys of the degree-d Legendre PRF

In this section, a large class of weak keys for the higher-degree Legendre PRF is exhibited. The attacks are based on the observation that for some keys, the corresponding monic polynomial factors as a product of polynomials of lower degree.

Consider the Legendre PRF of degree $d \geq 2$ over $\mathbb{F}_p$. Recall that the key k in $\mathbb{F}_p^d$ of the PRF corresponds to the monic polynomial $f(x) = x^d + \sum_{i=0}^{d-1} k_{i+1} x^i$. The attack in this section is based on the observation that, with high probability, the polynomial f has a factor of degree $t = \lfloor d/2 \rfloor$. In this case, there exist two monic polynomials g and h with $\deg g = t$ and $\deg h = d - t$ such that $f = gh$.

Suppose that the outputs for $m = d\lceil \log p \rceil$ arbitrary inputs are given, for example the sequence $L_k([m])$. By the multiplicativity of the Legendre symbol[6],

$$L_k([m]) = l(g([m])) + l(h([m]))\,.$$

Hence, finding the secret key k reduces to a simple collision search:

[6]For convenience, let $l(g([m])) = (l(g(0)), \ldots, l(g(m-1)))$ similar to Definition 10.3.

1. Query the sequence $L_k([m])$ from the PRF. For each monic polynomial g of degree t, store the pair $(L_k([m]) + l(g([m])), g)$ in a table $\mathcal{T}$.
2. Sample monic polynomials h of degree $d-t$ until $(l(h([m])), g) \in \mathcal{T}$ for some monic polynomial g of degree t. For each such g, recover a candidate key from the coefficients of gh. By Assumption 10.1, the number of candidate keys will be at most $\mathcal{O}(1)$.

For $t = \lfloor d/2 \rfloor$, this attack requires $\mathcal{O}(p^{\lfloor d/2 \rfloor} d \log p)$ bits of memory and its time complexity is dominated by $\mathcal{O}(p^{\lceil d/2 \rceil} d \log p)$ operations. These estimates use the fact that all Legendre symbols modulo p can be precomputed in $\mathcal{O}(p)$ operations. The attack requires only $m = \mathcal{O}(d \log p)$ queries to the PRF.

Using Van Oorschot-Wiener golden collision search [272], an improved time-memory trade-off can be obtained: given M bits of memory, the key can be recovered with a time complexity of $\mathcal{O}(d \log p \sqrt{p^{3d/2}/M})$ Legendre symbol evaluations.

Even if the polynomial f does not have a factor of degree exactly $\lfloor d/2 \rfloor$, it might still have a factor of large degree $t < \lfloor d/2 \rfloor$. In this case, the same strategy results in an attack with time complexity $\mathcal{O}(p^{d-t} d \log p)$ and memory complexity $\mathcal{O}(p^t d \log p)$. This gives a trade-off between more efficient attacks on a smaller fraction of keys (when t is large) or less efficient attacks on a larger fraction of the keys (when t is small). This trade-off is illustrated in Figure 10.4. The figure shows the time complexity of the attack for a desired fraction of weak keys.

The construction of Figure 10.4 is based on the following fact [260]: the fraction of monic degree-d polynomials whose factorization has exactly c_i monic irreducible factors of degree i is $1/\prod_{i=1}^{d} c_i!\, i^{c_i}$ as $p \to \infty$. By summing these probabilities over all integer partitions of d that allow a $(t, d-t)$ split, one obtains the probability that a uniformly random key is weak.

It follows from the above that if the key is chosen uniformly at random, then the higher-degree Legendre PRF has security only up to the birthday bound. To completely prevent this class of attacks, one can choose the key k such that the corresponding polynomial f is irreducible.

Reduction to the unique n-XOR problem. More generally, the secret polynomial could factor into n polynomials of degree roughly d/n. For example, if d is divisible by n and $f = \prod_{i=1}^{n} f_i$ with $\deg f_i = d/n$, then

$$L_k([m]) = \sum_{i=1}^{n} l(f_i([m])).$$

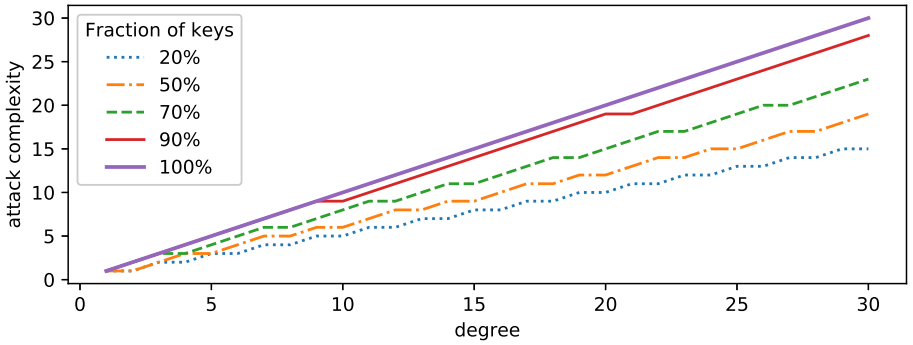


Figure 10.4: The complexity of the attack, measured as a power of p, as a function of the degree of f and the desired fraction of keys we want to attack.

That is, it suffices to find a solution to a variant of the n-XOR problem. Specifically, since each list has length $p^{d/n}$, a unique solution is expected. This makes Wagner's approach [279] inapplicable, but some improvements over the attack above are nevertheless possible.

In particular, for $k = 4$, the algorithm of Chose, Joux and Mitton [92] leads to a time complexity $\tilde{\mathcal{O}}(p^{d/2})$ with only $\tilde{\mathcal{O}}(p^{d/4})$ memory. Corresponding time-memory trade-offs can also be obtained.

There exist asymptotically better quantum algorithms. Bernstein *et al.* [33] give an $\widetilde{\mathcal{O}}(p^{0.3d})$ algorithm requiring $\widetilde{\mathcal{O}}(p^{0.2n})$ quantum-accessible quantum memory for $k = 4$. For any $k \geq 3$, Naya-Plasencia and Schrottenloher [221] give algorithms running in time $\widetilde{\mathcal{O}}(p^{\beta_k d})$ where $\beta_k = (k + \lceil k/5 \rceil)/(4k)$ using $\widetilde{\mathcal{O}}(p^{0.2n})$ quantum-accessible quantum memory. For $k = 3$, there is an algorithm using $\widetilde{\mathcal{O}}(p^{d/3})$ time and $\widetilde{\mathcal{O}}(p^{d/3})$ quantum-accessible *classical* memory.

10.4.7 Cryptanalysis of the Jacobi PRF

The Jacobi pseudorandom generator was proposed by Damgård [108] as a variation on the Legendre PRG. In this case, the public modulus is taken to be a product $n = \prod_{i=1}^{l} p_i$ of odd primes. Recall that the Jacobi symbol of an integer a is defined as

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{l} \left(\frac{a}{p_i}\right).$$

As discussed by Damgård [108, §5], the Jacobi PRG is potentially more efficient because it can be computed as the exclusive-or of several Legendre PRGs

with a relatively small modulus. In addition, Damgård showed that if the Legendre generator is weakly unpredictable, then the Jacobi generator is strongly unpredictable. A generator is defined to be weakly unpredictable if, for all polynomials f, there exist only finitely many integers $m \geq 0$ such that the next output bit in a sequence of length m can be predicted with probability greater than $1 - 1/f(m)$. Similarly, the generator is said to be strongly unpredictable if the probability of successful prediction exceeds $1/2 + 1/f(m)$ for only finitely many m. For a more formal definition, see [108, §3] and references therein.

This section investigates the security of the Jacobi PRF in the chosen-plaintext setting. Whereas the unpredictability result of Damgård could be regarded as a positive result related to the security of the Jacobi PRF, it remains inconclusive concerning its concrete security. Indeed, strong unpredictability is a weaker property than PRF-security and, in addition, it is only an asymptotic notion of security.

The cost of a key-recovery attack on the Jacobi PRF is at least the cost of attacking a Legendre PRF corresponding to a prime factor of the modulus. The following chosen-plaintext key-recovery attack on the Jacobi PRF below nearly attains this lower bound. Hence, for most purposes, the Jacobi PRF offers little benefit over the Legendre PRF.

Let $n = \prod_{i=1}^{m} p_i$ with $p_1, \ldots, p_m$ distinct odd primes – it can be assumed that the prime factors of n are distinct because

$$\left(\frac{x+k}{n}\right) = \left(\frac{x+k}{\prod_{i=1}^{m} p_i^{e_i}}\right) = \prod_{\substack{i=1 \\ e_i \text{ odd}}}^{m} \left(\frac{x+k}{p_i}\right).$$

Let $\lambda_j = \prod_{\substack{i=1 \\ i \neq j}}^{m} p_i$ and let λ'_j in $\mathbb{Z}$ such that $\lambda_j \lambda'_j \equiv 1 \pmod{p_j}$. Then

$$\left(\frac{\lambda_j\, x+k}{n}\right) = \prod_{i=1}^{m} \left(\frac{\lambda_j\, x+k}{p_i}\right) = \left(\frac{\lambda_j}{p_j}\right)\left(\frac{k}{n/p_j}\right)\left(\frac{x+\lambda'_j\, k}{p_j}\right).$$

Hence, in the chosen-plaintext setting, the key-recovery attack on the Legendre PRF from Section 10.4.4 can be used to recover the key modulo p_j. The Legendre symbol of k modulo n/p_j is not known to the attacker, but it is constant so the cost of the attack is increased by a factor of at most two. Given the value of the key modulo each prime factor of n, the Chinese remainder theorem yields the value of the key modulo n. Hence, key recovery for the Jacobi symbol costs at most $\mathcal{O}(mM^2 + \sum_{i=1}^{m} p_i \log^2 p_i / M^2)$ Legendre symbol evaluations. The same strategy is applicable to the higher-degree case and can be combined with the attacks in Section 10.4.8 below.

10.4.8 Cryptanalysis of the power residue PRF

The MPC protocol of Grassi *et al.* [153] for computing the Legendre PRF requires only three rounds of communication, which makes the Legendre PRF superior among the PRF constructions investigated by Grassi *et al.* in terms of latency. However, since the Legendre PRF only produces one bit of output, it does not compare favorably in terms of throughput to *e.g.* MiMC, GMiMC or HadesMiMC.

To mitigate this limitation of the Legendre PRF one can, as proposed by Damgård [108], consider higher power residue symbols rather than quadratic residue symbols. If r divides $p-1$, then the r^{th} power residue symbol of x is

$$\left(\frac{x}{p}\right)_r = x^{(p-1)/r}\,.$$

Computing r^{th} power residue symbols in the MPC setting can be done at essentially the same cost as computing Legendre symbols with the advantage that $\log r$ bit outputs are produced instead. Therefore, this modification has the potential to significantly increase the throughput of the Legendre PRF at essentially no cost – keeping in mind that r should not be too large, since the corresponding power residue PRF might lose its security. Generalizing the Legendre function and the Legendre PRF to higher power residue symbols, one obtains the following definitions.

Definition 10.4 (r^{th} power residue function)**.** Let p be a prime congruent to one modulo r and g a generator of $\mathbb{F}_p^\times$. The r^{th} power residue function $l^{(r)} : \mathbb{F}_p \to \mathbb{Z}/r\mathbb{Z}$ is defined as

$$l^{(r)}(a) = \begin{cases} k & \text{if } a \neq 0 \text{ and } a/g^k \text{ is an } r^{\text{th}} \text{ power modulo } p \\ 0 & \text{if } a = 0\,. \end{cases},$$

Definition 10.5 (r^{th} power residue PRF)**.** Let p be a prime congruent to one modulo r. The r^{th} power residue PRF over $\mathbb{F}_p$ is a family of functions $L_k^{(r)} : \mathbb{F}_p \to \mathbb{Z}/r\mathbb{Z}$ such that for each k in $\mathbb{F}_p$,

$$L_k^{(r)}(x) = l^{(r)}(k+x)\,.$$

This section provides the first security analysis of the power residue PRF. The attacks described in Section 10.4.4 and Section 10.4.5 do not use any properties of the Legendre symbol other than its multiplicativity. Therefore, they generalize to any multiplicative function with a hidden shift, including the r^{th} power residue function. The resulting attack requires $\mathcal{O}(p \log^2 p/(M^2 \log^2 r))$

power residue symbol evaluations and $\mathcal{O}(M^2 \log r)$ memory, because it suffices to consider L-sequences of length $\Theta(\log p / \log r)$. However, for large values of r, a better attack is explained below.

The attack is similar to the table-based collision search from Section 10.4.3. A speed-up by a factor r is obtained by querying the PRF at more carefully chosen arithmetic L-sequences. Let G be the subgroup of $\mathbb{F}_p^\times$ containing all $(p-1)/r^{\text{th}}$ roots of unity. If g is a generator of $\mathbb{F}_p^\times$, then the group G is generated by g^r.

If $L_k^{(r)}(0) = s$, then k/g^s is an r^{th} power modulo p. That is, $k \in g^s\, G$. This leads to the following procedure, with $m = \lceil \log p / \log r \rceil$ and $M < p/r$:

1. For M/m distinct a in G, store each pair $(L_k^{(r)}(a[m]), a)$ in a table $\mathcal{T}$.
2. Sample x uniformly at random from $g^s G$ until $(L_0^{(r)}(x + [m]), a) \in \mathcal{T}$ for some a. For each such collision, a candidate key is obtained as $k' = xa$. By a variant of Assumption 10.1, the number of candidates is $\mathcal{O}(1)$.

The first step of the above attack uses $M = m\,(M/m)$ queries to $L_k^{(r)}$ and uses $\mathcal{O}(M \log r)$ memory to store the table $\mathcal{T}$. On average, $|G|/(M/m) = \mathcal{O}(pm/(Mr))$ iterations of the second step are sufficient to find a candidate key. Since each iteration requires m power residue symbol computations to evaluate $L_0^{(r)}(x + [m])$, it follows that the total time complexity of the attack consists of $\mathcal{O}(M)$ storage operations and $\mathcal{O}(pm^2/(Mr)) = \mathcal{O}(p \log^2 p/(Mr \log^2 r))$ power residue symbol evaluations.

10.4.9 Implementation results

Using the attack from Section 10.4.4, three out of the six challenges proposed by the Ethereum foundation [141] were solved – including the test instance with a 40-bit prime. The implementation of the attack is available online[7]. A summary of the challenge parameters and the time and memory requirements of the attack is given in Table 10.4.

The implementation is written in C++ and was compiled with Clang 6.0.0. The attacks were executed on a Dell C6420 server with two Intel Xeon Gold 6132 CPUs clocked at 2.6 GHz (28 cores) and 128 GB of RAM. The optimizations described in Section 10.4.4 allow to significantly reduce the required memory and the number of evaluations of the Legendre symbol. As a result, the table lookups are the bottleneck in the implementation. Further details about the implementation can be found in the ToSC paper [36].

[7] `https://github.com/cryptolu/LegendrePRF`

Table 10.4: Parameters of the concrete challenges [141]. For all instances, the first $M = 2^{20}$ consecutive PRF outputs are given.

p	Security‡ *bits*	Time *core-hours*	Memory *GB/thread*	Key
$2^{40} - 87$	20	< 0.001	< 1	`4e2dea1f3c`
$2^{64} - 59$	44	1.5	3	`90644c931a3fba5`
$2^{74} - 35$	54	1500	3	`384f17db02976dcf63d`
$2^{84} - 35$	64	2^{21}†	3	
$2^{100} - 15$	80	2^{37}†	3	
$2^{148} - 167$	128	2^{65}†	3	

† Estimate, see the ToSC paper [36] for more details.
‡ Expected security level (conservative estimate) prior to this work.

11

Side-channel countermeasures

Masking is one of the most common countermeasures against side-channel attacks. This chapter shows that linear cryptanalysis can be used to evaluate the security of masked cryptographic primitives. The new techniques make it possible to obtain concrete security bounds in a variant of the probing model that allows the adversary to make only a bounded, but possibly large, number of measurements.

The contents of this chapter are based on the paper "Cryptanalysis of masked ciphers: a not so random idea" [53] from Asiacrypt 2020 (joint work with Siemen Dhooghe and Zhenda Zhang). Siemen Dhooghe and myself contributed equally to this work. The results of this paper were subsequently used to design several masked implementations in the papers "A low-randomness second-order masked AES" [52] from SAC 2021 (joint work with Siemen Dhooghe, Adrián Ranea and Danilo Šijačić) and "Cryptanalysis of efficient masked ciphers: applications to low latency" [51] from TCHES 2022 (joint work with Siemen Dhooghe, Amir Moradi and Aein Rezaei Shahmirzadi).

11.1 Introduction

Side-channel attacks such as differential power analysis [187] are an important concern for the security of implementations of cryptographic primitives in hardware and software. Accordingly, several adversarial models and side-channel countermeasures have been developed during the past two decades. Many of these countermeasures attempt to achieve security in the probing model of Ishai, Sahai and Wagner [167], or slight variants thereof.

A common theme among different countermeasures is that they rely on splitting all secret variables in the circuit into $d + 1$ or more random shares. As demonstrated by Ishai *et al.* [167], this approach can be used to achieve probing security against adversaries who can observe the values of up to d wires in the circuit. However, the probing security model is not quite sufficient for hardware-oriented countermeasures. Indeed, glitches may allow the adversary to obtain more than one wire value from a single probe. To counter this,

Nikova, Rechberger, and Rijmen [223] introduced the threshold implementation approach. From a formal point of view, the security of hardware-oriented countermeasures should be analyzed in a glitch-extended or *robust probing model* as formalized by Faust *et al.* [138] and it can be shown that threshold implementations achieve such first-order robust probing security [116].

Unsurprisingly, achieving probing security often comes at a cost with respect to area usage, latency, energy consumption, and so on. This chapter is primarily concerned with another important cost factor, namely the reliance of many countermeasures on the availability of a large number of random bits. Creating these bits can be quite expensive, especially since their generation should also be gray-box secure. In this regard, first-order threshold implementations provide an efficient countermeasure. In particular, if one ensures that each circuit layer satisfies the so-called *uniformity property*, glitch-extended first-order probing security can be achieved without using any randomness beyond what is necessary to share the state. If instead good randomness is readily available, threshold implementations allow trading this off for reduced area [62]. At Asiacrypt 2014, Bilgin *et al.* [61] proposed a higher-order variant of threshold implementations. However, Reparaz [239] later demonstrated that it succumbs to multivariate attacks. In further work at Crypto 2015, Reparaz *et al.* [240] propose to use remasking with fresh randomness to address this issue. However, as pointed out by Moos *et al.* [218], this and other schemes still lack a formal security analysis in the robust probing model.

As proposed by Faust *et al.* [138], an alternative approach is to design maskings based on a robust variant of the strong non-interference framework of Barthe *et al.* [23]. This has the benefit of allowing formal security proofs, which rely on establishing the composability of different gadgets in the shared circuit. However, ensuring composability unfortunately comes at an inherent randomness cost. Amortizing this cost is possible to some extent, but remains nontrivial – see for instance the work of Faust, Paglialonga, and Schneider [139] in the context of software-oriented masking. In addition, as for example pointed out by De Meyer, Wegener, and Moradi [113], it is often desirable to mask Boolean functions directly as opposed to falling back to a gate-level approach. Although verifying larger gadgets directly is possible within the strong non-interference framework, it requires nontrivial tools such as *maskVerif* due to Barthe *et al.* [22]. Of course, this does not directly address how to design efficient sharings. Also, one might hope to quantify to what extent verification fails; in the words of Barthe *et al.*: "It would be interesting to extend our work beyond purely qualitative security definitions, and to consider quantitative definitions that upper bound how much leakage reveals about secrets – using total variation distance or more recent metrics that directly or indirectly relate to noisy leakage security" [22, §7].

This chapter overcomes the composability problem for second-order threshold

implementations without relying on fresh randomness. As a result, second-order probing secure masked ciphers that require no or almost no randomness beyond what is necessary to share the input are obtained. In order to achieve these results, a variant of the probing model in which the adversary can make only a bounded number of queries is introduced. The approach in this chapter is based on a completely formal reduction from this model to the security of the masked cipher against linear cryptanalysis and leads to concrete upper bounds on the advantage (*i.e.* total variation distance) of such bounded-query adversaries.

From a practical point of view, the proposed methods provide a means to reason about and to correct potential flaws in the higher-order threshold implementations of Bilgin *et al.* [61]. Importantly, the additional requirements imposed by the analysis are relatively easy to satisfy when the underlying cipher has been designed with linear cryptanalysis in mind. As a result, one can benefit from the desirable properties of first-order threshold implementations – in particular their low randomness requirements – while simultaneously maintaining demonstrable security in the second-order probing model with glitches.

From a theoretical point of view, this chapter introduces a radically different approach to the security-evaluation of masked ciphers. Rather than attempting to show perfect probing security against adversaries making an arbitrary number of queries, a limited amount of leakage is allowed but it is show that it can not be exploited unless the adversary makes an infeasibly large number of measurements. In this approach, the concrete security bound of a masked cipher directly depends on the maximum absolute correlation of certain linear approximations over parts of the design. To estimate correlation upper bounds, standard techniques from linear cryptanalysis can be used. In particular, one can use the dominant trail approximation. Although the latter is only a heuristic, it is an integral part of the security argument of essentially all modern symmetric-key primitives and results in meaningful estimates if properly used. In a sense, the dominant trail approximation acts as a substitute for the strong composability requirements that are typically imposed. An important advantage of this approach is that it provides additional insight into the design of masked ciphers, and allows for a quantifiable trade-off between performance and security. In addition, one can benefit from the literature on linear cryptanalysis.

After introducing a number of preliminaries in Section 11.2, a bounded-query variant of the glitch-extended probing model is formalized in Section 11.3. A noisy extension of this model, developed jointly with Siemen Dhooghe, Amir Moradi and Aein Rezaei Shahmirzadi in the TCHES paper [51], is also briefly discussed. The reduction to linear cryptanalysis is spread over Sections 11.4 and 11.5. To limit the scope, only second-order probing adversaries are considered.

Section 11.6 presents a high-level overview of the properties the masked cipher needs to satisfy and the cryptanalytic process that should be followed to obtain concrete security bounds. Roughly speaking, for probes that are separated by a small number of rounds of the cipher, zero-correlation linear approximations can be exploited. If the adversary places its probes further apart, the analysis relies on upper bounds for the absolute correlation of linear approximations.

In Section 11.7, the framework developed in Sections 11.4 to 11.6 is illustrated by the design and analysis of a second-order masking of the block cipher LED [156]. The implementation requires a total 664 bits of randomness, *i.e.* 24 bits more than what is needed to share the plaintext and key, but no serious attempt was made to optimize this number. The choice for LED is mainly motivated by didactical reasons: LED is a classical wide-trail design with 4-bit S-boxes, which results in a very transparent security analysis. The same technique is used in [52] to build a low-randomness masked implementation of AES and in [51] to construct low-latency implementations of LED, Midori [18], Skinny [29] and PRINCE [76]. These results are not included in this thesis, but the main outcomes are briefly discussed in Section 11.8.

11.2 Masking and threshold implementations

This section introduces the masking countermeasure. In a masked implementation of a cryptographic primitive, every secret variable is split into two or more randomized shares. This is conceptually the same idea as secret sharing, but the connection to this area is limited in practice because the main difficulty is not the sharing scheme itself but the modification of the implementation to operate on shares in a correct and secure manner.

Section 11.2.1 describes Boolean masking, which is the most commonly used sharing scheme and the one that will be used throughout this chapter. Section 11.2.2 introduces threshold implementations, which are a popular method to transform a given circuit into one that operates on shares.

11.2.1 Boolean masking

Boolean masking was independently introduced by Goubin and Patarin [147] and Chari *et al.* [88]. It serves as a sound and widely-deployed countermeasure against side-channel attacks. The technique is based on splitting each secret variable x in the circuit into shares $\bar{x} = (x^1, x^2, \ldots, x^{s_x})$ such that $x = \sum_{i=1}^{s_x} x^i$ over a finite field k. If $k = \mathbb{F}_2$, then this masking approach is referred to as

Boolean masking. A random Boolean masking of a fixed secret is called uniform if all sharings of that secret are equally likely.

11.2.2 Threshold implementations

There are many ways to modify a given circuit in order to ensure that it operates on shared inputs and intermediates. For example, this can be done at the level of individual gates, or at a higher level involving generic Boolean functions. However, care must be taken to ensure that the sharing of the circuit is not only correct but also secure. This is especially challenging in hardware implementations due to the presence of glitches. Nikova *et al.* [223] introduced threshold implementations as a particular approach to share circuits. This approach achieves first-order glitch-extended probing security in the sense defined in Section 11.3 below. Later Bilgin *et al.* [61] generalized the threshold implementation approach in order to achieve higher-order univariate security. In the following, the main properties of threshold implementations are reviewed.

A threshold implementation consists of several layers of Boolean functions, as shown in Figure 11.1. Like for every masked implementation, a black-box encoder function generates a uniform random sharing of the input before it enters the shared circuit and the output shares are recombined by a decoder function. At the end of each layer, synchronization is ensured by means of registers.

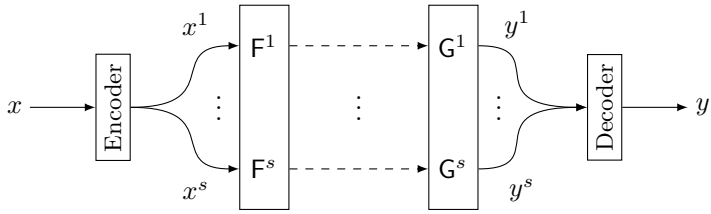


Figure 11.1: Schematic illustration of a threshold implementation with an equal number of input and output shares.

Let $\bar{\mathsf{F}}$ be a layer in the threshold implementation corresponding to a part of the circuit $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^m$. For example, F might be the linear layer of a block cipher. The function $\bar{\mathsf{F}} : \mathbb{F}_2^{ns_x} \to \mathbb{F}_2^{ms_y}$, where we assume s_x shares per input bit and s_y shares per output bit, is called a *sharing* of F. Sharings can have a number of properties that are relevant in the security argument for a threshold implementation; these properties are summarized in Definition 11.1.

Definition 11.1 (Properties of sharings [61, 223])**.** Let $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a function and $\bar{\mathsf{F}} : \mathbb{F}_2^{ns_x} \to \mathbb{F}_2^{ms_y}$ a sharing of F. The sharing $\bar{\mathsf{F}}$ is said to be

1. *correct* if $\sum_{i=1}^{s_y} \mathsf{F}^i(x^1, \ldots, x^{s_x}) = \mathsf{F}(x)$ for all x in $\mathbb{F}_2^n$ and for all shares $x^1, \ldots, x^{s_x}$ in $\mathbb{F}_2^n$ such that $\sum_{i=1}^{s_x} x^i = x$,
2. *d^{th}-order non-complete* if any function in d or fewer component functions $\bar{\mathsf{F}}_i$ depends on at most $s_x - 1$ input shares,
3. *uniform* if $\bar{\mathsf{F}}$ maps the uniform random sharing of every x in $\mathbb{F}_2^n$ to a uniform random sharing of $\mathsf{F}(x)$.

The correctness property from Definition 11.1 is an absolute minimum requirement to obtain a meaningful implementation. Furthermore, if all layers of a threshold implementation are first-order non-complete and uniform, then the resulting shared circuit can be proven secure in the first-order probing model considering glitches [116].

In the higher-order setting, the situation is more complicated. Using higher-order non-completeness and uniformity, one can secure a threshold implementation against higher-order univariate attacks. Univariate attacks do not combine information from multiple layers of a threshold implementation, contrary to multivariate attacks. However, perfect multivariate security can not be guaranteed using uniform sharings alone [239]. Instead, the threshold implementation approach was generalized to use fresh randomness [240]. However, even this last work has been shown to exhibit flaws against higher-order attacks [218].

In Section 11.3, a variant of the probing model – which will be called the *bounded-query probing model* – is introduced. In the main body of this chapter, it will then be shown that the issues surrounding higher-order threshold implementations can be overcome if the bounded-query probing model is adopted.

11.3 Bounded-query probing model

Section 11.3.1 introduces a variant of the threshold probing model of Ishai *et al.* [167] in which the adversary can make only a bounded number of queries. In addition, Section 11.3.2 discusses a further extension of this model in order to account for the effect of glitches. An extension that takes into account measurement noise is introduced in Section 11.3.3.

