KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Mathematical and Provable Security Aspects of Post-Quantum Cryptography

Alan Szepieniec

Supervisor:
Prof. dr. ir. B. Preneel
Co-supervisor:
Prof. dr. ir. F. Vercauteren

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

December 2018

Mathematical and Provable Security Aspects of Post-Quantum Cryptography

Alan SZEPIENIEC

Examination committee:
Prof. dr. Bart Demoen, chair
Prof. dr. ir. B. Preneel, supervisor
Prof. dr. ir. F. Vercauteren
Prof. dr. Wim Veys
Dr. Aysajan Abidin
Dr. Wouter Castryck
Prof. dr. Jacques Patarin
(Université de Versailles
Saint-Quentin-en-Yvelines)

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

December 2018

Uitgegeven in eigen beheer, Alan Szepieniec, Leuven, B-3001 Leuven (Belgium)

Acknowledgements

To credit myself alone with the production of this work would be a crime of omission. The help and support I received from many sources was instrumental. Let credit be done where it is due.

First — to the members of the jury, for taking the time to trudge through my often needlessly verbose verbiage, mastering the material for an honest appraisal, and for lacing that appraisal with a healthy dose of critique; and to Jens Hermans too, whose sudden burial under a massive workload most conveniently satisfied paperwork constraints — thank you. I am sorry for making them dress up.

Second — to supervisors Bart Preneel and Frederik Vercauteren in particular, for providing me with an appropriate mixture of guidance and free reign – thank you. Their