11.3.1 Threshold probing

Let $\ell \geq t$ be positive integers. A *t-threshold-probing adversary on* $\mathbb{F}_2^\ell$ is an algorithm $\mathcal{A}$ that interacts as follows with an oracle that holds an arbitrary sequence $(x_1, \ldots, x_\ell)$ in $\mathbb{F}_2^\ell$:

1. $\mathcal{A}$ specifies a set $\mathcal{I} = \{i_1, \ldots, i_{|\mathcal{I}|}\} \subset \{1, \ldots, \ell\}$ of cardinality at most t,
2. $\mathcal{A}$ then receives $(x_{i_1}, \ldots, x_{i_{|\mathcal{I}|}})$.

Note in particular that the adversary $\mathcal{A}$ is computationally unbounded, and must specify the location of the probes before querying the oracle. However, the adversary can change the location of the probes over multiple queries.

Ishai *et al.* [167] define a randomized stateless circuit C to be t-probing secure if it can be simulated from scratch such that no t-threshold probing adversary can distinguish $\mathsf{Dec} \circ C \circ \mathsf{Enc}$ from the simulation. Importantly, the adversary's interaction with the circuit or simulator is mediated through the encoder and decoder algorithms Enc and Dec, neither of which can be probed.

In this chapter, the security of a circuit C with input k against a t-threshold-probing adversary will be quantified by means of a left-or-right security game as depicted in Figure 11.2. The challenger picks a random bit b and provides the oracle $\mathcal{O}^b$, to which adversary $\mathcal{A}$ is given query access. The adversary queries the oracle by choosing up to t wires to probe, denoted by $\mathcal{P}$ in Figure 11.2, and sends it to the oracle along with the inputs k_0 and k_1. Note that the input of the circuit consists of both the plaintext and the key. The oracle responds by giving back the probed wire values of $C(k_b)$. After a total of q queries, the adversary responds to the challenger with a guess for b. Denote the result of the adversary after interacting with the oracle $\mathcal{O}^b$ using q queries by $\mathcal{A}^{\mathcal{O}^b}$. For left-or-right security, the advantage of the adversary $\mathcal{A}$ is then

$$\mathrm{Adv}_{t\text{-}\mathsf{thr}}(\mathcal{A}) = \left| \Pr[\mathcal{A}^{\mathcal{O}^0} = 1] - \Pr[\mathcal{A}^{\mathcal{O}^1} = 1] \right|.$$

This security notion will be referred to as the *bounded-query probing model*.

If an arbitrary number of queries is allowed, the above security definition is equivalent to the simulation-based definition of Ishai *et al.* [167] for stateless circuits. Indeed, if the simulator simply evaluates the circuit for an arbitrary choice of the secret inputs, then no adversary can distinguish the simulation from the real circuit with advantage higher than $\mathrm{Adv}_{t\text{-}\mathsf{thr}}(\mathcal{A})$. The left-or-right formulation leads to a slightly more direct proof of Theorem 11.1 in Section 11.4. However, note that there exist stronger notions of security such as the *strong non-interference model* of Barthe *et al.* [23]. In the latter model, the adversary

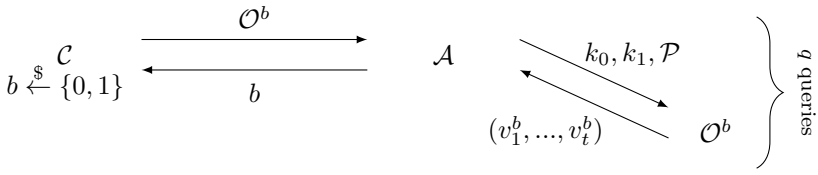


Figure 11.2: The privacy model for t-threshold-probing security consisting of a challenger $\mathcal{C}$, an adversary $\mathcal{A}$, a left-right oracle $\mathcal{O}^b$, two inputs k_0, k_1, a set of probes $\mathcal{P}$, and a set of probed wire values $(v_1^b, ..., v_t^b)$ of the circuit $C(k_b)$.

controls not only the unshared input of the circuit but also some of its shares. This is useful since probing security does not necessarily allow composition, as illustrated by Coron *et al.* [98]. As the approach developed in Sections 11.4 and 11.5 considers the circuit in its entirety, security under composition need not be considered. In fact, since the results in this chapter lead to secure sharings that do not use any randomness beyond what is necessary to encode the circuit input, it is clear that arbitrary composability cannot be achieved.

11.3.2 Glitches

It has been shown that hardware glitches can result in significant leakage that is not accounted for by the probing model, see for example the attacks of Mangard *et al.* on several masked AES implementations [213]. Consequently, it is necessary to extend the capabilities of threshold probing adversaries in order to capture the physical effect of glitches on a hardware platform. This chapter takes a conservative approach to the modeling of glitches by bundling groups of wires over which a glitch could carry information from one wire to another. Whereas one of the adversary's probes normally results in the value of a single wire, a glitch-extended probe allows obtaining the values of all wires in a bundle. This extension of the probing model has been discussed in the work of Reparaz *et al.* [240] and formalized by Faust *et al.* [138]. The formulation of the latter work is as follows: "For any ϵ-input circuit gadget G, combinatorial recombinations (aka glitches) can be modeled with specifically ϵ-extended probes so that probing any output of the function allows the adversary to observe all its ϵ inputs."

In the setting of threshold implementations, the above extension can be simplified. Recall that each layer of a threshold implementation consists of Boolean functions $\bar{\mathsf{F}}_i$, for which the synchronization of the inputs is ensured by

means of registers. Thus, a glitch-extended probe placed in the circuit for $\bar{\mathsf{F}}_i$ yields at most all of the input bits on which $\bar{\mathsf{F}}_i$ depends – but no more, since the layers of a threshold implementation are separated by registers.

Note that, apart from the glitch extension of the probing model, other effects such as transition leakage can be considered. More information on other leakage effects can be found in the work of Faust *et al.* [138]. The scope of this chapter is limited to the modeling of the effects that are traditionally taken into account in threshold implementations, thus only hardware implementations in the presence of glitches are considered.

11.3.3 Measurement noise

In the TCHES paper [51] (joint work with Siemen Dhooghe, Amir Moradi and Aein Rezaei Shahmirzadi), an extension of the bounded-query threshold probing model from Section 11.3.1 is introduced. In the modified model, the adversary can probe the circuit but it obtains noisy rather than exact results. This model will not be used for the analysis in this chapter, but a brief summary of the main differences is given below.

The noisy probing model resembles the *noisy leakage model* first introduced by Chari *et al.* [88] and extended by Prouff and Rivain [236]. The main difference between the two models is in the information given to the adversary. In the noisy leakage model, the adversary is given a noisy function of all wire values in the circuit. In the noisy probing model, as in the (glitch-extended) probing model, an adversary can only probe the circuit locally. However, unlike in the probing model, the adversaries' probes reveal only a noisy leakage function of the wire values. That makes the model similar to that of Dziembowski *et al.* [131]. However, the models differ in the way noisy leakage functions are defined. In addition, as opposed to the model of Dziembowski *et al.*, the proposed model is purely information-theoretic, non-asymptotic, and limits the number of queries that can be made by the adversary.

Formally, the threshold probing model is adapted by changing the oracle. More specifically, the notion of a probe is extended to a *noisy probe*. Instead of giving back the exact values on the wire/or bundle, the noisy probe returns a *noisy leakage function* of the values. The formal definition of noisy leakage functions is somewhat technical, and the reader is referred to [51, §3.2] for details. The noisy probing model is depicted in Figure 11.3.

In practice, the noisy probing model relates to an attacker performing a t^{th}-order attack on *traces*. A trace is a time series of the power consumption of an implementation. The attacker only has a limited number of traces which relates

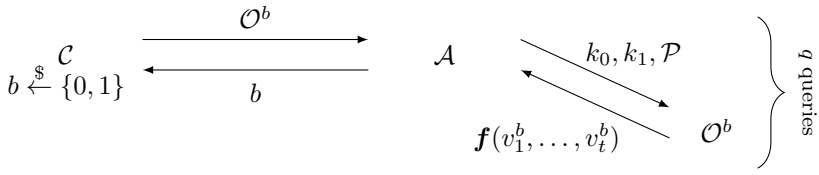


Figure 11.3: The privacy model for glitch-extended t-threshold-probing security consisting of a challenger $\mathcal{C}$, an adversary $\mathcal{A}$, a left-right oracle $\mathcal{O}^b$, two inputs k_0, k_1, a set of probes $\mathcal{P}$, and a noisy leakage function $\boldsymbol{f}(v_1^b, \ldots, v_t^b)$ of the probed wire values $v_1^b, \ldots, v_t^b$ in the circuit $C(k_b)$.

to a limited number of queries. The adversary can pick two secret values for the masked circuit which resembles a so-called *fixed vs. fixed* test.

11.4 Bound on the advantage

This section connects the bounded-query probing model from Section 11.3 to the cryptanalytic approach that will be developed in Sections 11.5 and 11.6. The link is established by means of Theorem 11.1 below, which provides an upper bound on the advantage of threshold probing adversaries in terms of the nontrivial Fourier coefficients of certain probability distributions associated with probed wire values. As a first step towards this result, the following lemma gives an upper bound on the entropy of a probability distribution in terms of its Fourier transformation. As in Chapters 6 to 8, the dual group of $\mathbb{F}_2^n$ is identified with $\mathbb{F}_2^n$.

Lemma 11.1. *Let $\boldsymbol{x}$ be a random variable on $\mathbb{F}_2^m$ with probability distribution $p_{\boldsymbol{x}}$ with Fourier transform $\widehat{p}_{\boldsymbol{x}}$. It holds that*

$$m - \mathsf{H}(\boldsymbol{x}) \leq \|\widehat{p}_{\boldsymbol{x}} - \delta_0\|_2^2 / \log 2\,,$$

with $\mathsf{H}(\boldsymbol{x})$ the Shannon entropy of $\boldsymbol{x}$ with respect to the binary logarithm.

Proof. By definition, the binary Shannon entropy of $\boldsymbol{x}$ is the quantity

$$\mathsf{H}(\boldsymbol{x}) = -\mathsf{E} \log_2 p_{\boldsymbol{x}}(\boldsymbol{x}) \leq m\,.$$

The goal is to upper bound the quantity $m - \mathsf{H}(\boldsymbol{x})$ in terms of the coordinates of the Fourier transformation of $p_{\boldsymbol{x}}$. By Jensen's inequality, it holds that

$$\mathsf{H}(\boldsymbol{x}) \geq -\log_2 \mathsf{E}\, p_{\boldsymbol{x}}(\boldsymbol{x}) = -\log_2 \|p_{\boldsymbol{x}}\|_2^2\,,$$

The right-hand side is equal to the Rényi entropy of $\boldsymbol{x}$. If $\widehat{p}_{\boldsymbol{x}}$ is the Fourier transformation of $p_{\boldsymbol{x}}$, then

$$\mathsf{H}(\boldsymbol{x}) \geq m - \log_2 \|\widehat{p}_{\boldsymbol{x}}\|_2^2.$$

Remark that $\widehat{p}_{\boldsymbol{x}}(0) = 1$, since $p_{\boldsymbol{x}}$ is a probability mass function. Isolating this coefficient, one obtains

$$m - \mathsf{H}(\boldsymbol{x}) \leq \log_2 \left(1 + \|\widehat{p}_{\boldsymbol{x}} - \delta_0\|_2^2\right) \leq \|\widehat{p}_{\boldsymbol{x}} - \delta_0\|_2^2 / \log 2\,. \qquad \square$$

Note that the inequality in Lemma 11.1 is rather sharp since $\|\widehat{p}_{\boldsymbol{x}} - \delta_0\|_2^2$ is small for the applications in this chapter. Furthermore, $\widehat{p}_{\boldsymbol{x}}$ typically has a small support, thereby enabling the use of Fourier-analytic methods.

Before turning to the proof of Theorem 11.1, it is useful to briefly consider the content of its statement. The theorem essentially shows that for a bounded-query probing secure circuit, all probed wire values either closely resemble uniform randomness or reveal nothing about the secret input. The usefulness of the result comes from the fact that it allows 'bad' probe values. These are values that might leak information about the secret, but which nevertheless cannot be distinguished from uniform random values unless a very large number of probing queries is made. In practice, the 'bad' values will be shares of the state resulting from probes placed far apart (*i.e.* separated by many rounds). The 'good' values then correspond to probes that are placed in nearby locations, such as within an S-box. As will be clarified in Sections 11.6 and 11.7, the 'good' values can also play an important role in the analysis of the key-schedule of a masked cipher.

Theorem 11.1. *Let $\mathcal{A}$ be a t-threshold-probing adversary for a circuit C. Assume that for every query made by $\mathcal{A}$ on the oracle $\mathcal{O}^b$, there exists a partitioning (depending only on the probe positions) of the resulting wire values into two random variables $\boldsymbol{x}$ ('good') and $\boldsymbol{y}$ ('bad') such that*

1. *The conditional probability distribution $p_{\boldsymbol{y}|\boldsymbol{x}}$ satisfies $\mathsf{E}_{\boldsymbol{x}} \|\widehat{p}_{\boldsymbol{y}|\boldsymbol{x}} - \delta_0\|_2^2 \leq \varepsilon$,*
2. *Any t-threshold-probing adversary for the same circuit C and making the same oracle queries as $\mathcal{A}$, but which only receives the 'good' wire values (*i.e. *corresponding to $\boldsymbol{x}$) for each query, has advantage zero.*

The advantage of $\mathcal{A}$ can be upper bounded as

$$\mathrm{Adv}_{t\text{-}\mathsf{thr}}(\mathcal{A}) \leq \sqrt{2q\,\varepsilon}\,,$$

where q is the number of queries to the oracle $\mathcal{O}^b$.

Proof. The first part of the proof consists of a standard game-hopping argument. Consider the following two additional games:

1. Game 't-thr-good' is a modification of the t-threshold probing game in which the oracle $\mathcal{O}^b$ replaces the 'bad' values in each query by uniform random values. In this game, $\mathcal{A}$ only receives information about 'good' wire values.
2. In the game 'Δ-bad', the adversary chooses a secret input k and is given access to an oracle with the same t-threshold-probing interface as $\mathcal{O}^b$. This oracle is either a t-threshold-probing oracle for the real circuit with input k, or a modification thereof in which the 'bad' values in each query are replaced by uniform random bits. The goal is to distinguish between these two cases.

We construct an adversary $\mathcal{B}$ for the game 'Δ-bad' by running $\mathcal{A}$. Specifically, $\mathcal{B}$ picks a uniform random bit b and forwards the corresponding secret k_b chosen by $\mathcal{A}$ to its challenger. Adversary $\mathcal{B}$ reports the oracle as real if and only if $\mathcal{A}$ correctly recovers b. Hence,

$$\mathrm{Adv}_{t\text{-thr}}(\mathcal{A}) \leq \mathrm{Adv}_{t\text{-thr-good}}(\mathcal{A}) + 2\mathrm{Adv}_{\Delta\text{-bad}}(\mathcal{B}).$$

The factor two in front of $\mathrm{Adv}_{\Delta\text{-bad}}(\mathcal{B})$ is due to our definition of 'advantage', *i.e.* the absolute difference between the winning and failure probabilities of $\mathcal{B}$. It is given that $\mathrm{Adv}_{t\text{-thr-good}}(\mathcal{A}) = 0$, so it suffices to upper bound $\mathrm{Adv}_{\Delta\text{-bad}}(\mathcal{B})$.

Since $\mathcal{B}$ makes exactly the same queries to its oracle as $\mathcal{A}$, the result of query i made by $\mathcal{B}$ can also be partitioned into 'good' and 'bad' wire values. Denote these values by $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ respectively when $\mathcal{B}$ is interacting with the real threshold probing oracle, and by $\boldsymbol{x}'_i$ and $\boldsymbol{y}'_i$ when $\mathcal{B}$ interacts with the (partially) randomized oracle.

Let $\delta_{\mathsf{TV}}(\cdot,\cdot)$ denote the total variation distance and $\bigotimes$ the tensor product. The distinguishing advantage of the adversary $\mathcal{B}$ is then upper bounded by

$$\begin{aligned}
\mathrm{Adv}_{\Delta\text{-bad}}(\mathcal{B}) &\leq \delta_{\mathsf{TV}}\big(\textstyle\bigotimes_{i=1}^{q} p_{\boldsymbol{x}_i,\boldsymbol{y}_i},\, \bigotimes_{i=1}^{q} p_{\boldsymbol{x}'_i,\boldsymbol{y}'_i}\big) \\
&\leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}\big(\textstyle\bigotimes_{i=1}^{q} p_{\boldsymbol{x}_i,\boldsymbol{y}_i} \,\|\, \bigotimes_{i=1}^{q} p_{\boldsymbol{x}'_i,\boldsymbol{y}'_i}\big)} \\
&\leq \sqrt{\frac{q}{2} \max_{1\leq i\leq q} D_{\mathrm{KL}}\big(p_{\boldsymbol{x}_i,\boldsymbol{y}_i} \| \, p_{\boldsymbol{x}'_i,\boldsymbol{y}'_i}\big)},
\end{aligned}$$

where D_{KL} denotes the Kullback-Leibler divergence and the second inequality is due to Pinsker. By definition of 'Δ-bad', the random variables $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$ have

the same probability distribution. Consequently,

$$D_{\mathrm{KL}}\big(p_{\boldsymbol{x}_i,\boldsymbol{y}_i} \,\|\, p_{\boldsymbol{x}'_i,\boldsymbol{y}'_i}\big) = \mathsf{E}_{\boldsymbol{t}}\, D_{\mathrm{KL}}\big(p_{\boldsymbol{y}_i|\boldsymbol{x}_i=\boldsymbol{t}} \,\|\, p_{\boldsymbol{y}'_i|\boldsymbol{x}'_i=\boldsymbol{t}}\big).$$

Finally, note that $\boldsymbol{y}'_i$ is uniformly distributed and independent of $\boldsymbol{x}_i$. If the number of bits of $\boldsymbol{y}_i$ is denoted by m_i, then

$$D_{\mathrm{KL}}\big(p_{\boldsymbol{y}_i|\boldsymbol{x}_i=t} \,\|\, p_{\boldsymbol{y}'_i|\boldsymbol{x}'_i=t}\big) = (m_i - \mathsf{H}(\boldsymbol{y}_i|\boldsymbol{x}_i)) \log 2 \leq \|\widehat{p}_{\boldsymbol{y}_i|\boldsymbol{x}_i} - \delta_0\|_2^2 \,.$$

The inequality above follows from Lemma 11.1. Since it is given that, for all i, $\mathsf{E}_{\boldsymbol{x}_i} \|\widehat{p}_{\boldsymbol{y}_i|\boldsymbol{x}_i} - \delta_0\|_2^2 \leq \varepsilon$, it follows that

$$\mathrm{Adv}_{\Delta\text{-}\mathsf{bad}}(\mathcal{B}) \leq \sqrt{\frac{q\,\varepsilon}{2}} \,.$$

Hence, one can conclude that

$$\mathrm{Adv}_{t\text{-}\mathsf{thr}}(\mathcal{A}) \leq 2\mathrm{Adv}_{\Delta\text{-}\mathsf{bad}}(\mathcal{B}) \leq \sqrt{2q\,\varepsilon} \,. \qquad \square$$

Theorem 11.1 can be extended to the noisy probing model with a similar but more technical proof. This result can be found in [51, Theorem 1].

11.5 Linear cryptanalysis of masked primitives

Theorem 11.1 provides an upper bound on the advantage of t-threshold probing adversaries in terms of the Fourier coefficients of the probability distribution of observed wire values. This section clarifies why it is beneficial to express the advantage upper bound in this particular form. Specifically, it will be shown that this reveals a strong link with the linear cryptanalysis of shared functions.

11.5.1 Restrictions of shared functions

Remark that all probability distributions referred to in Theorem 11.1 are with respect to a fixed value of the secret inputs. Consequently, it is clear that the relevant Fourier coefficients can not be directly related to the correlation matrix of the shared function itself. Instead, the relevant properties are those of restrictions of the shared function to sets of all valid sharings of a specific secret. Below, it is argued that these restrictions are indeed well-defined and that they come with a natural notion of linear cryptanalysis.

Recall from Section 11.2 that Boolean masking and threshold implementations are based on linear secret sharing. In general, any $\mathbb{F}_2$-linear secret sharing

scheme can be thought of as an algorithm that maps a secret x in $\mathbb{F}_2^n$ to a random element of a corresponding coset of a vector space $\mathbb{V} \subset \mathbb{F}_2^\ell$. The vector space $\mathbb{V}$ consists of all possible sharings of 0 in $\mathbb{F}_2^n$. Let $\rho : \mathbb{F}_2^n \to \mathbb{F}_2^\ell$ be a map that sends secrets to their corresponding coset representative.

Example 11.1. In Boolean masking, a secret x in $\mathbb{F}_2$ is shared as $(x^1, \ldots, x^\ell)$ where $x^1, \ldots, x^{\ell-1}$ are sampled uniformly at random and $x^\ell = x + \sum_{i=1}^{\ell-1} x^i$. In this case, $\mathbb{V}$ corresponds to the parity bit code

$$\mathbb{V} = \big\{(x^1, \ldots, x^\ell) \in \mathbb{F}_2^\ell \mid \textstyle\sum_{i=1}^{\ell} x^i = 0\big\}.$$

Furthermore, one possible choice of ρ is $\rho(x) = (x, 0, \ldots, 0)$. ▷

Let $\mathsf{F} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be any function. Recall from Definition 11.1 that a function $\bar{\mathsf{F}} : \mathbb{F}_2^\ell \to \mathbb{F}_2^\ell$ is said to be a correct sharing of F if, for all x in $\mathbb{F}_2^n$,

$$\bar{\mathsf{F}}(\rho(x) + \mathbb{V}) \subseteq \rho(\mathsf{F}(x)) + \mathbb{V}. \tag{11.1}$$

If $\bar{\mathsf{F}}$ is a uniform sharing, then the above inclusion is in fact an equality. For convenience, let $\mathbb{V}_a = a + \mathbb{V}$. Due to (11.1), the restriction of $\bar{\mathsf{F}}$ to $\mathbb{V}_a$ is a well-defined function $\mathbb{V}_a \to \mathbb{V}_b$ whenever $a = \rho(x)$ and $b = \rho(\mathsf{F}(x))$ for some x in $\mathbb{F}_2^n$. By slight abuse of notation, the same notation will be used for $\bar{\mathsf{F}}$ and for its restrictions.

Every random variable $\boldsymbol{x}$ on $\mathbb{V}_a$ has a corresponding probability mass function $p_{\boldsymbol{x}} : \mathbb{V}_a \to [0, 1]$. Since $\mathbb{V}$ is a group, the Fourier transformation $\widehat{p}_{a+\boldsymbol{x}}$ of $p_{a+\boldsymbol{x}}$ is well-defined (see Definition 3.5). The characters of $\mathbb{V}$ are the functions $x \mapsto (-1)^{u^\intercal x}$, for u in $\mathbb{F}_2^n/\mathbb{V}^\perp$. Throughout this chapter, $\widehat{\mathbb{V}}$ will be identified with $\mathbb{F}_2^n/\mathbb{V}^\perp$. Explicitly, the Fourier transformation of $p_{a+\boldsymbol{x}}$ at u in $\mathbb{F}_2^n/\mathbb{V}^\perp$ is

$$\widehat{p}_{a+\boldsymbol{x}}(u) = \sum_{x \in \mathbb{V}} (-1)^{u^\intercal x} p_{\boldsymbol{x}}(a + x)\,.$$

In addition, for any restriction $\bar{\mathsf{F}} : \mathbb{V}_a \to \mathbb{V}_b$, the correlation matrix of $x \mapsto \bar{\mathsf{F}}(a + x) + b$ is well defined by Definition 3.6. For convenience, we introduce the following definition. Note that it does not depend on the choice of the coset representatives a and b.

Definition 11.2. For $\mathbb{V} \subseteq \mathbb{F}_2^\ell$, let $\bar{\mathsf{F}} : \mathbb{V}_a \to \mathbb{V}_b$ be a well-defined restriction of a shared function. Let $\bar{\mathsf{F}}'(x) = \bar{\mathsf{F}}(x + a) + b$. The correlation matrix of $\bar{\mathsf{F}}$ is defined as the correlation matrix of $\bar{\mathsf{F}}'$.

11.5.2 Correlations between probed values

As shown in Section 11.4, the advantage of a probing adversary can be upper bounded in terms of $\|\widehat{p}_{\boldsymbol{z}} - \delta_0\|_2$ where $p_{\boldsymbol{z}}$ is the probability distribution of any

measured set of 'bad' wire values, possibly conditioned on several 'good' wire values. Note that the conditioning on 'good' values simply corresponds to fixing some variables in the circuit to constants before applying the results below. This section provides the link between $\widehat{p}_{\boldsymbol{z}}$ and the linear cryptanalysis of the shared circuit that will make it possible to upper bound the quantity $\|\widehat{p}_{\boldsymbol{z}} - \delta_0\|_2$ for a concrete masked cipher in Section 11.7.

For simplicity, from this point on, only second-order probing adversaries are considered. To obtain the desired link with linear cryptanalysis, it will be shown that the coordinates of $\widehat{p}_{\boldsymbol{z}}$ are entries of the correlation matrix of the state-transformation between the specified probe locations. This is illustrated in Figure 11.4.

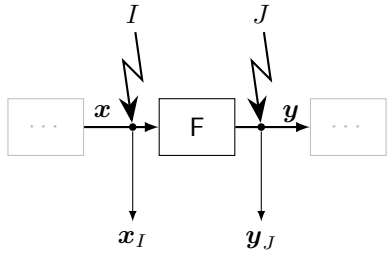


Figure 11.4: Two probes giving the observation $\boldsymbol{z} = (\boldsymbol{x}_I, \boldsymbol{y}_J)$.

The main result is stated in Theorem 11.2. To obtain it, the following property of correlation matrices will be used.

Lemma 11.2. *Let $\mathbb{V} \subset \mathbb{F}_2^\ell$ be a vector space and $\mathsf{L} : \mathbb{V} \to \mathbb{F}_2^m$ a linear map. If $\boldsymbol{x}$ is a random variable on $\mathbb{V}$ with probability distribution $p_{\boldsymbol{x}}$, then it holds that*

$$\widehat{p}_{\mathsf{L}(\boldsymbol{x})}(u) = \widehat{p}_{\boldsymbol{x}}(\mathsf{L}^{\mathsf{T}}(u))\,,$$

where $\mathsf{L}^{\mathsf{T}}(u) = \mathsf{L}^{\mathsf{T}}(u) + \mathbb{V}^\perp$ for notational convenience.

Proof. The result follows from $\widehat{p}_{\mathsf{L}(\boldsymbol{x})} = C^{\mathsf{L}}\widehat{p}_{\boldsymbol{x}}$ and Theorem 3.5 (2). □

For an index set $I = \{i_1, \ldots, i_m\}$, the restriction of x in $\mathbb{V}$ to I is denoted by $x_I = (x_{i_1}, \ldots, x_{i_m})$, which is in $\mathbb{F}_2^{|I|}$. Note that $x \mapsto x_I$ is a linear map.

Theorem 11.2. *Let $\mathsf{F} : \mathbb{V}_a \to \mathbb{V}_b$ be a function with $\mathbb{V} \subset \mathbb{F}_2^\ell$ and $I, J \subset \{1, \ldots, \ell\}$ sets. For $\boldsymbol{x}$ uniform random on $\mathbb{V}_a$ and $\boldsymbol{y} = \mathsf{F}(\boldsymbol{x})$, let $\boldsymbol{z} = (\boldsymbol{x}_I, \boldsymbol{y}_J)$. The Fourier transformation of the probability mass function of $\boldsymbol{z}$ then satisfies*

$$\left|\widehat{p}_{\boldsymbol{z}}(u, v)\right| = \left|C^{\mathsf{F}}_{\widetilde{v},\,\widetilde{u}}\right|\,,$$

where $\widetilde{u}$ and $\widetilde{v}$ in $\mathbb{F}_2^\ell/\mathbb{V}^\perp$ are such that $\widetilde{u}_I = u$, $\widetilde{u}_{[\ell]\setminus I} = 0$, $\widetilde{v}_J = v$ and $\widetilde{v}_{[\ell]\setminus J} = 0$.

Proof. Note that $(a + \boldsymbol{x}, b + \boldsymbol{y})$ is a well-defined random variable on $\mathbb{V}^2$. Let $\boldsymbol{z}' = (a_I, b_J) + \boldsymbol{z}$, then $\widehat{p}_{\boldsymbol{z}}(u, v) = (-1)^{u^\mathsf{T} a_I + v^\mathsf{T} b_J} \widehat{p}_{\boldsymbol{z}'}(u, v)$. Due to Lemma 11.2, the distribution of $\boldsymbol{z}'$ satisfies

$$\widehat{p}_{\boldsymbol{z}'}(u, v) = \widehat{p}_{(a+\boldsymbol{x}, b+\boldsymbol{y})}(\widetilde{u}, \widetilde{v}).$$

The probability distribution of $(a + \boldsymbol{x}, b + \boldsymbol{y})$ satisfies

$$p_{(a+\boldsymbol{x}, b+\boldsymbol{y})} = \big(I \otimes T^{\mathsf{F}'}\big) p_{(a+\boldsymbol{x}, a+\boldsymbol{x})},$$

where $\mathsf{F}'(x) = \mathsf{F}(x + a) + b$. Taking the Fourier transformation, one obtains

$$\widehat{p}_{(a+\boldsymbol{x}, b+\boldsymbol{y})} = \big(I \otimes C^{\mathsf{F}'}\big) \widehat{p}_{(a+\boldsymbol{x}, a+\boldsymbol{x})}.$$

By the definition of C^{F}, it holds that $C^{\mathsf{F}}_{\tilde{v}, \tilde{u}} = C^{\mathsf{F}'}_{\tilde{v}, \tilde{u}}$. Hence,

$$\begin{aligned}
\big|\widehat{p}_{\boldsymbol{z}}(u, v)\big| &= \left| \sum_{u', v' \in \mathbb{F}_2^\ell / \mathbb{V}^\perp} \delta_{\tilde{u}, u'} \, C^{\mathsf{F}}_{\tilde{v}, v'} \, \widehat{p}_{(a+\boldsymbol{x}, a+\boldsymbol{x})}(u', v') \right| \\
&= \left| \sum_{v' \in \mathbb{F}_2^\ell / \mathbb{V}^\perp} C^{\mathsf{F}}_{\tilde{v}, v'} \, \widehat{p}_{(a+\boldsymbol{x}, a+\boldsymbol{x})}(\widetilde{u}, v') \right| \\
&= \left| \sum_{v' \in \mathbb{F}_2^\ell / \mathbb{V}^\perp} C^{\mathsf{F}}_{\tilde{v}, v'} \, \widehat{p}_{a+\boldsymbol{x}}(\tilde{u} + v') \right| .
\end{aligned}$$

Since $p_{a+\boldsymbol{x}}$ is the uniform distribution on $\mathbb{V}$, it holds that $\widehat{p}_{a+\boldsymbol{x}} = \delta_0$. It follows that all terms except $v' = \widetilde{u}$ in the sum vanish, whence $|\widehat{p}_{\boldsymbol{z}}(u, v)| = |C^{\mathsf{F}}_{\tilde{v}, \tilde{u}}|$. □

Theorem 11.2 relates the linear approximations of F to $\widehat{p}_{\boldsymbol{z}}(u)$ and hence provides a method to upper bound $\|\widehat{p}_{\boldsymbol{z}} - \delta_0\|_2$ based on linear cryptanalysis. However, it should be noted that the result relates to linear cryptanalysis with respect to $\mathbb{V}$ rather than $\mathbb{F}_2^\ell$. The differences are mostly minor, but there is a subtle difference in relation to the important notion of 'activity'. In standard linear cryptanalysis, an S-box is said to be active if its output mask is nonzero. The same definition applies for linear cryptanalysis with respect to $\mathbb{V}$, but one must take into account that the mask is now an element of the quotient space $\mathbb{F}_2^\ell / \mathbb{V}^\perp$. In particular, if the mask corresponding to the shares of a particular bit can be represented by an all-one vector $(1, 1, \ldots, 1)$, it may be equivalently represented by the zero vector. It is still true that a valid linear approximation of a permutation must have either both input masks equivalent to zero or neither equivalent to zero. More generally, this condition is ensured by any uniform sharing.

Finally, note that Theorem 11.2 assumes that all intermediate states of the shared implementation are uniformly distributed on a coset of $\mathbb{V}$. This condition is guaranteed by the uniformity property of threshold implementations. In fact, it corresponds to the fact that the approximation with – up to equivalence – an all-zero input mask, must also have an all-zero output mask in order to have nonzero correlation. In particular, this is achieved if all shared functions are permutations. Accounting for a non-uniform distribution would require similar modifications to Theorem 11.2 as would be necessary to achieve higher than second-order security. In addition, if non-uniform sharings are used, the wide-trail argument [104] that will be used in later sections breaks down. For these reasons, the masking of LED in Section 11.7 relies on uniform sharings. A complete assessment of the consequences of non-uniformity on first and second order security is left as future work. Regarding this, it is worth noting that an analysis of the security degradation for non-uniform mappings has been made by Daemen [100] and has been tested in practice by Wegener *et al.* [284].

11.6 Cryptanalysis of masked ciphers

Theorems 11.1 and 11.2 provide the basic tools by which the security analysis of a masked cipher can be reduced to its linear cryptanalysis. This section provides a high-level overview of the analytic process. In addition, for each component of a typical masked cipher, the cryptanalytical properties that play a prominent role in the security analysis are discussed. This discussion can be useful not only to determine an appropriate masking of a cipher, but also as a factor in the design strategy of the cipher itself.

Our analysis of a masked cipher begins by partitioning the set of possible probe positions into three parts. This is closely related to the labeling of wire values as 'good' or 'bad' as required by Theorem 11.1. Each part corresponds to a different level of 'locality' and is analyzed by different methods. Specifically, the following cases can be distinguished:

S-box level. If both probes are placed within an S-box, perfect probing security is ensured so that such wire values can be labeled 'good' in the proof. Hence, the S-box must be shared such that it is higher-order probing secure. Based on this, one can verify the probing security of one round.

Nearby rounds. If the probes are separated by a small number of rounds, we rely on zero-correlation linear cryptanalysis. If the probe positions lead to zero-correlation approximations, then the probed values are uniformly distributed. In this case, from the point of view of Theorem 11.1, it does not matter if the values are marked as 'good' or 'bad'. Indeed, since the

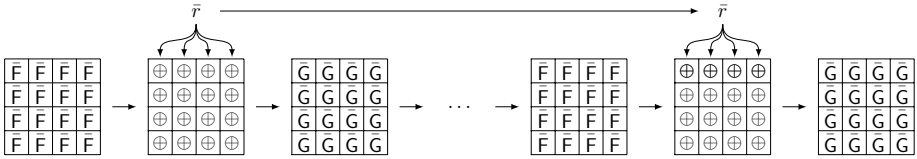


Figure 11.5: Addition of *static randomness* with S-box decomposition $\bar{\mathsf{S}} = \bar{\mathsf{G}} \circ \bar{\mathsf{F}}$.

distribution of the values is perfectly uniform in this case, one also has perfect probing security. This part of the analysis heavily depends on the linear layer of the cipher.

Distant rounds. When the probes are separated by many rounds, we rely on Theorem 11.2 and upper bound the absolute correlations of linear approximations. This is done using traditional techniques from linear cryptanalysis, in particular the dominant trail approximation. As discussed in more detail in Section 11.7.5, this is where the analysis leaves the realm of information-theoretical arguments and enters the domain of statistical cryptanalysis. Needless to say, all such wire values must be labeled as 'bad' from the point of view of Theorem 11.1.

For the key-schedule, the situation is slightly more complicated. If the key-schedule is sufficiently simple, as in the case of LED, one can label all key bits as 'good'. It then suffices – but is not necessary – to perform the analysis above for a fixed key. Several reasons for using this simplified approach are mentioned below. For more complicated key-schedules, a similar analysis as above for the key-schedule may be necessary.

A detailed example of the design of a secure sharing and its complete security evaluation is given in Section 11.7 for the block cipher LED. The remainder of this section briefly discusses how the analysis above translates to each of the components of a masked cipher.

S-box sharing and 'static' randomness. The S-box should be shared following the threshold implementation approach. For efficiency reasons, the S-box is often decomposed into several lower degree functions. The sharing of these functions should satisfy the uniformity property without using randomness, and be second-order non-complete. If the S-box is decomposed, the security of the composition must also be ensured. A simple way to achieve this is to add randomness between the decomposed functions. This randomness can be re-used in every S-box. This will be called *static randomness* as it is generated

by the black-box encoder and reused throughout the masked cipher. This is illustrated in Figure 11.5.

As discussed in Section 11.5.2, due to the uniformity of the shared S-box, the wide-trail strategy can be applied. In order to lower the potential advantage of the adversary, the sharing of the S-box is required to have strong nonlinear properties.

Linear layer. The linear layer of the cipher affects the security of the masked cipher for two reasons. The first is the diffusion between shares, resulting in zero-correlation trails. The second is that the layer ensures a minimum number of active S-boxes when probing distant rounds, resulting in correlation upper bounds.

Key schedule. Section 11.7 opts for simplicity by analyzing the key-schedule and state-transformation separately. This comes at a potential loss in the upper bounds, since many linear approximations will have correlation zero when averaged over some of the unknown key bits. Nevertheless, there are several good reasons for making such a simplification:

- It allows sticking as close as possible to the basic wide-trail approach. Indeed, conventional linear cryptanalysis of block ciphers does not usually consider the combined effect of the key-schedule and state-transformation.
- Although many trails have average correlation zero for a random sharing of the key, this can be quite difficult to analyze as it depends not only on which key bits the adversary can measure but also on the details of the key-schedule (the key-dependence of the sign of trail correlations can cancel out).
- No additional arguments are required for cryptographic permutations. In particular, the masked cipher can be used with a fixed key in order to obtain a secure implementation of a cryptographically strong permutation provided that the cipher allows for such usage.

11.7 Application to LED

This section applies the techniques developed in Sections 11.4 and 11.5 to the block cipher LED. This results in a masking requiring less than 700 bits of randomness while attaining second-order probing security.

11.7.1 Description of LED

LED is a 64-bit block cipher designed by Guo *et al.* [156]. The cipher's state is divided into 16 four-bit cells. The variant considered here has a 128-bit master key, from which subkeys are derived using a nibble-wise permutation. The cipher consists of 12 steps, each comprising four rounds. The step function is shown in Figure 11.6. Further details can be found in [156].

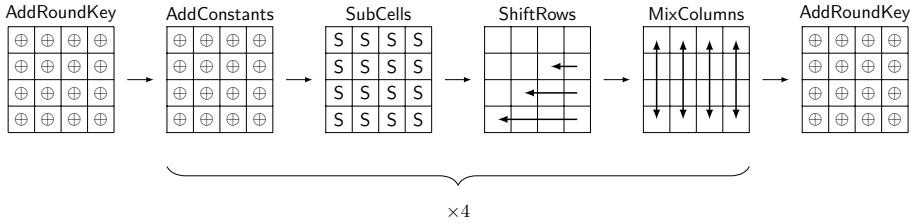


Figure 11.6: The step function of LED.

11.7.2 Sharing second-order LED

Following the principles outlined in Section 11.6, this section constructs a sharing of the LED cipher. Figure 11.7 gives an overview of the shared round function.

Masking state and key. The sharing of LED uses classical Boolean masking. The 64-bit state is shared using seven shares per bit, requiring 384 random bits. The 128-bit key is shared using three shares, which costs 256 random bits.

Sharing affine components. The masking of LED's linear components such as ShiftRows, MixColumns, and the key schedule are simply done share-wise. Constants are added to the first share of the concerning variable. The key addition is done by adding the key shares to the first three shares of the state.

Sharing the S-box. LED uses the PRESENT S-box. Following the decomposition given by Kutzner *et al.* [189], this S-box can be decomposed into two quadratic maps $\mathsf{S}_1 = \mathsf{G} \circ \mathsf{C}$ and $\mathsf{S}_2 = \mathsf{B} \circ \mathsf{G}$ where B and C are affine. Further details on this decomposition can be found in the Asiacrypt paper [53, Appendix A.1]. The sharing of the S-box is constructed from the sharing of G and has been verified to be uniform and second-order non-complete, the details can be

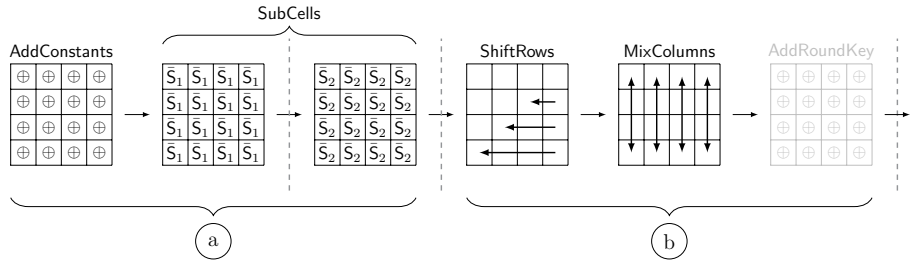


Figure 11.7: One round of masked LED. The locations of the registers are indicated by dashed lines. The round key addition is depicted in gray to show that it only happens every four rounds.

found in [53, Appendix A.2]. In between the two G functions, a layer of static randomness consisting of uniform zero sharings is added. This randomness is re-used in every S-box call and consists of 24 bits.

Before settling on the above choice, several other choices of the S-box sharing were considered. The randomness could be avoided altogether by using a second-order sharing of the entire S-box. However, as this would increase the number of shares, this option was not pursued. Alternatively, the S-box could be shared using fewer shares. For example, the work of Moradi *et al.* [219] constructs a uniform sharing using five input shares. Additionally, a uniform three-sharing is presented in [53, Appendix A.3]. However, both sharings achieve second-order probing security by first expanding their inputs and then re-compressing the cross products. Due to this expansion phase, there is an intermediate layer which is not uniform. As discussed in Section 11.5.2, the use of non-uniform functions would require more analysis.

The sharing of the S-box can also be adapted to improve its nonlinearity, leading to better security bounds. One such option based on composing with a nontrivial sharing of the identity function, is explored in [53, Appendix A.4].

Security. In Sections 11.7.3 to 11.7.6 below, the following concrete security claim will be established.

Security claim 11.1. For the masked LED described in this section, the following bound on the advantage of the adversary (assuming piling-up) in the probing model is claimed:

$$\mathrm{Adv}_{2\text{-thr}}(\mathcal{A}) \leq \sqrt{\frac{q}{2^{121}}}\,.$$

11.7.3 Probing security of one round

This section establishes the second-order probing security of one round of masked LED, such that all wire values corresponding to such probing queries can be labeled as 'good'. Recall that, since each layer of the masked cipher is uniformly shared, the input distribution to the round is uniform. To establish the probing-security claim, it suffices to consider all possible probe positions. If both probes are placed in the same layer, the claim follows directly from the second-order non-completeness of each function.

When both probes are placed in part (a) in Figure 11.7, the only nontrivial new case corresponds to placing one probe in $\bar{\mathsf{S}}_1$ and one in $\bar{\mathsf{S}}_2$. Due to the refreshing layer, the input to $\bar{\mathsf{S}}_2$ is uniformly random even if $\bar{\mathsf{S}}_1$ is probed. Since $\bar{\mathsf{S}}_2$ is second-order non-complete, placing the second probe in $\bar{\mathsf{S}}_2$ then reveals no information about the secret.

If one probe is placed in part (a) and another in part (b), then the second probe reveals at most a single share (the same) of each variable by the linearity of part (b). Due to a consistent choice of the covering scheme used for non-completeness, the previous arguments are not limited to the bit-level. Consequently, the analysis is the same as for the case with two probes in part (a).

Every four rounds, a round key is also added to the state. The effect of the key-schedule and key addition is discussed in Section 11.7.6.

11.7.4 Nearby rounds: zero correlation

This section shows that the distribution of any pair of measurements from probes which are at most three rounds apart almost always conforms to one of two cases: either the observations are uniformly distributed, or they do not reveal anything about the secret. To prove the uniformity claim, the analysis relies on techniques from zero-correlation linear cryptanalysis. The latter case, *i.e.* independence of the secret for possibly non-uniform observations, was discussed in the previous section. For these cases, the advantage of the adversary is zero as required by Theorem 11.1. All other cases will be considered in Section 11.7.5.

The argument consists of an analysis of all possible probe placements. As noted above, the analysis in this section is restricted to probes that are at most three rounds apart. This results in the following cases:

Rounds i and $i+1$. If the adversary probes in part (a) of round i, then the MDS matrix ensures that a full column of the state will be active at the input of round $i+1$. A measurement in part (a) of round $i+1$ can activate

shares from at most one cell of the state such that the corresponding approximations have correlation zero. Similarly, due to the ShiftRows operation, by probing in part (b) of round $i+1$, the adversary can never activate all cells of a single column at the input of round $i+1$. Hence, approximations with nonzero correlation can only be obtained by probing in part (b) of round i. However, in this case only a single share of each bit is learned, such that a second probe in part (a) or (b) of round $i+1$ reveals nothing about the secret by the same argument for the case where both probes are placed in round i.

Rounds i and $i+2$. If either part (a) or (b) of round i are probed, this results (up to symmetry) in one of the four activity patterns shown in Figure 11.8 for rounds $i+1$ and on. By probing anywhere in round $i+2$, the adversary can clearly activate at most four cells at the input of this round. In cases (1)–(3) in Figure 11.8, at least eight S-boxes are active at the input of round $i+2$ such that the correlation of such approximations is zero. In the remaining case, *i.e.* activity pattern (4), only a single column of the state is active at the input of round $i+2$. However, by probing in part (a) of round $i+2$, only a single cell can be activated. Probing part (b) allows activating four cells but never from the same column due to the shift rows step.

Rounds i and $i+3$. It is easy to see that activity patterns (2)-(4) in Figure 11.8 lead to correlation zero since at least eight S-boxes are then active at the input of round $i+3$. Indeed, if the second probe is placed anywhere in round $i+3$, at most four cells of the state can be activated. For pattern (1) in Figure 11.8, the correlation may be nonzero and will be bounded in Section 11.7.5.

The above case analysis shows that, when the probes are placed in nearby rounds, perfect security is obtained. The only remaining cases are probes in rounds i and $i+r$ for $r>4$ and the activity pattern (1) in Figure 11.8 when probes are placed in rounds i and $i+3$. These cases are analyzed in Section 11.7.5.

11.7.5 Five rounds or more: low correlation

As discussed in Section 11.7.4, if the probes are placed in rounds that are far apart, the observed values are usually not uniformly distributed. Nevertheless, it is possible to show that they will be nearly uniform in the sense that all nontrivial coordinates of the Fourier transformation of their probability distribution are small. To show this, the correlation of all linear trails whose activity pattern is compatible with the probe positions will be upper bounded.

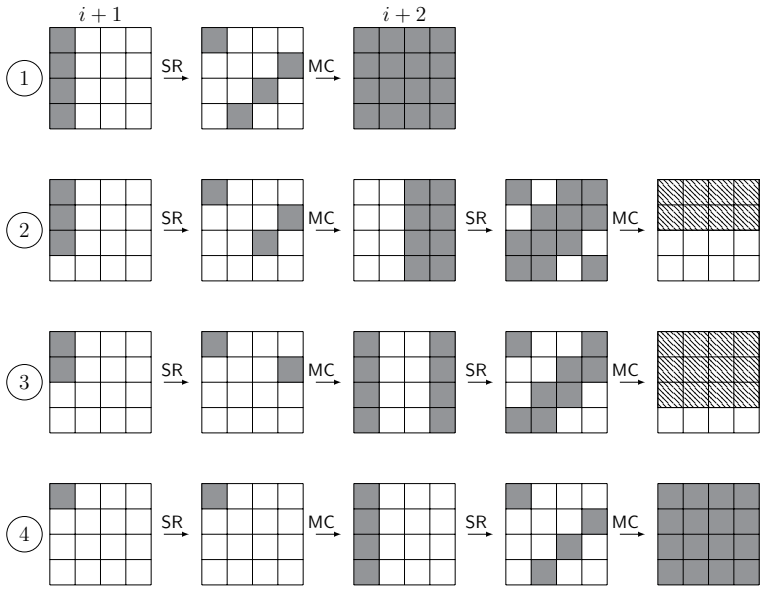


Figure 11.8: Activity patterns for masked LED, corresponding (up to symmetry) to the four possible patterns created by a probe placed in round i. SR is short for ShiftRows and MC for MixColumns. White cells are inactive, cells in gray are active, and hatched cells correspond to an example trail with a minimum number of active cells.

Remark 11.1. The analysis in this section relies on the piling-up principle, *i.e.* upper bounds on the correlations of the best individual trails will be used instead of upper bounds on the correlations of linear approximations. This is a arguably a reasonable starting point, given that every adversary that can distinguish the probed wire values from uniform randomness gives rise to a linear distinguisher. In fact, the security arguments for most symmetric-key primitives do not go further than such an analysis – although they should. As explained in Section 11.7.6, the correlation upper bounds need not hold for all key and refreshing variables but only in the average over the unobserved variables. Consequently, it is likely that the true values of the correlations are much lower than the estimates presented below. ▷

To upper bound absolute trail correlations, we rely on the standard wide-trail argument [104]. Specifically, the fact that any linear trail over four rounds of (shared) LED activates at least 25 S-boxes will be used. Additionally, an upper bound on the correlation of the best linear approximations over the shared S-box from Section 11.7.2 is required. Since the shared S-box is quite large, a

direct calculation of its nonlinearity is nontrivial. Instead, the following lemma for quadratic Boolean functions can be used. A slight restatement of this result can be found in the book by Carlet [86, Chapter 6].

Lemma 11.3 (Proposition 16 [86])**.** *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a quadratic Boolean function. Denote the rank of its symplectic form by r. That is, $r = \mathrm{rank}(S)$ where S in $\mathbb{F}_2^{n\times n}$ is the symmetric matrix such that $y^\mathsf{T} S\, x = f(x+y)+f(x)+f(y)$. Then*

$$\frac{1}{2^n}\left|\sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)}\right| \leq 2^{-r/2}.$$

Lemma 11.4. *Let $\bar{\mathsf{G}} : \mathbb{V}_a \to \mathbb{V}_b$ be any restriction of the sharing of G defined in Section 11.7.2. Denote its correlation matrix by $C^{\bar{\mathsf{G}}}$. For any u and v in $\mathbb{F}_2^\ell/\mathbb{V}^\perp$ such that $u_i^j \neq 0$ for some $i \neq 3$, it holds that $\left|C^{\bar{\mathsf{G}}}_{u,v}\right| \leq 2^{-3}$.*

Proof. Since $\bar{\mathsf{G}}$ is a function of 28 variables, bounding all of its correlations is nontrivial. However, one can use the fact that $\bar{\mathsf{G}}$ is a quadratic function. Indeed, if B in $\mathbb{F}_2^{\ell\times d}$ is a basis matrix for $\mathbb{V}$, then

$$\begin{aligned}\left|C^{\bar{\mathsf{G}}}_{u,v}\right| &\leq \max_{w\in\mathbb{F}_2^\ell/\mathbb{V}^\perp}\frac{1}{|\mathbb{V}|}\left|\sum_{x\in\mathbb{V}}(-1)^{u^\mathsf{T}\bar{\mathsf{G}}(x+a)+w^\mathsf{T}x}\right| \\ &\leq \max_{w\in\mathbb{F}_2^\ell/\mathbb{V}^\perp}\frac{1}{2^d}\left|\sum_{x\in\mathbb{F}_2^d}(-1)^{u^\mathsf{T}\bar{\mathsf{G}}(Bx+a)+w^\mathsf{T}Bx}\right|.\end{aligned}$$

Since $u^\mathsf{T}\bar{\mathsf{G}}(Bx+a)+w^\mathsf{T}Bx$ is a quadratic Boolean function, Lemma 11.3 is applicable. Let $S_{i,j}$ denote the symplectic form matrix of $\mathsf{G}_i^j(Bx+a)$. Since $S_{3,j}=0$ for $j=1,\ldots,7$, we must require that u_i^j is nonzero for some $i\neq 3$ to obtain a nonzero minimum rank. Specifically, it suffices to verify that for all nonzero u in $\mathbb{F}_2^\ell/\mathbb{V}^\perp$ with $u_3^j=0$ for $j=1,\ldots,7$,

$$\mathrm{rank}\left(\sum_{i=1}^{4}\sum_{j=1}^{7}u_i^j\,S_{i,j}\right)\geq 6.$$

Lower bounding the left-hand side above reduces to the MinRank problem. For our purposes, a brute force search over all representative choices of u is feasible. The verification code can be found online[1]. □

Theorem 11.3. *Let $\bar{\mathsf{S}} = \bar{\mathsf{S}}_2 \circ \bar{\mathsf{S}}_1 : \mathbb{V}_{a_1} \to \mathbb{V}_{a_3}$ be the sharing of $S = S_2 \circ S_1$ defined in Section 11.7.2. Denote the correlation matrix of $\bar{\mathsf{S}}_i : \mathbb{V}_{a_i} \to \mathbb{V}_{a_{i+1}}$ by $C^{\bar{\mathsf{S}}_i}$. For any u and v in $\mathbb{F}_2^\ell/\mathbb{V}^\perp$ not both equal to zero and for all w in $\mathbb{F}_2^\ell/\mathbb{V}^\perp$, it holds that $\left|C^{\bar{\mathsf{S}}_2}_{u,w}C^{\bar{\mathsf{S}}_1}_{w,v}\right| \leq 2^{-3}$.*

[1] `https://gitlab.esat.kuleuven.be/Zhenda.Zhang/LED_SHARING`

Proof. Since $\bar{\mathsf{S}}$ is affine equivalent to $\bar{\mathsf{G}} \circ \bar{\mathsf{G}}$, it suffices to analyze the latter function. By Lemma 11.4, it holds that $|C^{\bar{\mathsf{G}}}_{u,w}| \leq 2^{-3}$ unless $u^j_i = 0$ for $j = 1, \ldots, 7$ and for all $i \neq 3$. However, for such u, $|C^{\bar{\mathsf{G}}}_{u,w}| = 0$ whenever w also satisfies $w^j_i = 0$ for $j = 1, \ldots, 7$ and for all $i \neq 3$. Indeed, the i^{th}-share of the third bit G^i_3 does not depend on any shares from the third input variable. It follows that $|C^{\bar{\mathsf{G}}}_{u,w} C^{\bar{\mathsf{G}}}_{w,v}| \leq 2^{-3}$. □

Remark 11.2. Experimentally, the piling-up approximation was found to give the correct upper bound 2^{-3} for the maximum absolute correlation of the shared S-box. Due to resource constraints, the experiment was limited to the verification for one choice of static randomness. ▷

For probes placed in rounds i and $i + r$ with $r \geq 4$, the relevant linear trails all have at least 25 active S-boxes. This is a consequence of the wide-trail design strategy and can be derived in exactly the same way as for the AES [104]. Hence, by Theorem 11.3, the correlations of these trails are bounded by 2^{-75}. By Theorem 11.2, it then follows that the 2-norm of the nontrivial Fourier coefficients of the observed bits $\boldsymbol{z}$ can be upper bounded as

$$\|\widehat{p}_{\boldsymbol{z}} - \delta_0\|_2^2 \leq |\operatorname{supp} \widehat{p}_{\boldsymbol{z}}| \, \|\widehat{p}_{\boldsymbol{z}} - \delta_0\|_\infty^2 \leq 2^{22} \, 2^{-150} = 2^{-128} \,,$$

where the second step uses the inequality $|\operatorname{supp} \widehat{p}_{\boldsymbol{z}}| \leq 2^{22}$, which follows from the fact that the observed value $\boldsymbol{z}$ consists of at most 22 bits in the glitch-extended probing model: if an output coordinate of $\bar{\mathsf{G}}$ is read, at most 10 shares are learned; if an output of the shared linear layer is probed, at most 11 shares are observed. The latter number of shares is due to the fact that LED's MDS matrix has at least five zeros per row when represented over $\mathbb{F}_2$. Note that, in practice, the upper bound above is not likely to be tight, because it is unlikely that a glitch will reveal the exact value of all 11 bits in a single measurement.

The only remaining case is when the adversary probes in rounds i and $i + 3$, assuming the activity pattern in case (1) from Figure 11.8. In this case, only 24 S-boxes are active. Furthermore, we again have $|\operatorname{supp} \widehat{p}_{\boldsymbol{z}}| \leq 2^{22}$. Hence,

$$\|\widehat{p}_{\boldsymbol{z}} - \delta_0\|_2^2 \leq 2^{22} \, 2^{-144} = 2^{-122} \,.$$

A more careful analysis would result in slightly improved bounds. Nevertheless, since the bound is sufficiently small for all practical purposes, we avoid such an analysis and opt for simplicity instead.

11.7.6 Influence of the key-schedule

The arguments in Sections 11.7.4 and 11.7.5 establish the security of the proposed masked LED design against an adversary which does not look at shares

of the key or the bits which are added in the refreshing layer. Indeed, for such an adversary, all wire values for queries with probe positions considered in Section 11.7.4 are marked as 'good' and all others (considered in Section 11.7.5) as 'bad'. Theorem 11.1 then provides the desired security bound. However, showing security when all wires in the circuit can be probed requires a slightly more careful choice of 'good' and 'bad' wire values.

Fortunately, the LED key-schedule consists only of bit-permutations. Hence, its sharing is perfectly secure against second-order threshold-probing adversaries. The same holds for the random bits used in the refreshing layer. Hence, Theorem 11.1 can be applied with the following labeling of wire values:

Probes discussed in § 11.7.3–11.7.4. For all these probe positions, all wire values can be considered as 'good'. This includes any key bits (and additional randomness in the refreshing layer) that might be observed by the adversary. Indeed, even with glitch-extended probes, the adversary can observe at most two shares of each key bit.

Probes discussed in § 11.7.5. For these probe positions, all wire values corresponding to state shares should be marked as 'bad'; shares of the key (or additional randomness used in the refreshing layer) are labeled 'good'. The arguments in Section 11.7.5 then apply directly.

At least one probe in the key-schedule. In this case, all wire values may be considered 'good'. Indeed, recall that any non-complete subset of state bits at a particular layer is uniformly distributed and the adversary observes at most two shares of each key bit.

For the upper bound ε, the values derived in Section 11.7.5 may be used directly because the analysis of the trails there is valid for every choice of the key. Note that the latter assumption is stronger than necessary; it suffices to assume that the bounds derived in Section 11.7.5 are valid in the average over all unobserved randomness and key variables.

11.8 Application to other primitives

As discussed above, the security analysis and masking choice of LED can be adapted to several other primitives. In general, the same approach is often directly applicable to primitives following the wide-trail design strategy.

However, the LED masking presented in this chapter uses seven shares and a large number of register stages. This unfortunately leads to high latency and

large area, making it not competitive compared to the state of the art despite requiring almost no randomness beyond what is necessary to share the inputs. Nevertheless, it is possible to design more efficient maskings using the strategy outlined in this chapter.

In the SAC paper [52], a second-order secure masking of the AES is designed. It uses a total of 1800 random bits (900 when amortized over multiple calls), about an order of magnitude less than previous implementations at the time. The analysis is more complicated and relies on automated tools. However, its area-requirements are not yet competitive with previous work. This was joint work with Siemen Dhooghe, Adrán Ranea and Danilo Šijačić.

The TCHES paper [51] presents low-latency implementations of the block ciphers LED-128, Midori-64, Skinny-64 and PRINCE that do not require any fresh randomness. These implementations are competitive with the state-of-the-art in terms of area, latency and throughput. To achieve this, their security analysis is based on the noisy probing model that was briefly discussed in Section 11.3.3. This was joint work with Siemen Dhooghe, Amir Moradi and Aein Rezaei Shahmirzadi.

12

Backdoored ciphers

This chapter is concerned with block ciphers that have an intentional but hidden weakness or *backdoor*. An attack on the backdoored cipher LowMC-M is given, and its design strategy (the MALICIOUS framework) is analyzed. It is shown that 'trivial' instances of MALICIOUS can be constructed from any tweakable block cipher. In addition, a nontrivial backdoored variant of the AES is constructed. Finally, the backdoored block cipher Boomslang is introduced.

The contents of this chapter are based on the note "Cryptanalysis of the MALICIOUS framework" [54] (joint work with Chaoyun Li) and the paper "Constructing and deconstructing intentional weaknesses in symmetric ciphers" [25] from Crypto 2022 (joint work with Christof Beierle, Patrick Felke and Gregor Leander). From the latter work, only the results related to MALICIOUS are included in this chapter. I was the principal author of [54]. The authors of [25] contributed equally, with Malicious AES mainly due to Christof Beierle and Gregor Leander, and Boomslang mainly due to myself.

12.1 Introduction

The design of deliberate and often hidden weaknesses in cryptographic primitives has a long history. Among the most famous examples are the block cipher DES [237], for which the key length was deliberately reduced to 56 bits [170, page 232], and the pseudorandom generator Dual EC DRBG, which was equipped with a backdoor [35, 232]. More recently, it was discovered that the security of the widely deployed cipher GEA-1 was secretly weakened to 40 bits in order to fulfill European export restrictions [28]. The construction of the GEA-1 backdoor was reverse-engineered in the Crypto 2022 paper [25], but this analysis is not included in this chapter.

In the academic world, backdoored ciphers based on hiding strongly biased linear approximations have been proposed [235, 244]. Another approach is based on partitioning cryptanalysis [160], where the backdoor consists of a partition of the plaintext space that is preserved under the encryption function [19, 143, 231]. The latter approach is related to invariant subspace attacks [196] and nonlinear

invariant attacks [266]. In the case of hash functions, it was shown how to design malicious variants of SHA-1 with built-in collisions [4]. For all of these constructions, the designers either do not claim security of the backdoor in the sense that it cannot be recovered even if its general form is known, or there is an attack which recovers the backdoor from the specification of the cipher (see for example [286]).

At Crypto 2020, Peyrin and Wang [234] introduced the MALICIOUS framework to construct backdoored tweakable block ciphers. One of the interesting features of this framework is that the difficulty of recovering the backdoor relies on well-understood cryptographic principles. The basic idea is to construct a tweakable block cipher such that for a particular malicious tweak pair (t, t'), the cipher exhibits a differential with high probability that leads to a practical key-recovery attack. The tweak pair (t, t') is secured by being a pair of preimages for outputs of an extendable-output function H such as SHAKE [128]. The backdoor is undiscoverable in the sense that finding the hidden tweak pair requires finding a collision for H.

Concretely, Peyrin and Wang propose the tweakable block cipher LowMC-M by instantiating the MALICIOUS framework with the cipher LowMC [8]. One drawback of this construction is that the round function is based on a rather complex (randomly sampled) linear layer and a partial S-box layer. As suggested for future work in [234], it would be interesting to find similar constructions that are based on other cryptanalytic attacks rather than differential cryptanalysis, as this might lead to more natural instances.

Section 12.2 of this chapter shows that, although the embedded malicious tweak pair itself is hard to recover, it is feasible to find other weak tweak pairs that can be used to mount key-recovery attacks. This results in full-round key-recovery attacks on most instances of LowMC-M. Although the time complexity of these attacks is large for most cases, the attacks require only a small amount of data. Despite these findings, the MALICIOUS framework itself is not flawed and can be used to create backdoored variants of LowMC if the parameters are modified.

In Section 12.3, it is shown that *any* tweakable block cipher can be modified in a simple way to conform to the MALICIOUS framework. In a nutshell, the idea of constructing such an instance is to check if the tweak hashes to a certain constant and if so, return the key instead of the ciphertext. If the hash does not match, the cipher is executed unchanged. Although this example shows that the initial goals of the MALICIOUS framework can be achieved in a trivial way, it is an artificial construction and does not give further insight on how to construct *hidden* weaknesses. A malicious designer would rather aim to construct a 'natural' instance that follows modern symmetric-key design principles and for which a sound design rationale can be formulated.

Towards achieving this goal, two new instances of the MALICIOUS framework are proposed in Sections 12.4 and 12.5. Whereas LowMC-M relies on a hidden high-probability differential, the new constructions are based on invariants. It is argued that this allows for more natural instances. In particular, Section 12.4 shows how the round function of the AES can be used together with a modified key-schedule in order to embed a backdoor based on an invariant subspace over the round function. More precisely, the backdoor exploits an invariant subspace that was already known since 2004 [194]. In Section 12.5, a dedicated construction called Boomslang is proposed. It embeds a backdoor based on a nonlinear invariant over two consecutive round functions.

The new constructions constitute the first backdoored ciphers that follow modern design principles for general-purpose block ciphers, and are expected to achieve competitive performance. Compared to previous constructions not based on the MALICIOUS framework, in particular those of Paterson [231] and Posteuca and Ashur [235], the new proposals also directly improve upon the usability of the backdoor as they enable significantly more practical key-recovery attacks. Nevertheless, the constructions presented in this chapter and the MALICIOUS framework in general still have important limitations. These are discussed in Section 12.6.

12.2 Cryptanalysis of LowMC-M

The attack on LowMC-M is based on the observation that, apart from the malicious tweak pair, it is possible to find other pairs of tweaks that lead to differentials with probability one over a large number of rounds of the cipher. After a suitable weak tweak pair has been obtained, there are several ways to set up a key-recovery attack using the probability-one differential.

LowMC-M is specified in Section 12.2.1. Section 12.2.2 presents the main observation that leads to the attack. Finally, an example of a possible key-recovery procedure is given in Section 12.2.3.

12.2.1 Specification of LowMC-M

An overview of LowMC-M is shown in Figure 12.1. Throughout this chapter, n denotes the block size in bits, k is the number of key bits and r is the number of rounds. The round function of LowMC-M consists of three operations. In the first step, the round key, round constants and round tweak t_i are added to the state. The key-schedule is linear, but the details are not important for the analysis in this chapter. The round tweaks are derived using the extendable

output function H as $(t_1, t_2, \ldots, t_r) = \mathsf{H}(t)$. The next step consists of a partial S-box layer. The details of the S-box are not important, but LowMC-M uses the same 3-bit S-box as LowMC. The total number of bits operated on by the S-boxes will be denoted by s. Finally, an essentially random invertible linear map $\mathsf{L}_i : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is applied to the state.

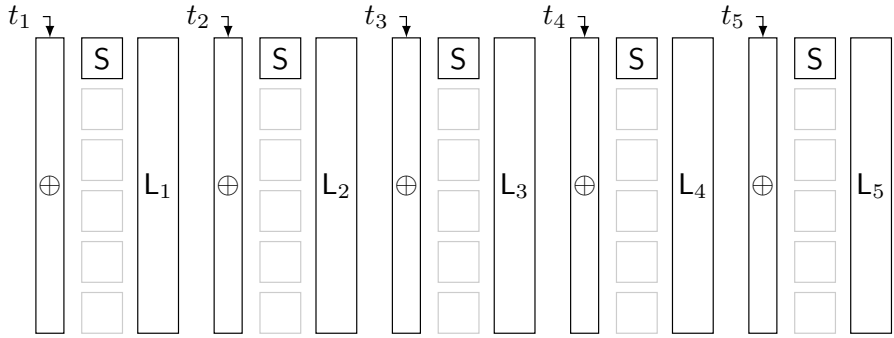


Figure 12.1: Five rounds of LowMC-M with $n = 18$ and $s = 3$.

To create a backdoored instance of LowMC-M with malicious tweak pair (t, t'), the linear layers $\mathsf{L}_1, \ldots, \mathsf{L}_r$ are chosen in a way that depends on $\mathsf{H}(t) + \mathsf{H}(t')$. Specifically, the designer chooses a secret input difference a_1 whose first s bits agree with $b_1 = t_1 + t'_1$. As a result, the difference propagates through the first partial S-box layer with probability one. The liner layer L_1 is then chosen such that the first s bits of $a_2 = \mathsf{L}_1(a_1 + b_1)$ agree with $b_2 = t_2 + t'_2$. This process can be repeated an arbitrary number of times.

Peyrin and Wang [234] argue that, since the malicious round tweak difference is unique with overwhelming probability, finding a malicious tweak pair costs roughly $2^{(n+(r-1)s)/2}$ evaluations of H – assuming that H is collision-resistant and the tweak is long enough. As noted by the authors, this reasoning does not take into account the existence of tweak pairs which might be a backdoor for a different input difference. The next section shows that it is easier to find alternative weak tweak pairs that result in a probability-one differential over all r rounds than it is to find the malicious tweak pair.

12.2.2 Weak tweak pairs

As described in Section 12.2.1, the specification of LowMC-M requires the designer to choose the secret input difference a_1 at random from $\mathbb{F}_2^n$. The authors argue that an attacker who tries to find the weak tweak pair has to match all n bits of a_1, in addition to $(r-1)s$ bits of the intermediate round

tweak differences. However, the analysis below shows that finding a collision on $rs - n < n + (r-1)s$ (for $s < 2n$) bits is actually sufficient. This is due to the fact that the input difference provides n additional degrees of freedom.

In the discussion below, a round tweak difference $\mathsf{H}(t) + \mathsf{H}(t') = (b_1, b_2, \ldots, b_r)$ will be called *weak* if there exists a differential characteristic $(a_1, \ldots, a_{r_1})$ with probability one for the first r_1 rounds of LowMC-M. If the difference in the first $i \geq 1$ rounds propagates deterministically, then $a_{i+1} = \mathsf{L}_i(a_i + b_i)$. Hence,

$$a_i = (\mathsf{L}_{i-1} \circ \cdots \circ \mathsf{L}_1)(a_1) + \sum_{j=1}^{i-1} (\mathsf{L}_{i-1} \circ \cdots \circ \mathsf{L}_j)(b_j).$$

Let $\lfloor x \rfloor_s$ denote the first s coordinates of x in $\mathbb{F}_2^n$. The probability in the first r_1 rounds is equal to one if $\lfloor a_i \rfloor_s = \lfloor b_i \rfloor_s$ or equivalently

$$\sum_{j=1}^{i} \left\lfloor (\mathsf{L}_{i-1} \circ \cdots \circ \mathsf{L}_j)(b_j) \right\rfloor_s = \left\lfloor (\mathsf{L}_{i-1} \circ \cdots \circ \mathsf{L}_1)(a_1) \right\rfloor_s, \tag{12.1}$$

for all i in $\{1, \ldots, r_1\}$. The term $j = i$ should be interpreted as b_i.

For any fixed choice of $b_1, \ldots, b_{r_1}$, (12.1) results in a system of sr_1 linear equations in n unknowns over $\mathbb{F}_2$. For random linear layers, and assuming $sr_1 \gg n$, such a system will be inconsistent with high probability. More precisely, the probability that a uniform random choice of the first r_1 round tweaks results in a right-hand side that makes the system consistent, is 2^{n-sr_1}. Indeed, the column space of the coefficient matrix of the linear system is of dimension n in an ambient space of dimension sr_1 [234, p. 21-22].

A tweak pair such that the round-tweak differences $(b_1, \ldots, b_{r_1})$ result in a consistent linear system can be found by using collision search methods at the cost of roughly $2^{(sr_1-n)/2}$ evaluations of H. The amount of memory required depends on the input size of H. For all applications in this chapter, the length of the tweak exceeds $(sr_1 - n)/2$.

The collision search proceeds as follows. Let A in $\mathbb{F}_2^{sr_1 \times (sr_1-n)}$ be a matrix with column space the orthogonal complement of the column space of the coefficient matrix of the system of equations (12.1). Let B in $\mathbb{F}_2^{sr_1 \times nr}$ be the matrix mapping the round tweaks to the right-hand side of the equations. The goal is to find a collision for the function $f : \mathbb{F}_2^{\ell} \to \mathbb{F}_2^{sr_1-n}$ defined by $f(t) = A^{\mathsf{T}} B\, \mathsf{H}(t)$. Since t is sufficiently long, a parallel collision search using Van Oorschot-Wiener collision search costs roughly $2^{(sr_1-n)/2}$ evaluations of f with little memory [273]. A small constant factor is neglected here, but this is justified because the evaluation of f likely takes significantly less time than a single LowMC-M evaluation.

The analysis above shows that a full-round weak tweak pair can be found with a computational cost of approximately $2^{(rs-n)/2}$ evaluations of f. Although this is a much lower cost than the cost of finding the backdoor itself, it unfortunately does not allow the attacker to find a full-round weak tweak pair in less time than the security level of 2^k for any of the LowMC-M instances because $rs - n > 2k$. Nevertheless, if the adversary is capable of $2^c \leq 2^k$ evaluations of f, then it can find a weak tweak pair with a probability one differential for the first $r_1 = \lfloor (2c + n)/s \rfloor$ rounds of LowMC-M. It will be shown in Section 12.2.3 that this is sufficient to set up full-round key-recovery attacks on LowMC-M.

12.2.3 Key-recovery attacks

A key-recovery attack that uses a small amount of data can be obtained by slightly modifying the difference-enumeration attacks of Rechberger, Soleimany and Tiessen [238]. These attacks enumerate all possible state differences in the forward and backward direction, searching for a match in the middle. Once a match is found, the corresponding characteristic can be determined, which can in turn be used to recover the key. For simplicity, it will be assumed that $k = n$.

The attack covers the first r_1 rounds of the cipher using a deterministic difference. In LowMC without a tweak, the largest possible[1] choice of r_1 is $\lfloor n/s \rfloor$. In LowMC-M, however, this number of rounds can be significantly increased by choosing a good weak tweak pair. Due to the results in Section 12.2.2, the number of rounds r_1 can be increased to

$$r_1 = \left\lfloor \frac{2c + n}{s} \right\rfloor ,$$

at the cost of 2^c evaluations of f.

Let δ denote the average number of possible output differences over the S-box layer for a uniform random input difference. For LowMC, it holds that $\delta = (29/8)^{s/3}$ [238, §3.1.3]. In the next r_2 rounds, all δ^{r_2} possible differences in the forward direction are enumerated. In the final r_3 rounds, the differences are enumerated in the backward direction. The differences are matched in the middle, which means that $\delta^{r_2+r_3} < 2^n$ should hold in order to avoid random collisions. That is, $r_2 + r_3 < n/\log_2 \delta$ must hold. The complexity of this distinguisher is dominated by the list creation, which amounts to $\max\{\delta^{r_2}, \delta^{r_3}\}$ memory accesses.

For key-recovery, one also has to compute the characteristic followed by the inputs. This can be done in roughly $\delta^{r_2} + \delta^{r_3}$ time for each input pair using

[1] A few more rounds may be possible if s does not divide n.

a meet-in-the-middle approach. Due to the fact that the LowMC S-box is differentially 2-uniform, the key-recovery step requires only two plaintext pairs. In fact, as noted by Rechberger *et al.* [238, §4.2.1], it is optimistic to assume that two pairs are sufficient and slightly more data are probably necessary in practice. Sticking with the estimate of two pairs, the time complexity of the entire attack is dominated by $2(\delta^{r_2} + \delta^{r_3})$ storage operations. The storage requirements are $n(\delta^{r_2} + \delta^{r_3})$ bits. To optimize the time complexity, one should choose $r_2 \approx r_3$. Specifically,

$$r_2 = \left\lfloor \frac{r - r_1}{2} \right\rfloor \text{ and } r_3 = \left\lceil \frac{r - r_1}{2} \right\rceil .$$

Some sample complexities for full-round LowMC-M are given in Table 12.1. For all instances specified by Peyrin and Wang [234, Table 1] except those with the largest value of s (for $n = 128$, $s = 90$ and for $n = 256$, $s = 120$), the attack improves over brute-force.

Some improvements to the costs reported in Table 12.1 could be obtained by optimizing the trade-off between the precomputation T_{off} and T_{on}. However, since the optimal trade-off depends on the context, this will not be discussed.

It is worth noting that Liu, Isobe and Meier have proposed an improvement of the difference enumeration attack at Crypto 2021 [202]. By combining the observations from Section 12.2.3 with their own attacks, they obtain improved results on LowMC-M.

12.3 Simple instance of MALICIOUS

This section presents a trivial modification of any tweakable block cipher that introduces a backdoor satisfying the goals of the MALICIOUS framework. More specifically, the backdoor must be practical, in the sense that it leads to efficient key-recovery. In addition, it should be undetectable even if the mechanism of the backdoor is known.

Let $\mathsf{H} : \mathbb{F}_2^{\mathbb{N}} \to \mathbb{F}_2^{\tau}$ be a cryptographic hash function and let $\mathsf{E} : \mathbb{F}_2^{\kappa} \times \mathbb{F}_2^{\tau} \times \mathbb{F}_2^{n} \to \mathbb{F}_2^{n}$ be a tweakable block cipher with block length n, tweak length τ and key length $\kappa = n$. The malicious designer chooses a secret tweak $t^\star$ in $\mathbb{F}_2^{\tau}$ and computes $s = \mathsf{H}(t^\star)$. The chosen tweak $t^\star$ will serve as the secret backdoor. The designer then defines the tweakable block cipher $\widetilde{\mathsf{E}} : \mathbb{F}_2^{\kappa} \times \mathbb{F}_2^{\tau} \times \mathbb{F}_2^{n} \to \mathbb{F}_2^{n}$ as

$$\widetilde{\mathsf{E}}(k, t, x) = \begin{cases} \mathsf{E}(k, t, x) & \text{if } \mathsf{H}(t) \neq s \\ x + k & \text{if } \mathsf{H}(t) = s\,. \end{cases} \tag{12.2}$$

Table 12.1: Cost of several difference-enumeration attacks on LowMC-M instances with $n = k$. Memory requirements are denoted by M_{on} and listed in bits. Precomputation time is denoted by $T_{\text{off}} = 2^c$ and expressed in evaluations of f. Online time is denoted by T_{on} and expressed in storage operations.

n	$\lceil \log_2 T_{\text{off}} \rfloor$	s	r	r_1	r_2	r_3	$\lceil \log_2 T_{\text{on}} \rfloor$	$\lceil \log_2 M_{\text{on}} \rfloor$
128	128	3	208	128	40	40	76	82
		6	104	64	20	20	76	82
		9	70	42	14	14	80	86
		30	23	12	5	6	112	118
	96	3	208	106	51	51	97	103
		6	104	53	25	26	98	104
		9	70	35	17	18	101	107
	64	3	208	85	61	62	117	123
		6	104	42	31	31	117	123
		9	70	28	21	21	119	125
256	256	3	384	256	64	64	121	128
		9	129	85	22	22	125	132
		60	21	12	4	5	187	194
	196	3	384	213	85	86	161	168
		9	129	71	29	29	164	171
		60	21	10	5	6	224	231
	128	3	384	170	107	107	201	208
		9	129	56	36	37	207	214

In other words, if the backdoor $t^\star$ is used as the tweak, the tweakable block cipher $\widetilde{\mathsf{E}}$ simply applies the permutation $x \mapsto x + k$, which allows the malicious designer to recover the key k with one known plaintext/ciphertext pair. Due to this simple key-recovery attack, the backdoor fulfills the notion of *practicability* [234, §2.2]. If the hash function H is preimage resistant, then a user cannot feasibly recover the backdoor $t^\star$. Therefore, the backdoor fulfills the notion of *undiscoverability* [234, §2.2]. More generally, under the same assumption on H, a user cannot even prove the existence of a secret backdoor. The reason is that the user cannot distinguish between whether the tweakable block cipher defined by (12.2) was designed by a malicious designer who knows $t^\star$ and generated $s = \mathsf{H}(t^\star)$ accordingly or by an honest designer who simply chose a random s in $\mathbb{F}_2^m$. In other words, the backdoor fulfills the notion of *undetectability* [234, §2.2].

Hence, (12.2) fulfills the same security notions as the backdoor in the original MALICIOUS framework. However, similar to the original MALICIOUS framework, the backdoor in $\widetilde{\mathsf{E}}$ does not fulfill the notion of *untraceability*: once $\widetilde{\mathsf{E}}$ is queried with the tweak $t^\star$, the full backdoor is revealed. Note that untraceability implies public-key cryptography.

12.4 Malicious AES

This section shows how to construct a tweakable variant of the AES with a modified key-schedule to obtain a more natural backdoored cipher based on the MALICIOUS framework. Instead of constructing a probability-one differential over the cipher for a secret pair of tweak values as in the original MALICIOUS framework, Malicious AES has an invariant subspace for a secret tweak value.

12.4.1 Specification of Malicious AES

Recall from Chapter 1 that the unkeyed AES round function R is of the form

$$\mathsf{R} = \mathsf{MixColumns} \circ \mathsf{ShiftRows} \circ \mathsf{SubBytes}\,.$$

For a detailed description of MixColumns, ShiftRows and SubBytes, see Chapter 1 or the Rijndael book [107]. One round of the AES consists of the composition of R and a round key addition.

The round function of Malicious AES is identical to that of the AES, but its key schedule is different and it supports an arbitrary-length tweak. Note that, for other reasons, changing the AES key-schedule has been discussed previously, *e.g.* in [177] and [115] to increase the resistance of AES against dedicated attacks.

Let k in $\mathbb{F}_2^\kappa$ be a κ-bit master key. The partial (64-bit) round keys $k_1, \ldots, k_{11}$ in $\mathbb{F}_2^{64}$ are derived from the master key using a key-scheduling function. The details of this function are left open. For reasons discussed in Section 12.4.2, it is required that there is an efficient algorithm to uniquely determine 64 bits of k given the value of k_{11}. The actual round keys are equal to $k'_1, \ldots, k'_{11}$, where the i^{th} round key k'_i is defined by

$$k'_i = \begin{bmatrix} k_{i,1} & k_{i,5} & k_{i,1} & k_{i,5} \\ k_{i,2} & k_{i,6} & k_{i,2} & k_{i,6} \\ k_{i,3} & k_{i,7} & k_{i,3} & k_{i,7} \\ k_{i,4} & k_{i,8} & k_{i,4} & k_{i,8} \end{bmatrix} \text{ for } i = 1, \ldots, 10 \text{ and } k'_{11} = \begin{bmatrix} k_{11,1} & k_{11,5} & 0 & 0 \\ k_{11,2} & k_{11,6} & 0 & 0 \\ k_{11,3} & k_{11,7} & 0 & 0 \\ k_{11,4} & k_{11,8} & 0 & 0 \end{bmatrix},$$

with $k_{i,1}, \ldots, k_{i,8}$ the bytes of k_i.

In order to support arbitrary-length tweaks, the 64-bit partial round tweaks $t_1, \ldots, t_{10}$ are derived from the master tweak t using an extendable output function H, *i.e.*, $(t_1, \ldots, t_{10}) = \mathsf{H}(t)$. The round tweaks are then $t'_1, \ldots, t'_{10}$, with t'_i defined by

$$t'_i = \begin{bmatrix} c_{i,1} & c_{i,5} & t_{i,1} & t_{i,5} \\ c_{i,2} & c_{i,6} & t_{i,2} & t_{i,6} \\ c_{i,3} & c_{i,7} & t_{i,3} & t_{i,7} \\ c_{i,4} & c_{i,8} & t_{i,4} & t_{i,8} \end{bmatrix},$$

where $t_{i,1}, \ldots, t_{i,8}$ are the bytes of t_i and $c_{i,1}, \ldots, c_{i,8}$ are the bytes of c_i. The choice of the round constants $c_1, \ldots, c_{10}$ is discussed below.

Let $\mathsf{Add}_y : x \mapsto x + y$. The i^{th} round function is defined by $\mathsf{R}_i = \mathsf{Add}_{k'_{i+1}+t'_{i+1}} \circ \mathsf{R}$ with R the unkeyed AES round function. The tweakable block cipher Malicious AES can then be described as

$$\text{Malicious AES}_{k,t} = \mathsf{Add}_{k'_{11}} \circ \mathsf{ShiftRows} \circ \mathsf{SubBytes} \circ \mathsf{R}_9 \circ \cdots \circ \mathsf{R}_1 \circ \mathsf{Add}_{k'_1+t'_1},$$

where $k'_1, \ldots, k'_{11}$ and $t'_1, \ldots, t'_{10}$ are as defined above. This is identical to the definition of the AES up to the choice of the round keys.

To set up a backdoored instance of Malicious AES, the attacker chooses a secret tweak $t^\star$ and computes the values $(t^\star_1, \ldots, t^\star_{10}) = \mathsf{H}(t^\star)$. The round constants $c_1, \ldots, c_{10}$ are then chosen as $c_i = t^\star_i$ for $i = 1, \ldots, 10$.

Remark 12.1. This chapter only defines a simple malicious variant of AES, which uses 64-bit round keys derived from a master key of length κ bits. This approach is especially suitable when the master key is short, such as $\kappa = 64$. There are several straightforward methods to construct instances with longer keys. For instance, one can build a similar construction based on Rijndael-192 or Rijndael-256 [107]. Depending on the details, this might make the key-recovery mechanism of the backdoor more complicated, but there are no significant difficulties. ▷

12.4.2 Description of the backdoor

The backdoor in Malicious AES is based on an invariant subspace for the unkeyed round function R of the AES. For the secret backdoor tweak $t^\star$, this subspace is preserved up to the penultimate round of the cipher. Below, the invariant subspace for R, its extension to Malicious AES and the key-recovery procedure are discussed.

Let $\mathcal{I} \subseteq \mathbb{F}_{2^8}^{4\times 4}$ be the subspace of states with equal first and third columns, and equal second and fourth columns:

$$\mathcal{I} = \left\{ \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} & a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} & a_{3,1} & a_{3,2} \\ a_{4,1} & a_{4,2} & a_{4,1} & a_{4,2} \end{bmatrix} \;\middle|\; \forall i \in \{1,2,3,4\}, j \in \{0,1\} \colon a_{i,j} \in \mathbb{F}_{2^8} \right\} .$$

It is well known (see for example [194, Proposition 5]) that $\mathcal{I}$ is an invariant subspace for the unkeyed AES round function R, *i.e.*, $\mathsf{R}(\mathcal{I}) = \mathcal{I}$. Indeed, $\mathcal{I}$ is invariant under each of the three operations SubBytes, ShiftRows, and MixColumns. However, since $\mathcal{I}$ is in general not invariant under addition with a round key, it cannot be used as an invariant subspace over multiple AES rounds interleaved with round key additions.

When the cipher is evaluated with the malicious tweak value $t^\star$, each round tweak t'_i is contained in $\mathcal{I}$. Hence, $\mathcal{I}$ is an invariant subspace for $\mathsf{Add}_{k'_i + t'_i}$ and more generally for every tweaked and keyed round function R_i. The propagation of the invariant subspace is depicted in Figure 12.2. Iterating this property yields

$$(\mathsf{Add}_{k'_{11}} \circ \text{Malicious AES}_{k,t^\star})(\mathcal{I}) = \mathcal{I} .$$

Under the assumption that for a tweak $t \neq t^\star$, each value t_i of $(t_1, \ldots, t_{10}) = \mathsf{H}(t)$ is sampled from a uniform distribution over $\mathbb{F}_2^{64}$, the probability that, for fixed j, the round tweak t'_j is contained in $\mathcal{I}$ is equal to $(2^{-8})^8 = 2^{-64}$. Therefore, for a tweak $t \neq t^\star$, the invariant subspace property is likely already broken after the first round of Malicious AES. Hence, if H is cryptographically secure, then the backdoor fulfills the notions of *undiscoverability* and *undetectability*.

Given $t^\star$, the last round key k'_{11} can be recovered using a chosen plaintext attack with a single plaintext/ciphertext pair. Specifically, for x in $\mathcal{I}$ and $y = \text{Malicious AES}_{k,t^\star}(x)$, it holds that

$$k_{11,i} = y_{i,1} + y_{i,3} ,$$

$$k_{11,4+i} = y_{i,2} + y_{i,4} ,$$

with $y_{i,j}$ the byte in row $1 \leq i \leq 4$ and column $1 \leq j \leq 4$ of the ciphertext y. Hence, the 64-bit partial round key k_{11} can be recovered directly. From k_{11}, the master key k can be recovered by guessing the remaining $\kappa - 64$ bits. Therefore, if κ is sufficiently small, Malicious AES fulfills the notion of *practicability*.

Remark 12.2. An explicit security analysis of Malicious AES will not be provided in this chapter, because (i) most of the security arguments for AES are equally valid for Malicious AES and (ii) increasing the number of rounds of Malicious AES does not invalidate the backdoor but should invalidate most potential non-backdoor based attacks. ▷

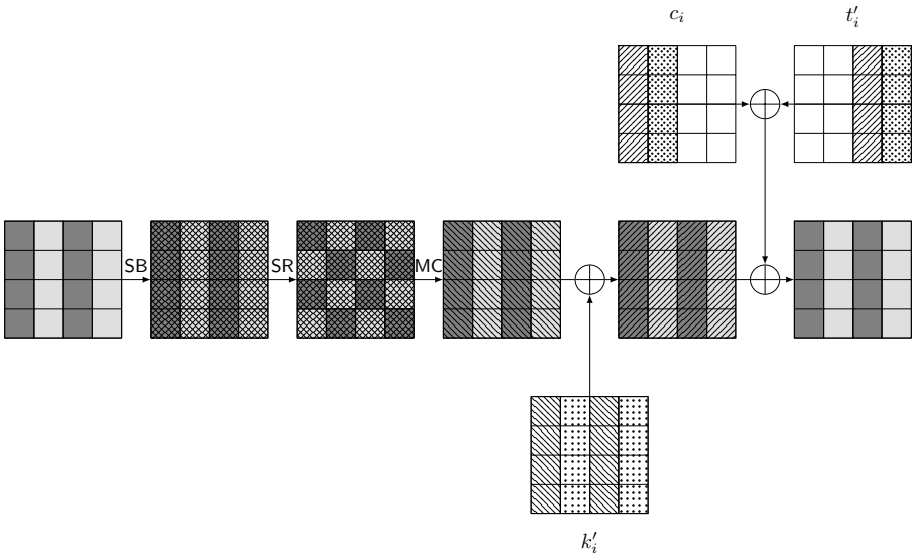


Figure 12.2: An invariant subspace for one round of Malicious AES with tweak $t^\star$. 'SB' is short for SubBytes, 'SR' for ShiftRows and 'MC' for MixColumns. Empty cells are zero.

12.5 Boomslang cipher

This section proposes the dedicated tweakable block cipher Boomslang. Similar to Malicious AES, the proposed cipher relies on the MALICIOUS framework to achieve undiscoverability. However, the backdoor is based on a nonlinear invariant rather than an invariant subspace. In fact, the backdoor implies the existence of an iterative perfect linear approximation over two rounds of the cipher. Hence, it can also be compared to the recently proposed block cipher ЯooD [235], which contains a backdoor based on linear cryptanalysis. However, the design rationale of ЯooD is weaker and it does not offer undiscoverability, so it has only limited practicability.

12.5.1 Specification of Boomslang

The cipher operates on 128-bit blocks and the state is represented by a 4×8 array of 4-bit cells. The key k is a 128-bit value, and the tweak t can be any bitstring of arbitrary (bounded) length.

The overall structure of the round function closely follows that of the AES and is shown in Figure 12.3. Specifically, the unkeyed round function of Boomslang can be written as

$$\mathsf{R} = \mathsf{MixColumns} \circ \mathsf{ShiftRows} \circ \mathsf{SubBytes}\,.$$

Below, each of the functions on the right-hand side will be briefly discussed.

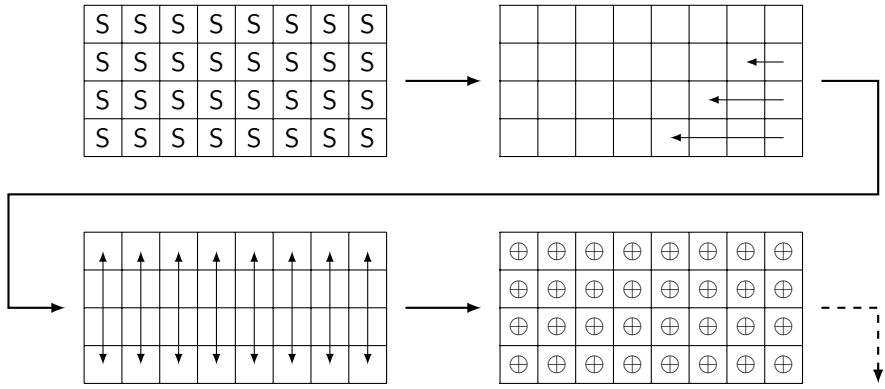


Figure 12.3: Overview of the round function: SubCells, ShiftRows, MixColumns and the addition of constants.

SubBytes consists of the parallel application of an S-box S to the 4-bit cells of the state. The S-box is the nonlinear function $\mathsf{S} : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ defined by Table 12.2.

ShiftRows is similar to the AES ShiftRows step. If the rows are numbered from zero to three with zero corresponding to the top row, then ShiftRows rotates the i^{th} row of the state over $4 \cdot i$ bits to the left.

MixColumns consists of a columnwise multiplication with a lightweight matrix from the family of quasi-MDS matrices that was proposed for Qarma [15]. Denote the cells within one column of the state by $(x_0, \ldots, x_3)$, where $x_i \in \mathbb{F}_2^4$. MixColumns maps each column $(x_0, \ldots, x_3)$ to a new column $(y_0, \ldots, y_3)$ defined by

$$y_i = x_{i+1} + (x_{i+2} \lll 1) + (x_{i+3} \lll 2)\,,$$

for $i = 0, \ldots, 3$ and where the addition of the indices is regarded modulo four. The inverse mapping is given by

$$x_i = y_{i+3} + (y_{i+1} \lll 2) + (y_{i+2} \lll 3)\,,$$

Table 12.2: The 4-bit S-box S for Boomslang.

0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
8	2	4	a	5	f	7	6	0	c	b	9	e	d	1	3

for $i = 0, \ldots, 3$. In software, MixColumns can be implemented using bitslicing.

The $(2i+1)^{\text{th}}$ round function is then defined as

$$\mathsf{R}_{2i+1} = \mathsf{Add}_{k_{i+1}} \circ \mathsf{Add}_{c_{2i+1}} \circ \mathsf{R}\,,$$

where c_i are round constants and k_i round keys. The round keys k_i can be derived using an arbitrary key schedule. Since we do not aim for related key security, we let $k_i = k$ for simplicity. The $2i^{\text{th}}$ round function is defined as

$$\mathsf{R}_{2i} = \mathsf{Add}_{t_i} \circ \mathsf{Add}_{c_{2i}} \circ \mathsf{R}\,.$$

The i^{th} round tweak t_i is derived from the master tweak t using a strong tweak-scheduling function. Specifically, $(t_1, t_2, \ldots) = \mathsf{H}(t)$ where H is an extendable output function. Some specific conditions on the round constants c_i, all of which are likely to be satisfied for a uniform random choice, are discussed in Section 12.5.2.

The overall cipher is constructed as the composition of the round functions $\mathsf{R}_1, \mathsf{R}_2, \ldots$. The number of rounds, denoted by r, is required to be even, so that the last round function always performs a round-tweak addition. Based on the analysis in Section 12.5.2, $r = 32$ is a reasonable choice. The key k is also added to the plaintext and $\sigma(k)$ is added to the state after the last round, with σ a linear orthomorphism, *i.e.*, both $k \mapsto \sigma(k)$ and $k \mapsto \sigma(k) + k$ are permutations. Specifically, let $\sigma(k) = (k \ggg 1) + (k \gg 63)$ as in PRINCE [76]. In summary, the overall cipher is defined by

$$\mathsf{Boomslang} = \mathsf{Add}_{\sigma(k)} \circ \mathsf{R}_r \circ \cdots \circ \mathsf{R}_2 \circ \mathsf{R}_1 \circ \mathsf{Add}_k\,.$$

To set up a backdoored instance of Boomslang, the designer chooses a secret tweak $t^\star$ and computes the corresponding round tweaks $(t_1^\star, t_2^\star, \ldots) = \mathsf{H}(t^\star)$. The round constants c_{2i} should then be chosen as $c_{2i} = t_i^\star$ for $i = 1, \ldots, r/2$. Given a few hundred known plaintext/ciphertext pairs encrypted under the tweak $t^\star$, it will then be possible to efficiently recover the full key. The mechanism for doing so will be explained in Section 12.5.3.

12.5.2 Design rationale

In this section, it is argued that the design of Boomslang follows standard modern design principles. The security analysis focuses on differential and linear cryptanalysis, and invariants. The latter two are of particular relevance with respect to the backdoor that will be revealed in Section 12.5.3.

The design follows the wide-trail strategy [107, Chapter 9] with some changes to obtain a more lightweight cipher. Whenever possible, the design was kept as simple as possible and close to that of the AES.

In general, the proposed cipher is geared towards hardware. This is the motivation for relying on 4-bit S-boxes rather than 8-bit S-boxes as in the AES. In software, the 4×8 state allows storing the rows as 32-bit words. The S-box and linear layer can then be implemented using bitslicing.

The key schedule is chosen as the identity function, although other key schedules could also be used. Since related key security was not a design goal, we decided to choose the simplest option. In addition, having a linear key schedule sometimes enables more straightforward security arguments. For example, the arguments from [27] related to the choice of round constants to prevent invariants are only applicable to linear key schedules.

Finally, the choice of the tweak schedule can be motivated by the goal of supporting arbitrary-length tweaks. Since related tweak security is important, it seems necessary to use a cryptographically strong hash function or extendable output function to derive round tweaks from the master tweak.

All of the basic components used in the cipher are individually acceptable choices from the point of view of the current state of the art.

SubCells. The S-box has a maximum absolute correlation of $1/2$ for nonzero masks and a maximum differential probability of $1/4$ for nonzero differences. The S-box is chosen such that it is not an involution.

ShiftRows. The cell permutation is chosen such that the cells of each column end up in different columns of the state. Shifting rows is a natural choice because it allows for an efficient software implementation, and it is the same as for the AES.

MixColumns. The MixColumns map is inspired by the linear layer of Qarma [15]. Specifically, the transformation of each column is defined by a circulant

matrix M of the form

$$M = \begin{bmatrix} 0 & X^a & X^b & X^c \\ X^c & 0 & X^a & X^b \\ X^b & X^c & 0 & X^a \\ X^a & X^b & X^c & 0 \end{bmatrix},$$

over the $\mathbb{F}_2$-vector space $\mathbb{F}_2[X]/(X^4 + 1)$. The input bitvector can be considered as an element of this space by the isomorphism $\delta_i \mapsto X^{i-1}$, where δ_i is the i^{th} standard basis vector of $\mathbb{F}_2^4$.

The matrix M is invertible with circulant inverse of the same form if and only if $a \equiv c \pmod 4$ or $a \equiv c + 2 \pmod 4$. All of these matrices have branch number four, which is the maximum possible for this type of matrix. Furthermore, the following criteria are imposed:

- Unlike in Qarma, we require that M is not an involution. Equivalently, $2b \not\equiv 0 \pmod 4$. The motivation for this requirement is that involutions more easily lead to 2-round invariants, as demonstrated in the case of Midori-64 (see Chapter 6).
- M should not be orthogonal or nearly orthogonal, *i.e.* $M^{-1} \neq \alpha M^{\mathsf{T}}$ for any α in $\mathbb{F}_2[X]/(X^4 + 1)$. This requirement is motivated by the fact that any quadratic form $\sum_{i=1}^{m} x_i^{\mathsf{T}} Q x_i$ is a nonlinear invariant for an $m \times m$ orthogonal matrix [267]. More generally, for a nearly orthogonal matrix, any such quadratic function which is also invariant under multiplication by α is a nonlinear invariant.

The second criterion leads to the requirement that $X^{a+b} \neq X^{b+c}$ or equivalently $a \not\equiv c \pmod 4$. From the viewpoint of software implementations, it makes sense to choose one of a, b or c equal to zero. Choosing $a = 0$ and $b = 1$ then gives $c = 2$.

The wide-trail strategy directly gives upper bounds on the absolute correlation of linear trails and on the probability of differential characteristics. In particular, since M has a branch number of four, the number of active S-boxes over four rounds is at least 16 [107, Theorem 9.4.1]. Hence, after 16 rounds the average probability of any differential characteristic is lower than 2^{-128} and the absolute correlation of any linear trail is at most 2^{-64}. The suggested choice of 32 rounds was obtained by taking twice as many rounds – taking into account potential improvements and key-recovery attacks. In fact, it is to some extent possible to extend the aforementioned upper bounds to linear approximations and differentials. In particular, for independent uniform random constants, [226, Corollary 1 & 2] imply that the average probability of any 4-round differential and the average squared correlation of any 4-round linear approximation is at most $(2^{-2\cdot(4-1)})^4 = 2^{-24}$.

Several lightweight ciphers have been found vulnerable to invariant subspace [196] and nonlinear invariant attacks [267]. Hence, it is natural to attempt to rule out the existence of invariants in Boomslang. The argument from [27] can be used to rule out joint invariants over all the affine layers (*i.e.*, linear layers together with the constant additions) for a large number of rounds using only the properties of the linear layer and the round constants. Specifically, the security argument depends on the dimension of the smallest subspace invariant under the linear layer and containing the differences of the constants. For the linear layer $\mathsf{L} = \mathsf{MixColumns} \circ \mathsf{ShiftRows}$ of Boomslang and constants $c_1, \ldots, c_r$, denote this space by $W_{\mathsf{L}}(c_1 + c_2, c_1 + c_3, \ldots, c_1 + c_r)$. If $W_{\mathsf{L}}(c_1 + c_2, c_1 + c_3, \ldots, c_1 + c_r) = \mathbb{F}_2^{128}$, then joint invariants for the affine layers can be ruled out with high probability. The linear map L has 16 invariant factors and its minimal polynomial is $(X + 1)^8$. Hence, by [27, Proposition 11],

$$\Pr_{c_1,\ldots,c_{24}}[\dim W_L(c_1 + c_2, c_1 + c_3, \ldots, c_1 + c_{24}) = 128] = \prod_{i=0}^{15} \left(1 - \frac{1}{2^{23-i}}\right) \geq 0.99\,,$$

for uniformly chosen random constants $c_1, \ldots, c_{24}$. Hence, 24 rounds are sufficient to rule out with high probability the existence of such invariants. Note that this argument does not yet rule out invariants over a small number of rounds and also does not rule out invariants that are not invariant for every round as in Chapter 6.

Most invariants considered in previous attacks are of rank-one type, as discussed in Chapter 6. Indeed, this leads to an easier analysis of the SubCells and ShiftRows steps. To investigate this in more detail, the tool from Section 3.7.2 was used to obtain the rank-one invariants of the linear layer M. Although M has some rank-one invariants, they do not correspond to Boolean functions or sets, and there are no shared invariants between M and the S-box layer.

12.5.3 Description of the backdoor

The backdoor is a two-round invariant, which is not invariant for one round. This is similar to the invariant for two rounds of Midori-64 that was described in Section 6.4.3, but unlike in that case the property is not invariant under the linear layer. Indeed, as discussed above, that would not be possible due to the choice of the linear layer. Importantly, the invariant only exists for the secret weak tweak for which the round constants in even rounds cancel out.

Let $f : \mathbb{F}_2^4 \to \mathbb{F}_2$ and $g : \mathbb{F}_2^4 \to \mathbb{F}_2$ be the Boolean functions defined by

$$f(z_1, z_2, z_3, z_4) = (z_1 + z_3)(z_2 + z_4) + z_1 + z_3 + z_4 + 1$$

$$g(z_1, z_2, z_3, z_4) = (z_1 + z_3)(z_2 + z_4) + z_3 \,.$$

The functions f and g can be used to form a perfect nonlinear approximation of M. This is due to the fact that the term $(z_1 + z_3)(z_2 + z_4)$ is invariant under rotations of $z_1, \ldots, z_4$. Hence, if $y = \mathsf{MixColumns}(x)$, then

$$\sum_{i=1}^{32} g(y_{4i-3}, y_{4i-2}, y_{4i-1}, y_{4i}) = \sum_{i=1}^{32} f(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}) \,.$$

Furthermore, it is easy to see that

$$\sum_{i=1}^{32} \mathsf{a}^\mathsf{T}(y_{4i-3}, y_{4i-2}, y_{4i-1}, y_{4i}) = \sum_{i=1}^{32} \mathsf{5}^\mathsf{T}(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}) \,.$$

The S-box S defined in Table 12.2 also satisfies

$$\mathsf{5}^\mathsf{T}\mathsf{S}(z_1, z_2, z_3, z_4) = g(z_1, z_2, z_3, z_4)$$

$$f(\mathsf{S}(z_1, z_2, z_3, z_4)) = \mathsf{a}^\mathsf{T}(z_1, z_2, z_3, z_4) \,.$$

Since linear functions are invariant under the addition of any constant, and because the constants are cancelled out by the tweak in even rounds, one obtains the following two-round invariant:

$$\sum_{i=1}^{32} g(y_{4i-3}, y_{4i-2}, y_{4i-1}, y_{4i}) = \gamma + \sum_{i=1}^{32} g(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i}) \,,$$

where $y = (\mathsf{R}_{2i} \circ \mathsf{R}_{2i-1})(x)$ and γ is a key-dependent constant. The full nonlinear trail is illustrated in Figure 12.4. Note that the last step only works for one in 2^{64} constants, but the constants are chosen such that there exists a tweak so that the constants are weak in all odd-numbered rounds.

Alternatively, the nonlinear invariant discussed above can be described as in Chapter 6. Let

$$w = (0, -1, 0, 0, 1, 0, 0, 0, 0, 0, 0, -1, 0, \quad 0, -1, 0)/2$$

$$v = (0, \quad 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, \quad 0, 0, -1, \quad 0, 0)/2 \,.$$

In the above, w and v are the Walsh-Hadamard transform of f and g respectively. It holds that $C^M w^{\otimes 4} = v^{\otimes 4}$, with C^M the correlation matrix of the linear layer.

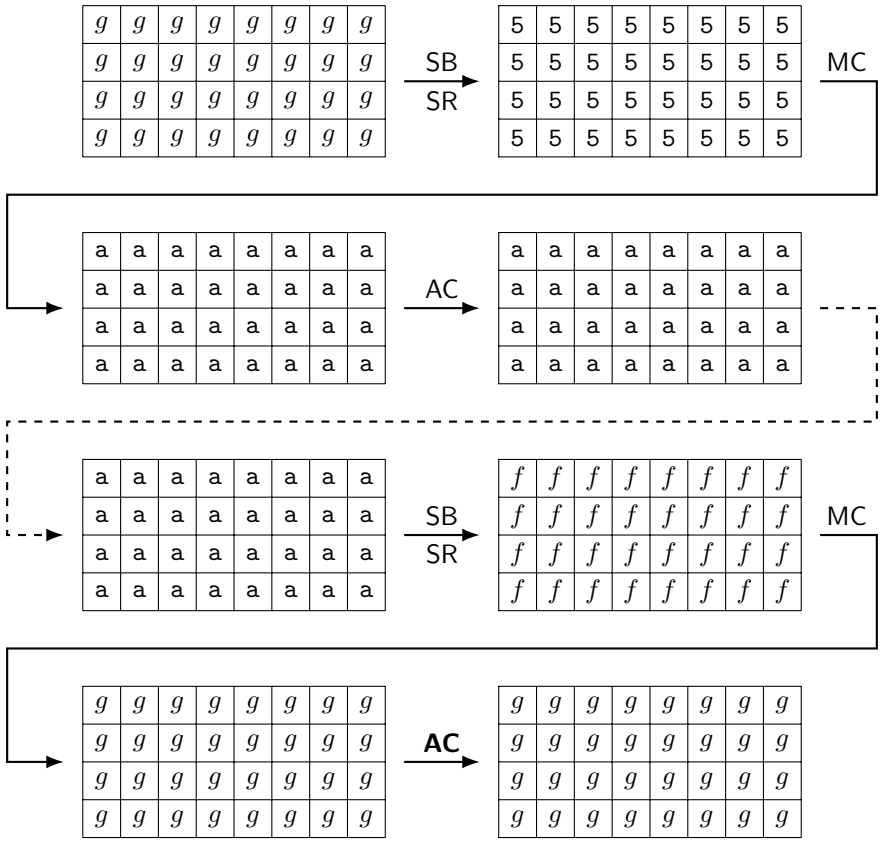


Figure 12.4: Two-round invariant for Boomslang.

Furthermore, the S-box satisfies $C^{\mathsf{S}}v = \delta_{\mathsf{5}}$ and $C^{\mathsf{S}}\delta_{\mathsf{a}} = w$ when $\mathbb{F}_2^{16}$ is identified with its dual. The vector v is invariant under one in four constants.

The addition of whitening keys k and $\sigma(k)$ leads to an efficient key-recovery attack. Specifically, one can use the fact that there exists a mask ℓ in $\mathbb{F}_2^{128}$ and a constant b in $\mathbb{F}_2$ such that for every plaintext/ciphertext pair (x, y) encrypted under the backdoored tweak,

$$\sum_{i=1}^{32} g(x_i + k_i) + \sum_{i=1}^{32} g(y_i + \sigma(k)_i) = \ell^{\mathsf{T}} k + b\,,$$

with $x_1, \ldots x_{32}$, $y_1, \ldots, y_{32}$ and $k_1, \ldots, k_{32}$ the nibbles of x, y and k, respectively. Since σ is an orthomorphism, the 64 bits of k that are nonlinearly mixed with x are linearly independent from the bits of k that are nonlinearly mixed with

y. Hence, given q messages, one can on average recover q bits of the key even when $q \geq 64$.

Solving the system of equations is easy because of the low number of quadratic terms. One can either use Gröbner basis methods, exploiting the low degree of regularity of the system, or one can directly rely on linearization. Since g contains only a single quadratic term, each equation contains at most 64 quadratic terms. Hence, given 192 known plaintext/ciphertext pairs, the full key can be recovered using less than $192^3 \leq 2^{23}$ bit operations.

Remark 12.3 (Construction of the backdoor). The construction of the backdoor primarily relies on the choice of the S-box. The tool from Section 3.7.2 was used to find symmetric nonlinear rank-one approximations of the linear layer. This resulted in the choice of the vectors w and v listed above. One can then easily generate S-boxes such that the conditions $C^{\mathsf{S}}v = \delta_{\mathsf{5}}$ and $C^{\mathsf{S}}\delta_{\mathsf{a}} = w$ are satisfied. There are still significant degrees of freedom left in the choice of the S-box. These could be used to satisfy additional design criteria, or to argue that the S-box was generated based on certain magic constants. ▷

12.6 Limitations

The ciphers presented in this chapter do not hide the general mechanism of their backdoor. Hence, using the malicious tweak once potentially spoils the backdoor. Using the terminology introduced by the authors of the MALICIOUS framework, none of the backdoors presented in the symmetric-key literature is untraceable. In fact, this limitation is precisely the gap between symmetric-key and asymmetric-key cryptography.

Nevertheless, achieving untraceability may not be necessary for a successful backdoor in practice. It is sufficient that the mechanism of the backdoor is difficult to uncover using state-of-the-art – or perhaps just 'well-known' – techniques. The examples in this chapter follow standard design principles and, although this cannot be proven, it seems likely that a cryptanalyst with limited knowledge about nonlinear invariants would not have had much success in uncovering the mechanism of the Boomslang backdoor. Furthermore, even if the mechanism is found, the designer still has a strong counterargument as long as the malicious tweak is not revealed. For example, as shown in Chapter 6, the designers of Midori-64 introduced a similar two-round invariant *by accident*.

Finally, it should be noted that Boomslang's backdoor is designed to be simple. If this requirement is dropped, then a variety of methods could have been used to make it more difficult to uncover its mechanism.

13

Conclusion

The following two sections summarize the main contributions of this thesis and suggest some directions for future work. Section 13.1 presents the conclusions of Part I on the theory of cryptanalysis. Section 13.2 concludes Part II of this thesis, on the applications of cryptanalysis.

13.1 Theory

As mentioned in Chapter 1, the main goal of Part I of this thesis was the development of a general approach to symmetric-key cryptanalysis. Such a theory was proposed in Chapter 2. It describes cryptanalytic properties as pairs of subspaces of the free k-vector space on a set, and its dual space. There should be an absolute value function defined on the field k, giving the approach a geometric flavor. Propagation is described by pushforward and pullback operators. For properties defined by one-dimensional subspaces, expressing these operators as matrices relative to a basis yields a general notion of trails. The basis can be chosen to diagonalize a group or monoid action. Using the dominant trail approximation, trails make it possible to evaluate the properties of iterated functions. For the case of higher-dimensional subspaces, basis-free generalizations of these concepts were developed.

It was shown in Chapter 3 that choosing a basis that diagonalizes the action of a commutative group yields the theory of ordinary linear cryptanalysis, and more specifically its description using correlation matrices. The higher-dimensional case of the theory was used to describe extensions of linear cryptanalysis, and sheds light on the connections between them. Consequences include a characterization of invariant subspaces and nonlinear invariants as eigenvectors of correlations matrices, an intuitive link between zero-correlation and perfect approximations, and additional insight into the relation between invariants and linear approximations. Linear cryptanalysis takes place over the complex numbers, giving additional geometric structure in form of an inner product. This leads to the principal correlations of an approximation, which were shown to determine the data complexity of known-plaintext distinguishers. Finally, rank-one approximations were introduced.

Applying the one-dimensional geometric approach to differential cryptanalysis shows that its standard description is incomplete. This led to the introduction of quasidifferential trails in Chapter 4. The quasidifferential basis diagonalizes the action of a commutative group on pairs of elements, while including the indicator functions of sets of pairs with a constant difference as basis functions. Expressing the pushforward operator in this basis leads to quasidifferential transition matrices. The properties of these matrices were discussed, and it was shown how they can be computed. Together with the dominant trail approximation, quasidifferential trails provide a practical solution to the problem of independence heuristics. They can be used to calculate the probability of differential characteristics without the need to rely on the hypothesis of stochastic equivalence and the independence of round keys.

Chapter 5 introduced a generalization of integral cryptanalysis by applying the geometric approach over an extension field of the p-adic numbers. The one-dimensional theory of ultrametric trails was constructed by choosing a basis that diagonalizes the action of a commutative inverse monoid. Specializing this theory to $\mathbb{F}_q^n$ (for q a power of p) with its coordinate-wise product yields a rich extension of integral cryptanalysis. Ordinary integral cryptanalysis is obtained by reducing the theory for $q = 2$ to the residue field. It was shown how the ultrametric triangle inequality makes it possible to use the dominant trail approximation to deduce 'approximate zero-correlation' properties. The relation between the ultrametric transition matrix of a function and its algebraic normal form was investigated. The theory suggests natural extensions of parity sets and the conventional division property, with divisibility by powers of p leading to a spectrum of properties between zero-sums and saturation. As a proof of concept, it was shown that several zero-sum properties of reduced-round PRESENT are in fact stronger properties of divisibility by a higher power of two.

That it is at all possible to combine all three techniques (linear, differential and integral cryptanalysis) into a single framework, is perhaps one of the great advantages of symmetric-key cryptanalysis compared to other areas of cryptography. The author of this thesis hopes, but also expects, that the proposed geometric approach will lead to further developments. On the one hand, the multidimensional theories of differential and integral cryptanalysis have not yet been investigated. On the other hand, especially in the differential case, several other bases are worth examining for the one-dimensional theory. Ultrametric trails also warrant further research. Another major area will be the extension to properties based on triplets and beyond. This direction has received little attention, but looks especially promising.

13.2 Applications

Applications of the geometric approach were given in Chapters 6 to 8. In Chapter 6, the characterization of block cipher invariants as eigenvectors of correlation matrices was used to obtain reduced-round attacks on the block ciphers Midori-64 and MANTIS. These attacks only work for weak keys, but their data complexity is low compared to previous work. The South-Korean and American format-preserving encryption standards FEA and FF3-1 were broken in Chapter 3, with data- and time-complexities that are low enough to be a practical concern in some applications. The analysis of FF3-1 relies on multidimensional linear cryptanalysis over $\mathbb{Z}/N\mathbb{Z}$. Chapter 8 reevaluated differential attacks on Rectangle, KNOT and Speck using the theory of quasidifferential trails. Some attacks were shown to be invalid, others only work for weak keys.

Chapters 9 and 10 discussed other cryptanalytic attacks. In Chapter 9, a generic truncated differential attack on contracting Feistel ciphers was used to attack the Chinese commercial encryption standard SM4 with a reduced number of rounds. The simplest variant of the same generic attack, and a similar one for expanding Feistel ciphers, gives full-round attacks on some instances of GMiMC-erf and GMiMC-crf in Chapter 10. This chapter also included cryptanalytic results on other arithmetization-oriented primitives. In particular, it was shown how the partial linear layer of HadesMiMC can be exploited to set up integral distinguishers and, for some choices of the MDS matrix, preimage attacks. Finally, improved attacks on the Legendre PRF and its variants were presented.

Chapters 11 and 12 took up unconventional applications of cryptanalysis. An application of linear cryptanalysis to the design of side-channel countermeasures was presented in Chapter 11. The bounded-query probing model was introduced to capture adversaries that can make only a bounded number of probing queries, and it was shown that the advantage of such adversaries can be bounded in terms of the correlations of linear approximations over the masked primitive. This approach makes it possible to achieve second-order probing security without using any randomness beyond what is necessary to share the state. In Chapter 12, the MALICIOUS framework for constructing backdoored tweakable block ciphers was reconsidered. An attack on LowMC-M was given, and a trivial instance of the framework was pointed out. In addition, the tweakable block ciphers Malicious AES and Boomslang were proposed. They follow standard design principles, yet embed a MALICIOUS-style backdoor based on invariants.

Due to the chronology of this thesis, not all of the theoretical results from Part I have found their way to applications. Some ideas from Chapter 3, such as rank-one approximations, have not yet been pursued with sufficient force. Although Chapter 8 demonstrated that quasidifferential trails are useful to correct errors

in previous analyses, they did not yet serve as the basis for new attacks. For this, the direction of collision attacks on hash functions seems promising. In retrospect, the theory of integral cryptanalysis from Chapter 5 would have been useful for the cryptanalysis of arithmetization-oriented primitives in Chapter 10.

As a result of the attacks on FEA and FF3-1 from Chapter 7, South-Korea and the United States will have to revise their format-preserving encryption standards and look for alternatives. The cryptanalysis of arithmetization-oriented primitives in Chapter 10 contributed to StarkWare's hash-function choice. Future designs, especially those based on generalized Feistel networks or using partial S-box layers, should also take into account these attacks. Likewise, if Ethereum wants to use the Legendre PRF in its protocols, then the analysis in Chapter 10 will have to be taken into account.

Although Chapter 11 showed how the randomness requirements of masked ciphers can be reduced, side-channel cryptanalysis and countermeasures have a long way to go. The analytic techniques used in most attacks are unsophisticated, in part due the difficultly of making precise security claims. Countermeasures do not take into account the way randomness is generated, and addressing this issue would require moving away from information-theoretical security notions.

Finally, although the ciphers Malicious AES and Boomslang achieve the goals of the MALICIOUS framework, the backdoor is potentially revealed as soon as it is used. The construction of ciphers with truly hidden backdoors is of great interest, because it implies public-key cryptography.

Bibliography

[1] Abdelraheem, M. A., Ågren, M., Beelen, P., and Leander, G. On the distribution of linear biases: Three instructive examples. In *CRYPTO 2012* (Aug. 2012), R. Safavi-Naini and R. Canetti, Eds., vol. 7417 of *LNCS*, Springer, Heidelberg, pp. 50–67.

[2] Abed, F., List, E., Lucks, S., and Wenzel, J. Differential cryptanalysis of round-reduced Simon and Speck. In *FSE 2014* (Mar. 2015), C. Cid and C. Rechberger, Eds., vol. 8540 of *LNCS*, Springer, Heidelberg, pp. 525–545.

[3] Aladov, N. Sur la distribution des résidus quadratiques et non-quadratiques d'un nombre premier p dans la suite $1, 2, \ldots, p-1$. *Matematicheskii Sbornik 18*, 1 (1896), 61–75.

[4] Albertini, A., Aumasson, J.-P., Eichlseder, M., Mendel, F., and Schläffer, M. Malicious hashing: Eve's variant of SHA-1. In *SAC 2014* (Aug. 2014), A. Joux and A. M. Youssef, Eds., vol. 8781 of *LNCS*, Springer, Heidelberg, pp. 1–19.

[5] Albrecht, M. R., Cid, C., Grassi, L., Khovratovich, D., Lüftenegger, R., Rechberger, C., and Schofnegger, M. Algebraic cryptanalysis of STARK-friendly designs: Application to MARVELlous and MiMC. In *ASIACRYPT 2019, Part III* (Dec. 2019), S. D. Galbraith and S. Moriai, Eds., vol. 11923 of *LNCS*, Springer, Heidelberg, pp. 371–397.

[6] Albrecht, M. R., Grassi, L., Perrin, L., Ramacher, S., Rechberger, C., Rotaru, D., Roy, A., and Schofnegger, M. Feistel structures for MPC, and more. In *ESORICS 2019, Part II* (Sept. 2019), K. Sako, S. Schneider, and P. Y. A. Ryan, Eds., vol. 11736 of *LNCS*, Springer, Heidelberg, pp. 151–171.

[7] Albrecht, M. R., Grassi, L., Rechberger, C., Roy, A., and Tiessen, T. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In *ASIACRYPT 2016, Part I* (Dec. 2016), J. H. Cheon and T. Takagi, Eds., vol. 10031 of *LNCS*, Springer, Heidelberg, pp. 191–219.

[8] Albrecht, M. R., Rechberger, C., Schneider, T., Tiessen, T., and Zohner, M. Ciphers for MPC and FHE. In *EUROCRYPT 2015,*

Part I (Apr. 2015), E. Oswald and M. Fischlin, Eds., vol. 9056 of *LNCS*, Springer, Heidelberg, pp. 430–454.

[9] Aly, A., Ashur, T., Ben-Sasson, E., Dhooghe, S., and Szepieniec, A. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Trans. Symm. Cryptol. 2020*, 3 (2020), 1–45.

[10] Amon, O., Dunkelman, O., Keller, N., Ronen, E., and Shamir, A. Three third generation attacks on the format preserving encryption scheme FF3. In *EUROCRYPT 2021, Part II* (Oct. 2021), A. Canteaut and F.-X. Standaert, Eds., vol. 12697 of *LNCS*, Springer, Heidelberg, pp. 127–154.

[11] Ankele, R., Dobraunig, C., Guo, J., Lambooij, E., Leander, G., and Todo, Y. Zero-correlation attacks on tweakable block ciphers. *IACR Trans. Symm. Cryptol. 2019*, 1 (2019), 192–235.

[12] Ankele, R., and Kölbl, S. Mind the gap - A closer look at the security of block ciphers against differential cryptanalysis. In *SAC 2018* (Aug. 2019), C. Cid and M. J. Jacobson Jr:, Eds., vol. 11349 of *LNCS*, Springer, Heidelberg, pp. 163–190.

[13] Ashur, T., Beyne, T., and Rijmen, V. Revisiting the wrong-key-randomization hypothesis. *Journal of Cryptology 33*, 2 (Apr. 2020), 567–594.

[14] Ashur, T., and Dhooghe, S. MARVELlous: a STARK-friendly family of cryptographic primitives. Cryptology ePrint Archive, Report 2018/1098, 2018. `https://eprint.iacr.org/2018/1098`.

[15] Avanzi, R. The QARMA block cipher family. *IACR Trans. Symm. Cryptol. 2017*, 1 (2017), 4–44.

[16] Baignères, T., Junod, P., and Vaudenay, S. How far can we go beyond linear cryptanalysis? In *ASIACRYPT 2004* (Dec. 2004), P. J. Lee, Ed., vol. 3329 of *LNCS*, Springer, Heidelberg, pp. 432–450.

[17] Baignères, T., Stern, J., and Vaudenay, S. Linear cryptanalysis of non binary ciphers. In *SAC 2007* (Aug. 2007), C. M. Adams, A. Miri, and M. J. Wiener, Eds., vol. 4876 of *LNCS*, Springer, Heidelberg, pp. 184–211.

[18] Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., and Regazzoni, F. Midori: A block cipher for low energy. In *ASIACRYPT 2015, Part II* (Nov. / Dec. 2015), T. Iwata and J. H. Cheon, Eds., vol. 9453 of *LNCS*, Springer, Heidelberg, pp. 411–436.

[19] Bannier, A., Bodin, N., and Filiol, E. Partition-based trapdoor ciphers. Cryptology ePrint Archive, Report 2016/493, 2016. https://eprint.iacr.org/2016/493.

[20] Bar-On, A., Dinur, I., Dunkelman, O., Lallemand, V., Keller, N., and Tsaban, B. Cryptanalysis of SP networks with partial non-linear layers. In *EUROCRYPT 2015, Part I* (Apr. 2015), E. Oswald and M. Fischlin, Eds., vol. 9056 of *LNCS*, Springer, Heidelberg, pp. 315–342.

[21] Bardet, M., Faugère, J.-C., and Salvy, B. On the complexity of the F5 gröbner basis algorithm. *Journal of Symbolic Computation 70* (2015), 49–70.

[22] Barthe, G., Belaïd, S., Cassiers, G., Fouque, P.-A., Grégoire, B., and Standaert, F.-X. maskVerif: Automated verification of higher-order masking in presence of physical defaults. In *ESORICS 2019, Part I* (Sept. 2019), K. Sako, S. Schneider, and P. Y. A. Ryan, Eds., vol. 11735 of *LNCS*, Springer, Heidelberg, pp. 300–318.

[23] Barthe, G., Belaïd, S., Dupressoir, F., Fouque, P.-A., Grégoire, B., and Strub, P.-Y. Verified proofs of higher-order masking. In *EUROCRYPT 2015, Part I* (Apr. 2015), E. Oswald and M. Fischlin, Eds., vol. 9056 of *LNCS*, Springer, Heidelberg, pp. 457–485.

[24] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. https://eprint.iacr.org/2013/404.

[25] Beierle, C., Beyne, T., Felke, P., and Leander, G. Constructing and deconstructing intentional weaknesses in symmetric ciphers. In *Advances in Cryptology – CRYPTO 2022* (Cham, 2022), Y. Dodis and T. Shrimpton, Eds., Springer Nature Switzerland, pp. 748–778.

[26] Beierle, C., Canteaut, A., and Leander, G. Nonlinear approximations in cryptanalysis revisited. *IACR Trans. Symm. Cryptol. 2018*, 4 (2018), 80–101.

[27] Beierle, C., Canteaut, A., Leander, G., and Rotella, Y. Proving resistance against invariant attacks: How to choose the round constants. In *CRYPTO 2017, Part II* (Aug. 2017), J. Katz and H. Shacham, Eds., vol. 10402 of *LNCS*, Springer, Heidelberg, pp. 647–678.

[28] Beierle, C., Derbez, P., Leander, G., Leurent, G., Raddum, H., Rotella, Y., Rupprecht, D., and Stennes, L. Cryptanalysis of the GPRS encryption algorithms GEA-1 and GEA-2. In *EUROCRYPT 2021,*

Part II (Oct. 2021), A. Canteaut and F.-X. Standaert, Eds., vol. 12697 of *LNCS*, Springer, Heidelberg, pp. 155–183.

[29] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., and Sim, S. M. The SKINNY family of block ciphers and its low-latency variant MANTIS. In *CRYPTO 2016, Part II* (Aug. 2016), M. Robshaw and J. Katz, Eds., vol. 9815 of *LNCS*, Springer, Heidelberg, pp. 123–153.

[30] Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. A concrete security treatment of symmetric encryption. In *38th FOCS* (Oct. 1997), IEEE Computer Society Press, pp. 394–403.

[31] Bellare, M., Hoang, V. T., and Tessaro, S. Message-recovery attacks on Feistel-based format preserving encryption. In *ACM CCS 2016* (Oct. 2016), E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds., ACM Press, pp. 444–455.

[32] Bellare, M., Ristenpart, T., Rogaway, P., and Stegers, T. Format-preserving encryption. In *SAC 2009* (Aug. 2009), M. J. Jacobson Jr., V. Rijmen, and R. Safavi-Naini, Eds., vol. 5867 of *LNCS*, Springer, Heidelberg, pp. 295–312.

[33] Bernstein, D. J., Jeffery, S., Lange, T., and Meurer, A. Quantum algorithms for the subset-sum problem. In *International Workshop on Post-Quantum Cryptography* (2013), Springer, pp. 16–33.

[34] Bernstein, D. J., and Lange, T. Non-uniform cracks in the concrete: The power of free precomputation. In *ASIACRYPT 2013, Part II* (Dec. 2013), K. Sako and P. Sarkar, Eds., vol. 8270 of *LNCS*, Springer, Heidelberg, pp. 321–340.

[35] Bernstein, D. J., Lange, T., and Niederhagen, R. Dual EC: A standardized back door. In *The New Codebreakers* (2016), P. Y. A. Ryan, D. Naccache, and J. Quisquater, Eds., vol. 9100 of *LNCS*, Springer, pp. 256–281.

[36] Beullens, W., Beyne, T., Udovenko, A., and Vitto, G. Cryptanalysis of the Legendre PRF and generalizations. *IACR Trans. Symm. Cryptol. 2020*, 1 (2020), 313–330.

[37] Beyne, T. Block cipher invariants as eigenvectors of correlation matrices. In *ASIACRYPT 2018, Part I* (Dec. 2018), T. Peyrin and S. Galbraith, Eds., vol. 11272 of *LNCS*, Springer, Heidelberg, pp. 3–31.

[38] Beyne, T. Linear cryptanalysis in the weak key model. Master's thesis, KU Leuven, 2019.

[39] Beyne, T. Block cipher invariants as eigenvectors of correlation matrices. *Journal of Cryptology 33*, 3 (July 2020), 1156–1183.

[40] Beyne, T. A geometric approach to linear cryptanalysis. In *ASIACRYPT 2021, Part I* (Dec. 2021), M. Tibouchi and H. Wang, Eds., vol. 13090 of *LNCS*, Springer, Heidelberg, pp. 36–66.

[41] Beyne, T. Linear cryptanalysis of FF3-1 and FEA. In *CRYPTO 2021, Part I* (Virtual Event, Aug. 2021), T. Malkin and C. Peikert, Eds., vol. 12825 of *LNCS*, Springer, Heidelberg, pp. 41–69.

[42] Beyne, T., and Bilgin, B. Uniform first-order threshold implementations. In *SAC 2016* (Aug. 2016), R. Avanzi and H. M. Heys, Eds., vol. 10532 of *LNCS*, Springer, Heidelberg, pp. 79–98.

[43] Beyne, T., Canteaut, A., Dinur, I., Eichlseder, M., Leander, G., Leurent, G., Naya-Plasencia, M., Perrin, L., Sasaki, Y., Todo, Y., and Wiemer, F. Out of oddity - new cryptanalytic techniques against symmetric primitives optimized for integrity proof systems. In *CRYPTO 2020, Part III* (Aug. 2020), D. Micciancio and T. Ristenpart, Eds., vol. 12172 of *LNCS*, Springer, Heidelberg, pp. 299–328.

[44] Beyne, T., Canteaut, A., Leander, G., Naya-Plasencia, M., Perrin, L., and Wiemer, F. On the security of the Rescue hash function. Cryptology ePrint Archive, Report 2020/820, 2020. `https://eprint.iacr.org/2020/820`.

[45] Beyne, T., and Chen, Y. L. Provably secure reflection ciphers. In *Advances in Cryptology – CRYPTO 2022* (Cham, 2022), Y. Dodis and T. Shrimpton, Eds., Springer Nature Switzerland, pp. 234–263.

[46] Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Elephant v1. Submission to the NIST lightweight cryptography competition, 2019.

[47] Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Dumbo, Jumbo, and Delirium: Parallel authenticated encryption for the lightweight circus. *IACR Trans. Symm. Cryptol. 2020*, S1 (2020), 5–30.

[48] Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Status update on Elephant, 2020.

[49] Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Elephant v2. Submission to the NIST lightweight cryptography competition, 2021.

[50] Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Multi-user security of the Elephant v2 authenticated encryption mode. In

Selected Areas in Cryptography: 28th International Conference, Virtual Event, September 29–October 1, 2021, Revised Selected Papers (2022), Springer, pp. 155–178.

[51] Beyne, T., Dhooghe, S., Moradi, A., and Rezaei Shahmirzadi, A. Cryptanalysis of efficient masked ciphers: Applications to low latency. *IACR Transactions on Cryptographic Hardware and Embedded Systems 2022*, 1 (Nov. 2021), 679–721.

[52] Beyne, T., Dhooghe, S., Ranea, A., and Sijacic, D. A low-randomness second-order masked AES. In *Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers* (2021), R. AlTawy and A. Hülsing, Eds., vol. 13203 of *Lecture Notes in Computer Science*, Springer, pp. 87–110.

[53] Beyne, T., Dhooghe, S., and Zhang, Z. Cryptanalysis of masked ciphers: A not so random idea. In *ASIACRYPT 2020, Part I* (Dec. 2020), S. Moriai and H. Wang, Eds., vol. 12491 of *LNCS*, Springer, Heidelberg, pp. 817–850.

[54] Beyne, T., and Li, C. Cryptanalysis of the MALICIOUS framework. Cryptology ePrint Archive, Report 2020/1032, 2020. `https://eprint.iacr.org/2020/1032`.

[55] Beyne, T., and Liu, Y. Truncated differential attacks on contracting Feistel ciphers. *IACR Transactions on Symmetric Cryptology 2022*, 2 (Jun. 2022), 141–160.

[56] Beyne, T., and Rijmen, V. Differential cryptanalysis in the fixed-key model. In *Advances in Cryptology - CRYPTO 2022 - 42st Annual International Cryptology Conference, CRYPTO 2022, Santa Barabara, CA, USA, August 15-18, 2022, Proceedings, Part I* (2022), Y. Dodis and T. Shrimpton, Eds., Lecture Notes in Computer Science, Springer.

[57] Biham, E. New types of cryptanalytic attacks using related keys. *Journal of Cryptology 7*, 4 (Dec. 1994), 229–246.

[58] Biham, E., and Shamir, A. Differential cryptanalysis of DES-like cryptosystems. In *CRYPTO'90* (Aug. 1991), A. J. Menezes and S. A. Vanstone, Eds., vol. 537 of *LNCS*, Springer, Heidelberg, pp. 2–21.

[59] Biham, E., and Shamir, A. Differential cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer. In *CRYPTO'91* (Aug. 1992), J. Feigenbaum, Ed., vol. 576 of *LNCS*, Springer, Heidelberg, pp. 156–171.

[60] Biham, E., and Shamir, A. Differential cryptanalysis of the full 16-round DES. In *CRYPTO'92* (Aug. 1993), E. F. Brickell, Ed., vol. 740 of *LNCS*, Springer, Heidelberg, pp. 487–496.

[61] Bilgin, B., Gierlichs, B., Nikova, S., Nikov, V., and Rijmen, V. Higher-order threshold implementations. In *ASIACRYPT 2014, Part II* (Dec. 2014), P. Sarkar and T. Iwata, Eds., vol. 8874 of *LNCS*, Springer, Heidelberg, pp. 326–343.

[62] Bilgin, B., Gierlichs, B., Nikova, S., Nikov, V., and Rijmen, V. Trade-offs for threshold implementations illustrated on AES. *IEEE Trans. on CAD of Integrated Circuits and Systems 34*, 7 (2015), 1188–1200.

[63] Biryukov, A., De Cannière, C., and Quisquater, M. On multiple linear approximations. In *CRYPTO 2004* (Aug. 2004), M. Franklin, Ed., vol. 3152 of *LNCS*, Springer, Heidelberg, pp. 1–22.

[64] Biryukov, A., and Perrin, L. State of the art in lightweight symmetric cryptography. Cryptology ePrint Archive, Report 2017/511, 2017. `https://eprint.iacr.org/2017/511`.

[65] Biryukov, A., Roy, A., and Velichkov, V. Differential analysis of block ciphers SIMON and SPECK. In *FSE 2014* (Mar. 2015), C. Cid and C. Rechberger, Eds., vol. 8540 of *LNCS*, Springer, Heidelberg, pp. 546–570.

[66] Björck, Å., and Golub, G. H. Numerical methods for computing angles between linear subspaces. *Mathematics of computation 27*, 123 (1973), 579–594.

[67] Black, J., and Rogaway, P. Ciphers with arbitrary finite domains. In *CT-RSA 2002* (Feb. 2002), B. Preneel, Ed., vol. 2271 of *LNCS*, Springer, Heidelberg, pp. 114–130.

[68] Blondeau, C., and Gérard, B. On the data complexity of statistical attacks against block ciphers (full version). Cryptology ePrint Archive, Report 2009/064, 2009. `https://eprint.iacr.org/2009/064`.

[69] Boak, D. G. *A History of U.S. Communications Security*, vol. 1. National Security Agency, 1973.

[70] Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varici, K., and Verbauwhede, I. Spongent: A lightweight hash function. In *CHES 2011* (Sept. / Oct. 2011), B. Preneel and T. Takagi, Eds., vol. 6917 of *LNCS*, Springer, Heidelberg, pp. 312–325.

[71] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe, C. PRESENT: An ultra-lightweight block cipher. In *CHES 2007* (Sept. 2007), P. Paillier and I. Verbauwhede, Eds., vol. 4727 of *LNCS*, Springer, Heidelberg, pp. 450–466.

[72] Bogdanov, A., Leander, G., Nyberg, K., and Wang, M. Integral and multidimensional linear distinguishers with correlation zero. In *ASIACRYPT 2012* (Dec. 2012), X. Wang and K. Sako, Eds., vol. 7658 of *LNCS*, Springer, Heidelberg, pp. 244–261.

[73] Bogdanov, A., and Rijmen, V. Linear hulls with correlation zero and linear cryptanalysis of block ciphers. *Designs, Codes and Cryptography 70*, 3 (Mar 2014), 369–383.

[74] Bonnetain, X. Collisions on Feistel-MiMC and univariate GMiMC. Cryptology ePrint Archive, Report 2019/951, 2019. `https://eprint.iacr.org/2019/951`.

[75] Bordes, N., Daemen, J., Kuijsters, D., and Van Assche, G. Thinking outside the superbox. In *CRYPTO 2021, Part III* (Virtual Event, Aug. 2021), T. Malkin and C. Peikert, Eds., vol. 12827 of *LNCS*, Springer, Heidelberg, pp. 337–367.

[76] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E. B., Knežević, M., Knudsen, L. R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S. S., and Yalçın, T. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In *ASIACRYPT 2012* (Dec. 2012), X. Wang and K. Sako, Eds., vol. 7658 of *LNCS*, Springer, Heidelberg, pp. 208–225.

[77] Boura, C., and Canteaut, A. Zero-sum distinguishers for iterated permutations and application to Keccak-f and Hamsi-256. In *SAC 2010* (Aug. 2011), A. Biryukov, G. Gong, and D. R. Stinson, Eds., vol. 6544 of *LNCS*, Springer, Heidelberg, pp. 1–17.

[78] Boura, C., and Canteaut, A. On the influence of the algebraic degree of F^{-1} on the algebraic degree of $G \circ F$. *IEEE Transactions on Information Theory 59*, 1 (2012), 691–702.

[79] Boura, C., and Canteaut, A. Another view of the division property. In *CRYPTO 2016, Part I* (Aug. 2016), M. Robshaw and J. Katz, Eds., vol. 9814 of *LNCS*, Springer, Heidelberg, pp. 654–682.

[80] Brent, R. P., and Zimmermann, P. An $\mathcal{O}(M(n) \log n)$ algorithm for the Jacobi symbol. In *International Algorithmic Number Theory Symposium* (2010), Springer, pp. 83–95.

[81] Budaghyan, L., Carlet, C., and Pott, A. New classes of almost bent and almost perfect nonlinear polynomials. *IEEE Transactions on Information Theory 52*, 3 (2006), 1141–1152.

[82] Canteaut, A., Carpov, S., Fontaine, C., Lepoint, T., Naya-Plasencia, M., Paillier, P., and Sirdey, R. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. *Journal of Cryptology 31*, 3 (July 2018), 885–916.

[83] Canteaut, A., Lambooij, E., Neves, S., Rasoolzadeh, S., Sasaki, Y., and Stevens, M. Refined probability of differential characteristics including dependency between multiple rounds. *IACR Trans. Symm. Cryptol. 2017*, 2 (2017), 203–227.

[84] Canteaut, A., and Roué, J. On the behaviors of affine equivalent sboxes regarding differential and linear attacks. In *EUROCRYPT 2015, Part I* (Apr. 2015), E. Oswald and M. Fischlin, Eds., vol. 9056 of *LNCS*, Springer, Heidelberg, pp. 45–74.

[85] Canteaut, A., and Videau, M. Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis. In *EUROCRYPT 2002* (Apr. / May 2002), L. R. Knudsen, Ed., vol. 2332 of *LNCS*, Springer, Heidelberg, pp. 518–533.

[86] Carlet, C., Crama, Y., and Hammer, P. L. Boolean functions for cryptography and error correcting codes. *Boolean models and methods in mathematics, computer science, and engineering 2* (2010), 257–397.

[87] Carlet, C., and Guillot, P. A new representation of Boolean functions. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes* (1999), Springer, pp. 94–103.

[88] Chari, S., Jutla, C. S., Rao, J. R., and Rohatgi, P. Towards sound approaches to counteract power-analysis attacks. In *CRYPTO'99* (Aug. 1999), M. J. Wiener, Ed., vol. 1666 of *LNCS*, Springer, Heidelberg, pp. 398–412.

[89] Chen, Z., and Wang, X. Impossible differential cryptanalysis of Midori. Cryptology ePrint Archive, Report 2016/535, 2016. `https://eprint.iacr.org/2016/535`.

[90] Cho, J. Y. Linear cryptanalysis of reduced-round PRESENT. In *CT-RSA 2010* (Mar. 2010), J. Pieprzyk, Ed., vol. 5985 of *LNCS*, Springer, Heidelberg, pp. 302–317.

[91] Cho, J. Y., and Nyberg, K. Improved linear cryptanalysis of SMS4 block cipher. In *Workshop Record of SKEW 2011* (2011).

[92] Chose, P., Joux, A., and Mitton, M. Fast correlation attacks: An algorithmic point of view. In *EUROCRYPT 2002* (Apr. / May 2002), L. R. Knudsen, Ed., vol. 2332 of *LNCS*, Springer, Heidelberg, pp. 209–221.

[93] Cohn, P. M. *Algebra*, vol. 2. John Wiley & Sons, 1989.

[94] Collard, B., and Standaert, F.-X. A statistical saturation attack against the block cipher PRESENT. In *CT-RSA 2009* (Apr. 2009), M. Fischlin, Ed., vol. 5473 of *LNCS*, Springer, Heidelberg, pp. 195–210.

[95] Collard, B., Standaert, F.-X., and Quisquater, J.-J. Improving the time complexity of Matsui's linear cryptanalysis. In *ICISC 07* (Nov. 2007), K.-H. Nam and G. Rhee, Eds., vol. 4817 of *LNCS*, Springer, Heidelberg, pp. 77–88.

[96] Conrad, K. Bilinear forms. `https://kconrad.math.uconn.edu/blurbs/linmultialg/bilinearform.pdf`.

[97] Coppersmith, D. The Data Encryption Standard (DES) and its strength against attacks. *IBM journal of research and development 38*, 3 (1994), 243–250.

[98] Coron, J.-S., Prouff, E., Rivain, M., and Roche, T. Higher-order side channel security and mask refreshing. In *FSE 2013* (Mar. 2014), S. Moriai, Ed., vol. 8424 of *LNCS*, Springer, Heidelberg, pp. 410–424.

[99] Daemen, J. *Cipher and hash function design.* PhD thesis, KULeuven, 1995.

[100] Daemen, J. Spectral characterization of iterating lossy mappings. Cryptology ePrint Archive, Report 2016/090, 2016. `https://eprint.iacr.org/2016/090`.

[101] Daemen, J., Govaerts, R., and Vandewalle, J. Correlation matrices. In *FSE'94* (Dec. 1995), B. Preneel, Ed., vol. 1008 of *LNCS*, Springer, Heidelberg, pp. 275–285.

[102] Daemen, J., Knudsen, L. R., and Rijmen, V. The block cipher Square. In *FSE'97* (Jan. 1997), E. Biham, Ed., vol. 1267 of *LNCS*, Springer, Heidelberg, pp. 149–165.

[103] Daemen, J., and Rijmen, V. The block cipher BKSQ. In *CARDIS* (1998), vol. 1820, Springer, pp. 236–245.

[104] Daemen, J., and Rijmen, V. The wide trail design strategy. In *8th IMA International Conference on Cryptography and Coding* (Dec. 2001), B. Honary, Ed., vol. 2260 of *LNCS*, Springer, Heidelberg, pp. 222–238.

[105] Daemen, J., and Rijmen, V. Plateau characteristics. *IET Inf. Secur. 1*, 1 (2007), 11–17.

[106] Daemen, J., and Rijmen, V. Probability distributions of correlation and differentials in block ciphers. *J. Math. Cryptol. 1*, 3 (2007), 221–242.

[107] Daemen, J., and Rijmen, V. *The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition.* Information Security and Cryptography. Springer, 2020.

[108] Damgård, I. On the randomness of Legendre and Jacobi sequences. In *CRYPTO'88* (Aug. 1990), S. Goldwasser, Ed., vol. 403 of *LNCS*, Springer, Heidelberg, pp. 163–172.

[109] Davenport, H. On the distribution of quadratic residues (mod p). *Journal of the London Mathematical Society 1*, 1 (1931), 49–54.

[110] Davenport, H. On character sums in finite fields. *Acta Mathematica 71*, 1 (1939), 99–121.

[111] Davey, B. A., and Priestley, H. A. *Introduction to lattices and order.* Cambridge university press, 2002.

[112] De Cannière, C., and Rechberger, C. Finding SHA-1 characteristics: General results and applications. In *ASIACRYPT 2006* (Dec. 2006), X. Lai and K. Chen, Eds., vol. 4284 of *LNCS*, Springer, Heidelberg, pp. 1–20.

[113] De Meyer, L., Wegener, F., and Moradi, A. A note on masking generic Boolean functions. Cryptology ePrint Archive, Report 2019/1247, 2019. `https://eprint.iacr.org/2019/1247`.

[114] De Silva, V., and Lim, L.-H. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications 30*, 3 (2008), 1084–1127.

[115] Derbez, P., Fouque, P.-A., Jean, J., and Lambin, B. Variants of the AES key schedule for better truncated differential bounds. In *SAC 2018* (Aug. 2019), C. Cid and M. J. Jacobson Jr:, Eds., vol. 11349 of *LNCS*, Springer, Heidelberg, pp. 27–49.

[116] Dhooghe, S., Nikova, S., and Rijmen, V. Threshold implementations in the robust probing model. In *Proceedings of ACM Workshop on Theory of Implementation Security Workshop, TIS@CCS 2019, London, UK, November 11, 2019* (2019), B. Bilgin, S. Petkova-Nikova, and V. Rijmen, Eds., ACM, pp. 30–37.

[117] Diffie, W., and Hellman, M. E. Privacy and authentication: An introduction to cryptography. *Proceedings of the IEEE 67*, 3 (1979), 397–427.

[118] Diffie, W., and (translators), G. L. SMS4 encryption algorithm for wireless networks. Cryptology ePrint Archive, Report 2008/329, 2008. `https://eprint.iacr.org/2008/329`.

[119] Dinur, I. Improved differential cryptanalysis of round-reduced Speck. In *SAC 2014* (Aug. 2014), A. Joux and A. M. Youssef, Eds., vol. 8781 of *LNCS*, Springer, Heidelberg, pp. 147–164.

[120] Dinur, I., Liu, Y., Meier, W., and Wang, Q. Optimized interpolation attacks on LowMC. In *ASIACRYPT 2015, Part II* (Nov. / Dec. 2015), T. Iwata and J. H. Cheon, Eds., vol. 9453 of *LNCS*, Springer, Heidelberg, pp. 535–560.

[121] Dinur, I., and Shamir, A. Cube attacks on tweakable black box polynomials. In *EUROCRYPT 2009* (Apr. 2009), A. Joux, Ed., vol. 5479 of *LNCS*, Springer, Heidelberg, pp. 278–299.

[122] Dobraunig, C., Eichlseder, M., Grassi, L., Lallemand, V., Leander, G., List, E., Mendel, F., and Rechberger, C. Rasta: A cipher with low ANDdepth and few ANDs per bit. In *CRYPTO 2018, Part I* (Aug. 2018), H. Shacham and A. Boldyreva, Eds., vol. 10991 of *LNCS*, Springer, Heidelberg, pp. 662–692.

[123] Dobraunig, C., Eichlseder, M., Kales, D., and Mendel, F. Practical key-recovery attack on MANTIS5. *IACR Trans. Symm. Cryptol. 2016*, 2 (2016), 248–260.

[124] Dobraunig, C., Eichlseder, M., and Mendel, F. Higher-order cryptanalysis of LowMC. In *ICISC 15* (Nov. 2016), S. Kwon and A. Yun, Eds., vol. 9558 of *LNCS*, Springer, Heidelberg, pp. 87–101.

[125] Dobraunig, C., Eichlseder, M., Mendel, F., and Schläffer, M. Ascon v1.2. Submission to the NIST lightweight cryptography competition, 2019.

[126] Dunkelman, O., Kumar, A., Lambooij, E., and Sanadhya, S. K. Cryptanalysis of feistel-based format-preserving encryption. Cryptology ePrint Archive, Report 2020/1311, 2020. `https://eprint.iacr.org/2020/1311`.

[127] Durak, F. B., and Vaudenay, S. Breaking the FF3 format-preserving encryption standard over small domains. In *CRYPTO 2017, Part II*

(Aug. 2017), J. Katz and H. Shacham, Eds., vol. 10402 of *LNCS*, Springer, Heidelberg, pp. 679–707.

[128] Dworkin, M. SHA-3 standard: Permutation-based hash and extendable-output functions, 2015.

[129] Dworkin, M. Recommendation for block cipher modes of operation: methods for format-preserving encryption. *NIST Special Publication 800 38Gr1* (February 2019).

[130] Dworkin, M. J. Recommendation for block cipher modes of operation: Galois/Counter mode (GCM) and GMAC, 2007.

[131] Dziembowski, S., Faust, S., Herold, G., Journault, A., Masny, D., and Standaert, F.-X. Towards sound fresh re-keying with hard (physical) learning problems. In *CRYPTO 2016, Part II* (Aug. 2016), M. Robshaw and J. Katz, Eds., vol. 9815 of *LNCS*, Springer, Heidelberg, pp. 272–301.

[132] Eckart, C., and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika 1*, 3 (1936), 211–218.

[133] Eichlseder, M., and Kales, D. Clustering related-tweak characteristics: Application to MANTIS-6. *IACR Trans. Symm. Cryptol. 2018*, 2 (2018), 111–132.

[134] Etrog, J., and Robshaw, M. J. B. The cryptanalysis of reduced-round SMS4. In *SAC 2008* (Aug. 2009), R. M. Avanzi, L. Keliher, and F. Sica, Eds., vol. 5381 of *LNCS*, Springer, Heidelberg, pp. 51–65.

[135] Faugère, J.-C., Gianni, P., Lazard, D., and Mora, T. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation 16*, 4 (1993), 329–344.

[136] Faugère, J.-C., and Mou, C. Fast algorithm for change of ordering of zero-dimensional Gröbner bases with sparse multiplication matrices. In *Proceedings of the 36th international symposium on Symbolic and algebraic computation* (2011), pp. 115–122.

[137] Faugère, J.-C., and Perret, L. Algebraic attacks against STARK-friendly ciphers. Personal communication, 2019.

[138] Faust, S., Grosso, V., Pozo, S. M. D., Paglialonga, C., and Standaert, F.-X. Composable masking schemes in the presence of physical defaults & the robust probing model. *IACR TCHES 2018*, 3 (2018), 89–120.

[139] Faust, S., Paglialonga, C., and Schneider, T. Amortizing randomness complexity in private circuits. In *ASIACRYPT 2017, Part I* (Dec. 2017), T. Takagi and T. Peyrin, Eds., vol. 10624 of *LNCS*, Springer, Heidelberg, pp. 781–810.

[140] Feist, D. Cryptanalyzing the Legendre PRF. CRYPTO rump session talk, August 2019.

[141] Feist, D. Legendre pseudo-random function. `https://legendreprf.org`, 2019. Accessed: 18/11/2019.

[142] Feller, W. *An introduction to probability theory and its applicatons*, vol. 1. John Wiley & Sons, 1950.

[143] Filiol, E. BSEA-1 – A stream cipher backdooring technique. *arXiv preprint arXiv:1903.11063* (2019).

[144] Friedman, W. F. Elements of cryptanalysis, training pamphlet no. 3. Prepared in the office of the Chief Signal Officer.

[145] Goldwasser, S., and Micali, S. Probabilistic encryption. *Journal of Computer and System Sciences 28*, 2 (1984), 270–299.

[146] Gorodilova, A., Tokareva, N., Agievich, S., Beterov, I., Beyne, T., Budaghyan, L., Carlet, C., Dhooghe, S., Idrisova, V., Kolomeec, N., Kutsenko, A., Malygina, E., Mouha, N., Pudovkina, M., Sica, F., and Udovenko, A. An overview of the eight international olympiad in cryptography "non-stop university crypto", 2022.

[147] Goubin, L., and Patarin, J. DES and differential power analysis (the "duplication" method). In *CHES'99* (Aug. 1999), Çetin Kaya. Koç and C. Paar, Eds., vol. 1717 of *LNCS*, Springer, Heidelberg, pp. 158–172.

[148] Granboulan, L., Levieil, É., and Piret, G. Pseudorandom permutation families over Abelian groups. In *FSE 2006* (Mar. 2006), M. J. B. Robshaw, Ed., vol. 4047 of *LNCS*, Springer, Heidelberg, pp. 57–77.

[149] Grassi, L., Kales, D., Khovratovich, D., Roy, A., Rechberger, C., and Schofnegger, M. Starkad and Poseidon: New hash functions for zero knowledge proof systems. Cryptology ePrint Archive, Report 2019/458, 2019. `https://eprint.iacr.org/2019/458`.

[150] Grassi, L., Khovratovich, D., Rechberger, C., Roy, A., and Schofnegger, M. Poseidon: A new hash function for zero-knowledge

proof systems. In *USENIX Security 2021* (Aug. 2021), M. Bailey and R. Greenstadt, Eds., USENIX Association, pp. 519–535.

[151] Grassi, L., Lüftenegger, R., Rechberger, C., Rotaru, D., and Schofnegger, M. On a generalization of substitution-permutation networks: The HADES design strategy. Cryptology ePrint Archive, Report 2019/1107, 2019. `https://eprint.iacr.org/2019/1107`.

[152] Grassi, L., Lüftenegger, R., Rechberger, C., Rotaru, D., and Schofnegger, M. On a generalization of substitution-permutation networks: The HADES design strategy. In *EUROCRYPT 2020, Part II* (May 2020), A. Canteaut and Y. Ishai, Eds., vol. 12106 of *LNCS*, Springer, Heidelberg, pp. 674–704.

[153] Grassi, L., Rechberger, C., Rotaru, D., Scholl, P., and Smart, N. P. MPC-friendly symmetric key primitives. In *ACM CCS 2016* (Oct. 2016), E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds., ACM Press, pp. 430–443.

[154] Guo, J., Jean, J., Nikolic, I., Qiao, K., Sasaki, Y., and Sim, S. M. Invariant subspace attack against Midori64 and the resistance criteria for S-box designs. *IACR Trans. Symm. Cryptol. 2016*, 1 (2016), 33–56.

[155] Guo, J., Jean, J., Nikolic, I., and Sasaki, Y. Meet-in-the-middle attacks on classes of contracting and expanding Feistel constructions. *IACR Trans. Symm. Cryptol. 2016*, 2 (2016), 307–337.

[156] Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M. J. B. The LED block cipher. In *CHES 2011* (Sept. / Oct. 2011), B. Preneel and T. Takagi, Eds., vol. 6917 of *LNCS*, Springer, Heidelberg, pp. 326–341.

[157] Halmos, P. R. *Finite dimensional vector spaces*, 1 ed., vol. 8 of *Undergraduate Texts in Mathematics*. Springer-Verlag New York, 1958.

[158] Hao, Y., Leander, G., Meier, W., Todo, Y., and Wang, Q. Modeling for three-subset division property without unknown subset - improved cube attacks against Trivium and Grain-128AEAD. In *EUROCRYPT 2020, Part I* (May 2020), A. Canteaut and Y. Ishai, Eds., vol. 12105 of *LNCS*, Springer, Heidelberg, pp. 466–495.

[159] Harpes, C., Kramer, G. G., and Massey, J. L. A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma. In *EUROCRYPT'95* (May 1995), L. C. Guillou and J.-J. Quisquater, Eds., vol. 921 of *LNCS*, Springer, Heidelberg, pp. 24–38.

[160] Harpes, C., and Massey, J. L. Partitioning cryptanalysis. In *FSE'97* (Jan. 1997), E. Biham, Ed., vol. 1267 of *LNCS*, Springer, Heidelberg, pp. 13–27.

[161] Hebborn, P., Leander, G., and Udovenko, A. Mathematical aspects of division property. Cryptology ePrint Archive, Paper 2022/736, 2022. `https://eprint.iacr.org/2022/736`.

[162] Hermelin, M., Cho, J. Y., and Nyberg, K. Multidimensional linear cryptanalysis of reduced round Serpent. In *ACISP 08* (July 2008), Y. Mu, W. Susilo, and J. Seberry, Eds., vol. 5107 of *LNCS*, Springer, Heidelberg, pp. 203–215.

[163] Heys, H. M. Key dependency of differentials: Experiments in the differential cryptanalysis of block ciphers using small S-boxes. ePrint, Report 2020/1349, 2020.

[164] Hoang, V. T., Miller, D., and Trieu, N. Attacks only get better: How to break FF3 on large domains. In *EUROCRYPT 2019, Part II* (May 2019), Y. Ishai and V. Rijmen, Eds., vol. 11477 of *LNCS*, Springer, Heidelberg, pp. 85–116.

[165] Hoang, V. T., Tessaro, S., and Trieu, N. The curse of small domains: New attacks on format-preserving encryption. In *CRYPTO 2018, Part I* (Aug. 2018), H. Shacham and A. Boldyreva, Eds., vol. 10991 of *LNCS*, Springer, Heidelberg, pp. 221–251.

[166] Hu, K., Sun, S., Wang, M., and Wang, Q. An algebraic formulation of the division property: Revisiting degree evaluations, cube attacks, and key-independent sums. In *ASIACRYPT 2020, Part I* (Dec. 2020), S. Moriai and H. Wang, Eds., vol. 12491 of *LNCS*, Springer, Heidelberg, pp. 446–476.

[167] Ishai, Y., Sahai, A., and Wagner, D. Private circuits: Securing hardware against probing attacks. In *CRYPTO 2003* (Aug. 2003), D. Boneh, Ed., vol. 2729 of *LNCS*, Springer, Heidelberg, pp. 463–481.

[168] Jacobsthal, E. E. *Anwendungen einer Formel aus der Theorie der quadratischen Reste*. PhD thesis, Friedrich-Wilhelms Universität zu Berlin, 1906.

[169] Jean, J., Nikolic, I., and Peyrin, T. Tweaks and keys for block ciphers: The TWEAKEY framework. In *ASIACRYPT 2014, Part II* (Dec. 2014), P. Sarkar and T. Iwata, Eds., vol. 8874 of *LNCS*, Springer, Heidelberg, pp. 274–288.

[170] Johnson, T. R. *American Cryptology during the Cold War, 1945-1989: Book III: Retrenchment and Reform, 1972-1980.* National Security Agency, 1998.

[171] Jordan, C. Essai sur la géométrie à n dimensions. *Bulletin de la Société mathématique de France 3* (1875), 103–174.

[172] Kahn, D. *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet.* Simon and Schuster, 1996.

[173] Kaliski Jr., B. S., and Robshaw, M. J. B. Linear cryptanalysis using multiple approximations. In *CRYPTO'94* (Aug. 1994), Y. Desmedt, Ed., vol. 839 of *LNCS*, Springer, Heidelberg, pp. 26–39.

[174] Kaluđerović, N., Kleinjung, T., and Kostic, D. Improved key recovery on the Legendre PRF. Cryptology ePrint Archive, Report 2020/098, 2020. `https://eprint.iacr.org/2020/098`.

[175] Keliher, L., and Sui, J. Exact maximum expected differential and linear probability for two-round Advanced Encryption Standard. *IET Information Security 1*, 2 (2007), 53–57.

[176] Keller, N., and Rosemarin, A. Mind the middle layer: The HADES design strategy revisited. In *EUROCRYPT 2021, Part II* (Oct. 2021), A. Canteaut and F.-X. Standaert, Eds., vol. 12697 of *LNCS*, Springer, Heidelberg, pp. 35–63.

[177] Khoo, K., Lee, E., Peyrin, T., and Sim, S. M. Human-readable proof of the related-key security of AES-128. *IACR Trans. Symm. Cryptol. 2017*, 2 (2017), 59–83.

[178] Khovratovich, D. Key recovery attacks on the Legendre PRFs within the birthday bound. Cryptology ePrint Archive, Report 2019/862, 2019. `https://eprint.iacr.org/2019/862`.

[179] Kim, T., Kim, J., Hong, S., and Sung, J. Linear and differential cryptanalysis of reduced SMS4 block cipher. Cryptology ePrint Archive, Report 2008/281, 2008. `https://eprint.iacr.org/2008/281`.

[180] Knudsen, L. R. Iterative characteristics of DES and s^2-DES. In *CRYPTO'92* (Aug. 1993), E. F. Brickell, Ed., vol. 740 of *LNCS*, Springer, Heidelberg, pp. 497–511.

[181] Knudsen, L. R. Truncated and higher order differentials. In *FSE'94* (Dec. 1995), B. Preneel, Ed., vol. 1008 of *LNCS*, Springer, Heidelberg, pp. 196–211.

[182] Knudsen, L. R., and Rijmen, V. Known-key distinguishers for some block ciphers. In *ASIACRYPT 2007* (Dec. 2007), K. Kurosawa, Ed., vol. 4833 of *LNCS*, Springer, Heidelberg, pp. 315–324.

[183] Knudsen, L. R., and Robshaw, M. J. B. Non-linear approximations in linear cryptanalysis. In *EUROCRYPT'96* (May 1996), U. M. Maurer, Ed., vol. 1070 of *LNCS*, Springer, Heidelberg, pp. 224–236.

[184] Knudsen, L. R., and Wagner, D. Integral cryptanalysis. In *FSE 2002* (Feb. 2002), J. Daemen and V. Rijmen, Eds., vol. 2365 of *LNCS*, Springer, Heidelberg, pp. 112–127.

[185] Koblitz, N. *p-adic numbers, p-adic analysis, and zeta-functions*, vol. 58 of *Graduate Texts in Mathematics*. Springer-Verlag New York, 2012.

[186] Koblitz, N., and Menezes, A. Another look at HMAC. Cryptology ePrint Archive, Report 2012/074, 2012. `https://eprint.iacr.org/2012/074`.

[187] Kocher, P. C., Jaffe, J., and Jun, B. Differential power analysis. In *CRYPTO'99* (Aug. 1999), M. J. Wiener, Ed., vol. 1666 of *LNCS*, Springer, Heidelberg, pp. 388–397.

[188] Kong, X., Wang, W., and Xu, Q. Improved rectangle attack on SMS4 reduced to 18 rounds. In *Ninth International Conference on Computational Intelligence and Security, CIS 2013, Emei Mountain, Sichan Province, China, December 14-15, 2013* (2013), IEEE Computer Society, pp. 575–578.

[189] Kutzner, S., Nguyen, P. H., Poschmann, A., and Wang, H. On 3-share threshold implementations for 4-bit S-boxes. In *COSADE 2013* (Mar. 2013), E. Prouff, Ed., vol. 7864 of *LNCS*, Springer, Heidelberg, pp. 99–113.

[190] Lai, X. Higher order derivatives and differential cryptanalysis. *Communications and Cryptography: Two Sides of One Tapestry* (1994), 227–233.

[191] Lai, X., Massey, J. L., and Murphy, S. Markov ciphers and differential cryptanalysis. In *EUROCRYPT'91* (Apr. 1991), D. W. Davies, Ed., vol. 547 of *LNCS*, Springer, Heidelberg, pp. 17–38.

[192] Lang, S. *Cyclotomic fields*. Springer New York, New York, NY, 1994, pp. 71–98.

[193] Langley, A., Chang, W., Mavrogiannopoulos, N., Strombergson, J., and Josefsson, S. ChaCha20-Poly1305 cipher suites for transport layer security (TLS), 2016.

[194] Le, T. V., Sparr, R., Wernsdorf, R., and Desmedt, Y. Complementation-like and cyclic properties of AES round functions. In *Advanced Encryption Standard - AES, 4th International Conference* (2004), H. Dobbertin, V. Rijmen, and A. Sowa, Eds., vol. 3373 of *LNCS*, Springer, pp. 128–141.

[195] Leander, G. On linear hulls, statistical saturation attacks, PRESENT and a cryptanalysis of PUFFIN. In *EUROCRYPT 2011* (May 2011), K. G. Paterson, Ed., vol. 6632 of *LNCS*, Springer, Heidelberg, pp. 303–322.

[196] Leander, G., Abdelraheem, M. A., AlKhzaimi, H., and Zenner, E. A cryptanalysis of PRINTcipher: The invariant subspace attack. In *CRYPTO 2011* (Aug. 2011), P. Rogaway, Ed., vol. 6841 of *LNCS*, Springer, Heidelberg, pp. 206–221.

[197] Lee, H., Kim, S., Kang, H., Hong, D., Sung, J., and Hong, S. Calculating the approximate probability of differentials for ARX-based cipher using SAT solver. *Journal of the Korea Institute of Information Security & Cryptology 28*, 1 (2018), 15–24.

[198] Lee, J.-K., Koo, B., Roh, D., Kim, W.-H., and Kwon, D. Format-preserving encryption algorithms using families of tweakable blockciphers. In *Information Security and Cryptology - ICISC 2014* (Cham, 2015), J. Lee and J. Kim, Eds., Springer International Publishing, pp. 132–159.

[199] Leurent, G. Analysis of differential attacks in ARX constructions. In *ASIACRYPT 2012* (Dec. 2012), X. Wang and K. Sako, Eds., vol. 7658 of *LNCS*, Springer, Heidelberg, pp. 226–243.

[200] Levy, S. *Crypto: How the code rebels beat the government–saving privacy in the digital age*. Penguin, 2001.

[201] Lin, L., and Wu, W. Meet-in-the-middle attacks on reduced-round Midori64. *IACR Trans. Symm. Cryptol. 2017*, 1 (2017), 215–239.

[202] Liu, F., Isobe, T., and Meier, W. Cryptanalysis of full LowMC and LowMC-M with algebraic techniques. In *CRYPTO 2021, Part III* (Virtual Event, Aug. 2021), T. Malkin and C. Peikert, Eds., vol. 12827 of *LNCS*, Springer, Heidelberg, pp. 368–401.

[203] Liu, M., and Chen, J. Improved linear attacks on the chinese block cipher standard. *J. Comput. Sci. Technol. 29*, 6 (2014), 1123–1133.

[204] Liu, Y., Liang, H., Wang, W., and Wang, M. New linear cryptanalysis of chinese commercial block cipher standard SM4. *Secur. Commun. Networks 2017* (2017), 1461520:1–1461520:10.

[205] Liu, Y., Zhang, W., Sun, B., Rijmen, V., Liu, G., Li, C., Fu, S., and Cao, M. The phantom of differential characteristics. *Des. Codes Cryptogr. 88*, 11 (2020), 2289–2311.

[206] Liu, Z., Gu, D., and Zhang, J. Multiple linear cryptanalysis of reduced-round SMS4 block cipher. *Chinese Journal of Electronics 19-3* (2010), 389–393.

[207] Lu, J. Attacking reduced-round versions of the SMS4 block cipher in the Chinese WAPI standard. In *ICICS 07* (Dec. 2008), S. Qing, H. Imai, and G. Wang, Eds., vol. 4861 of *LNCS*, Springer, Heidelberg, pp. 306–318.

[208] Luby, M., and Rackoff, C. How to construct pseudo-random permutations from pseudo-random functions (abstract). In *CRYPTO'85* (Aug. 1986), H. C. Williams, Ed., vol. 218 of *LNCS*, Springer, Heidelberg, p. 447.

[209] Luykx, A. *The Design and Analysis of Message Authentication and Authenticated Encryption Schemes*. PhD thesis, KULeuven, 2016.

[210] Luykx, A., Mennink, B., and Paterson, K. G. Analyzing multi-key security degradation. In *ASIACRYPT 2017, Part II* (Dec. 2017), T. Takagi and T. Peyrin, Eds., vol. 10625 of *LNCS*, Springer, Heidelberg, pp. 575–605.

[211] Luykx, A., Preneel, B., Tischhauser, E., and Yasuda, K. A MAC mode for lightweight block ciphers. In *FSE 2016* (Mar. 2016), T. Peyrin, Ed., vol. 9783 of *LNCS*, Springer, Heidelberg, pp. 43–59.

[212] Macaulay, F. S. Some formulae in elimination. *Proceedings of the London Mathematical Society 1*, 1 (1902), 3–27.

[213] Mangard, S., Pramstaller, N., and Oswald, E. Successfully attacking masked AES hardware implementations. In *CHES 2005* (Aug. / Sept. 2005), J. R. Rao and B. Sunar, Eds., vol. 3659 of *LNCS*, Springer, Heidelberg, pp. 157–171.

[214] Matsui, M. The first experimental cryptanalysis of the data encryption standard. In *CRYPTO'94* (Aug. 1994), Y. Desmedt, Ed., vol. 839 of *LNCS*, Springer, Heidelberg, pp. 1–11.

[215] Matsui, M. Linear cryptanalysis method for DES cipher. In *EUROCRYPT'93* (May 1994), T. Helleseth, Ed., vol. 765 of *LNCS*, Springer, Heidelberg, pp. 386–397.

[216] Méaux, P., Journault, A., Standaert, F.-X., and Carlet, C. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In *EUROCRYPT 2016, Part I* (May 2016), M. Fischlin and J.-S. Coron, Eds., vol. 9665 of *LNCS*, Springer, Heidelberg, pp. 311–343.

[217] Mendel, F., Rijmen, V., Toz, D., and Varici, K. Differential analysis of the LED block cipher. In *ASIACRYPT 2012* (Dec. 2012), X. Wang and K. Sako, Eds., vol. 7658 of *LNCS*, Springer, Heidelberg, pp. 190–207.

[218] Moos, T., Moradi, A., Schneider, T., and Standaert, F.-X. Glitch-resistant masking revisited. *IACR TCHES 2019*, 2 (2019), 256–292.

[219] Moradi, A., and Wild, A. Assessment of hiding the higher-order leakages in hardware - what are the achievements versus overheads? In *CHES 2015* (Sept. 2015), T. Güneysu and H. Handschuh, Eds., vol. 9293 of *LNCS*, Springer, Heidelberg, pp. 453–474.

[220] Morita, H., Ohta, K., and Miyaguchi, S. A switching closure test to analyze cryptosystems. In *CRYPTO'91* (Aug. 1992), J. Feigenbaum, Ed., vol. 576 of *LNCS*, Springer, Heidelberg, pp. 183–193.

[221] Naya-Plasencia, M., and Schrottenloher, A. Optimal merging in quantum k-xor and k-xor-sum algorithms. In *EUROCRYPT 2020, Part II* (May 2020), A. Canteaut and Y. Ishai, Eds., vol. 12106 of *LNCS*, Springer, Heidelberg, pp. 311–340.

[222] Niemetz, A., Preiner, M., and Biere, A. Boolector 2.0. *J. Satisf. Boolean Model. Comput. 9*, 1 (2014), 53–58.

[223] Nikova, S., Rechberger, C., and Rijmen, V. Threshold implementations against side-channel attacks and glitches. In *ICICS 06* (Dec. 2006), P. Ning, S. Qing, and N. Li, Eds., vol. 4307 of *LNCS*, Springer, Heidelberg, pp. 529–545.

[224] Nyberg, K. Linear approximation of block ciphers (rump session). In *EUROCRYPT'94* (May 1995), A. D. Santis, Ed., vol. 950 of *LNCS*, Springer, Heidelberg, pp. 439–444.

[225] Nyberg, K. Generalized Feistel networks. In *ASIACRYPT'96* (Nov. 1996), K. Kim and T. Matsumoto, Eds., vol. 1163 of *LNCS*, Springer, Heidelberg, pp. 91–104.

[226] Park, S., Sung, S. H., Lee, S., and Lim, J. Improving the upper bound on the maximum differential and the maximum linear hull probability for SPN structures and AES. In *FSE 2003* (Feb. 2003), T. Johansson, Ed., vol. 2887 of *LNCS*, Springer, Heidelberg, pp. 247–260.

[227] Patarin, J. New results on pseudorandom permutation generators based on the DES scheme. In *CRYPTO'91* (Aug. 1992), J. Feigenbaum, Ed., vol. 576 of *LNCS*, Springer, Heidelberg, pp. 301–312.

[228] Patarin, J. Generic attacks on Feistel schemes. In *ASIACRYPT 2001* (Dec. 2001), C. Boyd, Ed., vol. 2248 of *LNCS*, Springer, Heidelberg, pp. 222–238.

[229] Patarin, J. Security of random Feistel schemes with 5 or more rounds. In *CRYPTO 2004* (Aug. 2004), M. Franklin, Ed., vol. 3152 of *LNCS*, Springer, Heidelberg, pp. 106–122.

[230] Patarin, J., Nachef, V., and Berbain, C. Generic attacks on unbalanced Feistel schemes with contracting functions. In *ASIACRYPT 2006* (Dec. 2006), X. Lai and K. Chen, Eds., vol. 4284 of *LNCS*, Springer, Heidelberg, pp. 396–411.

[231] Paterson, K. G. Imprimitive permutation groups and trapdoors in iterated block ciphers. In *FSE'99* (Mar. 1999), L. R. Knudsen, Ed., vol. 1636 of *LNCS*, Springer, Heidelberg, pp. 201–214.

[232] Perlroth, N., Larson, J., and Shane, S. N.S.A. able to foil basic safeguards of privacy on web. International New York Times `https://www.nytimes.com/2013/09/06/us/nsa-foils-much-internet-encryption.html` (accessed September 30, 2021), 2013.

[233] Peyrin, T., and Seurin, Y. Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers. In *CRYPTO 2016, Part I* (Aug. 2016), M. Robshaw and J. Katz, Eds., vol. 9814 of *LNCS*, Springer, Heidelberg, pp. 33–63.

[234] Peyrin, T., and Wang, H. The MALICIOUS framework: Embedding backdoors into tweakable block ciphers. In *CRYPTO 2020, Part III* (Aug. 2020), D. Micciancio and T. Ristenpart, Eds., vol. 12172 of *LNCS*, Springer, Heidelberg, pp. 249–278.

[235] Posteuca, R., and Ashur, T. How to backdoor a cipher. Cryptology ePrint Archive, Report 2021/442, 2021. `https://eprint.iacr.org/2021/442`.

[236] Prouff, E., and Rivain, M. Masking against side-channel attacks: A formal security proof. In *EUROCRYPT 2013* (May 2013), T. Johansson and P. Q. Nguyen, Eds., vol. 7881 of *LNCS*, Springer, Heidelberg, pp. 142–159.

[237] PUB FIPS. 46: Data Encryption Standard (DES). *US Department of Commerce, National Bureau of Standards* (1977).

[238] Rechberger, C., Soleimany, H., and Tiessen, T. Cryptanalysis of low-data instances of full LowMCv2. *IACR Trans. Symm. Cryptol. 2018*, 3 (2018), 163–181.

[239] Reparaz, O. A note on the security of higher-order threshold implementations. Cryptology ePrint Archive, Report 2015/001, 2015. `https://eprint.iacr.org/2015/001`.

[240] Reparaz, O., Bilgin, B., Nikova, S., Gierlichs, B., and Verbauwhede, I. Consolidating masking schemes. In *CRYPTO 2015, Part I* (Aug. 2015), R. Gennaro and M. J. B. Robshaw, Eds., vol. 9215 of *LNCS*, Springer, Heidelberg, pp. 764–783.

[241] Rijmen, V. *Cryptanalysis and design of iterated block ciphers*. PhD thesis, KULeuven, 1997.

[242] Rijmen, V. Practical-titled attack on AES-128 using chosen-text relations. Cryptology ePrint Archive, Report 2010/337, 2010. `https://eprint.iacr.org/2010/337`.

[243] Rijmen, V., Daemen, J., Preneel, B., Bossalaers, A., and De Win, E. The cipher SHARK. In *FSE'96* (Feb. 1996), D. Gollmann, Ed., vol. 1039 of *LNCS*, Springer, Heidelberg, pp. 99–111.

[244] Rijmen, V., and Preneel, B. A family of trapdoor ciphers. In *FSE'97* (Jan. 1997), E. Biham, Ed., vol. 1267 of *LNCS*, Springer, Heidelberg, pp. 139–148.

[245] Rota, G.-C. On the foundations of combinatorial theory I. Theory of Möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 2*, 4 (1964), 340–368.

[246] Sajadieh, M., Dakhilalian, M., Mala, H., and Omoomi, B. On construction of involutory MDS matrices from Vandermonde matrices in GF(2^q). *Designs, Codes and Cryptography 64*, 3 (Sep 2012), 287–308.

[247] Schneier, B., and Kelsey, J. Unbalanced Feistel networks and block cipher design. In *FSE'96* (Feb. 1996), D. Gollmann, Ed., vol. 1039 of *LNCS*, Springer, Heidelberg, pp. 121–144.

[248] Schulte-Geers, E. On CCZ-equivalence of addition mod 2^n. *Des. Codes Cryptogr. 66*, 1-3 (2013), 111–127.

[249] Selçuk, A. A. On probability of success in linear and differential cryptanalysis. *Journal of Cryptology 21*, 1 (Jan. 2008), 131–147.

[250] Shannon, C. E. A mathematical theory of communication. *Bell Systems Technical Journal 27*, 3 (1948), 379–423.

[251] Shannon, C. E. Communication theory of secrecy systems. *Bell Systems Technical Journal 28*, 4 (1949), 656–715.

[252] Shi, T., Wang, W., and Xu, Q. Improved impossible differential cryptanalysis of SMS4. In *Eighth International Conference on Computational Intelligence and Security, CIS 2012, Guangzhou, China, November 17-18, 2012* (2012), IEEE Computer Society, pp. 492–496.

[253] Smith, S. T. Optimization techniques on Riemannian manifolds. *Fields institute communications 3*, 3 (1994), 113–135.

[254] Song, H., and Gao, H. Multiple differential attack on 21-round sms4 (in Chinese). *Journal of Cryptologic Research 2016*, 3(6) (2016), 584–595.

[255] Song, L., Huang, Z., and Yang, Q. Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. In *ACISP 16, Part II* (July 2016), J. K. Liu and R. Steinfeld, Eds., vol. 9723 of *LNCS*, Springer, Heidelberg, pp. 379–394.

[256] Steinberg, B. *Representation theory of finite monoids.* Springer Cham, 2016.

[257] Su, B., Wu, W., and Zhang, W. Security of the SMS4 block cipher against differential cryptanalysis. *J. Comput. Sci. Technol. 26*, 1 (2011), 130–138.

[258] Sun, B., Liu, Z., Rijmen, V., Li, R., Cheng, L., Wang, Q., AlKhzaimi, H., and Li, C. Links among impossible differential, integral and zero correlation linear cryptanalysis. In *CRYPTO 2015, Part I* (Aug. 2015), R. Gennaro and M. J. B. Robshaw, Eds., vol. 9215 of *LNCS*, Springer, Heidelberg, pp. 95–115.

[259] Sun, L., Wang, W., and Wang(66), M. More accurate differential properties of LED64 and Midori64. *IACR Trans. Symm. Cryptol. 2018*, 3 (2018), 93–123.

[260] Tao, T. Cycles of a random permutation and irreducible factors of a random polynomial. https://terrytao.wordpress.com/2015/07/15/cycles-of-a-random-permutation-and-irreducible-factors-of-a-random-polynomial/, 2015. Accessed: 2019-11-18.

[261] Tardy-Corfdir, A., and Gilbert, H. A known plaintext attack of FEAL-4 and FEAL-6. In *CRYPTO'91* (Aug. 1992), J. Feigenbaum, Ed., vol. 576 of *LNCS*, Springer, Heidelberg, pp. 172–181.

[262] Terras, A. *Fourier analysis on finite groups and applications*. Cambridge University Press, 1999.

[263] Todo, Y. Integral cryptanalysis on full MISTY1. In *CRYPTO 2015, Part I* (Aug. 2015), R. Gennaro and M. J. B. Robshaw, Eds., vol. 9215 of *LNCS*, Springer, Heidelberg, pp. 413–432.

[264] Todo, Y. Structural evaluation by generalized integral property. In *EUROCRYPT 2015, Part I* (Apr. 2015), E. Oswald and M. Fischlin, Eds., vol. 9056 of *LNCS*, Springer, Heidelberg, pp. 287–314.

[265] Todo, Y., and Aoki, K. FFT key recovery for integral attack. In *CANS 14* (Oct. 2014), D. Gritzalis, A. Kiayias, and I. G. Askoxylakis, Eds., vol. 8813 of *LNCS*, Springer, Heidelberg, pp. 64–81.

[266] Todo, Y., Leander, G., and Sasaki, Y. Nonlinear invariant attack - practical attack on full SCREAM, iSCREAM, and Midori64. In *ASIACRYPT 2016, Part II* (Dec. 2016), J. H. Cheon and T. Takagi, Eds., vol. 10032 of *LNCS*, Springer, Heidelberg, pp. 3–33.

[267] Todo, Y., Leander, G., and Sasaki, Y. Nonlinear invariant attack: Practical attack on full SCREAM, iSCREAM, and Midori64. *Journal of Cryptology 32*, 4 (Oct. 2019), 1383–1422.

[268] Todo, Y., and Morii, M. Bit-based division property and application to Simon family. In *FSE 2016* (Mar. 2016), T. Peyrin, Ed., vol. 9783 of *LNCS*, Springer, Heidelberg, pp. 357–377.

[269] Townsend, J., Koep, N., and Weichwald, S. Pymanopt: A Python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research 17*, 137 (2016), 1–5.

[270] Toz, D., and Dunkelman, O. Analysis of two attacks on reduced-round versions of the SMS4. In *ICICS 08* (Oct. 2008), L. Chen, M. D. Ryan, and G. Wang, Eds., vol. 5308 of *LNCS*, Springer, Heidelberg, pp. 141–156.

[271] van Dam, W., and Hallgren, S. Efficient quantum algorithms for shifted quadratic character problems. *arXiv preprint quant-ph/0011067* (2000).

[272] van Oorschot, P. C., and Wiener, M. J. Parallel collision search with application to hash functions and discrete logarithms. In *ACM CCS 94* (Nov. 1994), D. E. Denning, R. Pyle, R. Ganesan, and R. S. Sandhu, Eds., ACM Press, pp. 210–218.

[273] van Oorschot, P. C., and Wiener, M. J. Parallel collision search with cryptanalytic applications. *Journal of Cryptology 12*, 1 (Jan. 1999), 1–28.

[274] Vaudenay, S. An experiment on DES statistical cryptanalysis. In *ACM CCS 96* (Mar. 1996), L. Gong and J. Stern, Eds., ACM Press, pp. 139–147.

[275] Vaudenay, S. Decorrelation: A theory for block cipher security. *Journal of Cryptology 16*, 4 (Sept. 2003), 249–286.

[276] Verbauwhede, M. Tools for integral cryptanalysis. Master's thesis, KU Leuven, 2022.

[277] Vielhaber, M. Breaking ONE.FIVIUM by AIDA an algebraic IV differential attack. Cryptology ePrint Archive, Report 2007/413, 2007. `https://eprint.iacr.org/2007/413`.

[278] von Sterneck, R. Sur la distribution des résidus et des non-résidus quadratiques d'un nombre premier. *Matematicheskii Sbornik 20*, 2 (1898), 269–284.

[279] Wagner, D. A generalized birthday problem. In *CRYPTO 2002* (Aug. 2002), M. Yung, Ed., vol. 2442 of *LNCS*, Springer, Heidelberg, pp. 288–303.

[280] Wagner, D. Towards a unifying view of block cipher cryptanalysis. In *FSE 2004* (Feb. 2004), B. K. Roy and W. Meier, Eds., vol. 3017 of *LNCS*, Springer, Heidelberg, pp. 16–33.

[281] Wan, D. A Chevalley-Warning approach to p-adic estimates of character sums. *Proceedings of the American Mathematical Society* (1995), 45–54.

[282] Wang, G. Improved impossible differential crytanalysis on SMS4. In *Communications and Intelligence Information Security (ICCIIS) 2010* (2010), pp. 105–108.

[283] Wang, X., and Yu, H. How to break MD5 and other hash functions. In *EUROCRYPT 2005* (May 2005), R. Cramer, Ed., vol. 3494 of *LNCS*, Springer, Heidelberg, pp. 19–35.

[284] Wegener, F., Baiker, C., and Moradi, A. Shuffle and mix: On the diffusion of randomness in threshold implementations of Keccak. In *COSADE 2019* (Apr. 2019), I. Polian and M. Stöttinger, Eds., vol. 11421 of *LNCS*, Springer, Heidelberg, pp. 270–284.

[285] Weil, A. On some exponential sums. *Proceedings of the National Academy of Sciences of the United States of America 34*, 5 (1948), 204.

[286] Wu, H., Bao, F., Deng, R. H., and Ye, Q.-Z. Improved truncated differential attacks on SAFER. In *ASIACRYPT'98* (Oct. 1998), K. Ohta and D. Pei, Eds., vol. 1514 of *LNCS*, Springer, Heidelberg, pp. 133–147.

[287] Xu, Z., Li, Y., Jiao, L., Wang, M., and Meier, W. Do NOT misuse the Markov cipher assumption – Automatic search for differential and impossible differential characteristics in ARX ciphers. ePrint, Report 2022/135, 2022.

[288] Youssef, A., Mister, S., and Tavares, S. On the design of linear transformations for substitution permutation encryption networks. In *Workshop on Selected Areas of Cryptography (SAC96)* (1997), pp. 40–48.

[289] Yun, A., Park, J. H., and Lee, J. On Lai–Massey and quasi-Feistel ciphers. *Designs, Codes and Cryptography 58*, 1 (2011), 45–72.

[290] Zhang, L., Zhang, W., and Wu, W. Cryptanalysis of reduced-round SMS4 block cipher. In *ACISP 08* (July 2008), Y. Mu, W. Susilo, and J. Seberry, Eds., vol. 5107 of *LNCS*, Springer, Heidelberg, pp. 216–229.

[291] Zhang, L., Zhang, W., and Wu, W. Cryptanalysis of reduced-round SMS4 block cipher. In *Information Security and Privacy, 13th Australasian Conference, ACISP 2008, Wollongong, Australia, July 7-9, 2008, Proceedings* (2008), Y. Mu, W. Susilo, and J. Seberry, Eds., vol. 5107 of *Lecture Notes in Computer Science*, Springer, pp. 216–229.

[292] Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., and Verbauwhede, I. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. *Sci. China Inf. Sci. 58*, 12 (2015), 1–15.

[293] Zhang, W., Ding, T., Yang, B., Bao, Z., Xiang, Z., Ji, F., and Zhao, X. KNOT: Algorithm specifications and supporting document. *Submission to NIST lightweight cryptography project* (2019).

[294] Zhang, W., Ding, T., Zhou, C., and Ji, F. Security analysis of KNOT-AEAD and KNOT-Hash. NIST Lightweight Cryptography Workshop, 2020.

[295] Zhang, W., Wu, W., Feng, D., and Su, B. Some new observations on the SMS4 block cipher in the chinese WAPI standard. In *Information Security Practice and Experience, 5th International Conference, ISPEC 2009, Xi'an, China, April 13-15, 2009, Proceedings* (2009), F. Bao, H. Li, and G. Wang, Eds., vol. 5451 of *Lecture Notes in Computer Science*, Springer, pp. 324–335.

[296] Zhao, Y., Liu, Y., and Wang, M. Improved differential attack on 23-round SMS4 (in Chinese). *Journal of Software 2018*, 29(9) (2018), 2821–2828.

[297] Zheng, Y., Matsumoto, T., and Imai, H. On the construction of block ciphers provably secure and not relying on any unproved hypotheses. In *CRYPTO'89* (Aug. 1990), G. Brassard, Ed., vol. 435 of *LNCS*, Springer, Heidelberg, pp. 461–480.

Curriculum vitae

Education

Internship at COSIC, KULeuven June – July 2014
August – September 2015
July – August 2016
July 2017

Hosted by Vincent Rijmen

Bachelor of engineering at KULeuven 2014 – 2017
Computer science – electrical engineering
Summa cum laude with congratulations of the Board of Examiners

Research visit at INRIA-Paris September 2017
Hosted by Anne Canteaut

Honours program at KULeuven March 2019
Faculty of engineering science

Master of mathematical engineering at KULeuven 2017 – 2019
Summa cum laude with congratulations of the Board of Examiners

Research visit at Ruhr University Bochum September 2019
Hosted by Gregor Leander

Teaching

Teaching assistant 2019 – 2022
Exercise sessions "Toegepaste algebra", first bachelor of engineering science.

Internship supervision 2021
Summer internship by bachelor student Corentin Bonte

Master's thesis supervision 2021
Supervision of Michiel Verbauwhede, jointly with Chaoyun Li
Tools for integral cryptanalysis

Awards

Best student award 2015
Awarded by the faculty of engineering science, KULeuven
Given to ten students with the highest average grades in the first bachelor

Best paper at Asiacrypt 2018 December 2018
Awarded by the program committee on behalf of the IACR
Block cipher invariants as eigenvectors of correlation matrices

Onespan master's thesis award July 2019
Awarded by the faculty of engineering science, KULeuven (€1500)
Linear cryptanalysis in the weak-key model

Legendre PRF cryptanalysis challenge prize October 2019
Awarded by the Ethereum foundation (1 ETH)
Joint work with Ward Beullens

Most interesting paper about the Legendre PRF March 2021
Awarded by the Ethereum foundation ($1000)
Joint work with Ward Beullens, Aleksei Udovenko and Giuseppe Vitto
Cryptanalysis of the Legendre PRF and generalizations

Best early career researcher paper at Crypto 2021 August 2021
Honorable mention best paper award
Awarded by the program committee on behalf of the IACR
Linear cryptanalysis of FEA and FF3-1

Best student paper at Asiacrypt 2021 December 2021
Awarded by the program committee on behalf of the IACR
A geometric approach to linear cryptanalysis

List of publications

Conferences

1. Beierle, C., Beyne, T., Felke, P., and Leander, G. Constructing and deconstructing intentional weaknesses in symmetric ciphers. In *Advances in Cryptology – CRYPTO 2022* (Cham, 2022), Y. Dodis and T. Shrimpton, Eds., Springer Nature Switzerland, pp. 748–778

2. Beyne, T., and Rijmen, V. Differential cryptanalysis in the fixed-key model. In *Advances in Cryptology - CRYPTO 2022 - 42st Annual International Cryptology Conference, CRYPTO 2022, Santa Barabara, CA, USA, August 15-18, 2022, Proceedings, Part I* (2022), Y. Dodis and T. Shrimpton, Eds., Lecture Notes in Computer Science, Springer

3. Beyne, T., and Chen, Y. L. Provably secure reflection ciphers. In *Advances in Cryptology – CRYPTO 2022* (Cham, 2022), Y. Dodis and T. Shrimpton, Eds., Springer Nature Switzerland, pp. 234–263

4. Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Multi-user security of the Elephant v2 authenticated encryption mode. In *Selected Areas in Cryptography: 28th International Conference, Virtual Event, September 29–October 1, 2021, Revised Selected Papers* (2022), Springer, pp. 155–178

5. Beyne, T., Dhooghe, S., Ranea, A., and Sijacic, D. A low-randomness second-order masked AES. In *Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers* (2021), R. AlTawy and A. Hülsing, Eds., vol. 13203 of *Lecture Notes in Computer Science*, Springer, pp. 87–110

6. Beyne, T. A geometric approach to linear cryptanalysis. In *ASIACRYPT 2021, Part I* (Dec. 2021), M. Tibouchi and H. Wang, Eds., vol. 13090 of *LNCS*, Springer, Heidelberg, pp. 36–66
   Best student paper award

7. Beyne, T. Linear cryptanalysis of FF3-1 and FEA. In *CRYPTO 2021, Part I* (Virtual Event, Aug. 2021), T. Malkin and C. Peikert, Eds., vol. 12825 of *LNCS*, Springer, Heidelberg, pp. 41–69
   Best early career researcher award and honorable mention best paper

8. Beyne, T., Dhooghe, S., and Zhang, Z. Cryptanalysis of masked ciphers: A not so random idea. In *ASIACRYPT 2020, Part I* (Dec. 2020), S. Moriai and H. Wang, Eds., vol. 12491 of *LNCS*, Springer, Heidelberg, pp. 817–850

9. Beyne, T., Canteaut, A., Dinur, I., Eichlseder, M., Leander, G., Leurent, G., Naya-Plasencia, M., Perrin, L., Sasaki, Y., Todo, Y., and Wiemer, F. Out of oddity - new cryptanalytic techniques against symmetric primitives optimized for integrity proof systems. In *CRYPTO 2020, Part III* (Aug. 2020), D. Micciancio and T. Ristenpart, Eds., vol. 12172 of *LNCS*, Springer, Heidelberg, pp. 299–328

10. Beyne, T. Block cipher invariants as eigenvectors of correlation matrices. In *ASIACRYPT 2018, Part I* (Dec. 2018), T. Peyrin and S. Galbraith, Eds., vol. 11272 of *LNCS*, Springer, Heidelberg, pp. 3–31
Best paper award

11. Beyne, T., and Bilgin, B. Uniform first-order threshold implementations. In *SAC 2016* (Aug. 2016), R. Avanzi and H. M. Heys, Eds., vol. 10532 of *LNCS*, Springer, Heidelberg, pp. 79–98

Journals

12. Beyne, T., and Liu, Y. Truncated differential attacks on contracting Feistel ciphers. *IACR Transactions on Symmetric Cryptology 2022*, 2 (Jun. 2022), 141–160

13. Beyne, T., Dhooghe, S., Moradi, A., and Rezaei Shahmirzadi, A. Cryptanalysis of efficient masked ciphers: Applications to low latency. *IACR Transactions on Cryptographic Hardware and Embedded Systems 2022*, 1 (Nov. 2021), 679–721

14. Beyne, T. Block cipher invariants as eigenvectors of correlation matrices. *Journal of Cryptology 33*, 3 (July 2020), 1156–1183

15. Ashur, T., Beyne, T., and Rijmen, V. Revisiting the wrong-key-randomization hypothesis. *Journal of Cryptology 33*, 2 (Apr. 2020), 567–594

16. Beullens, W., Beyne, T., Udovenko, A., and Vitto, G. Cryptanalysis of the Legendre PRF and generalizations. *IACR Trans. Symm. Cryptol. 2020*, 1 (2020), 313–330

17. Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Dumbo, Jumbo, and Delirium: Parallel authenticated encryption for the lightweight circus. *IACR Trans. Symm. Cryptol. 2020*, S1 (2020), 5–30

Other

18. Gorodilova, A., Tokareva, N., Agievich, S., Beterov, I., Beyne, T., Budaghyan, L., Carlet, C., Dhooghe, S., Idrisova, V., Kolomeec, N., Kutsenko, A., Malygina, E., Mouha, N., Pudovkina, M., Sica, F., and Udovenko, A. An overview of the eight international olympiad in cryptography "non-stop university crypto", 2022

19. Beyne, T., and Li, C. Cryptanalysis of the MALICIOUS framework. Cryptology ePrint Archive, Report 2020/1032, 2020. `https://eprint.iacr.org/2020/1032`

20. Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Elephant v2. Submission to the NIST lightweight cryptography competition, 2021

21. Beyne, T., Canteaut, A., Leander, G., Naya-Plasencia, M., Perrin, L., and Wiemer, F. On the security of the Rescue hash function. Cryptology ePrint Archive, Report 2020/820, 2020. `https://eprint.iacr.org/2020/820`

22. Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Status update on Elephant, 2020

23. Beyne, T. Linear cryptanalysis in the weak key model. Master's thesis, KU Leuven, 2019

24. Beyne, T., Chen, Y. L., Dobraunig, C., and Mennink, B. Elephant v1. Submission to the NIST lightweight cryptography competition, 2019

Index

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
IMEC-COSIC
Kasteelpark Arenberg 10 – bus 2452
B-3001 Leuven
http://cosic.esat.kuleuven.be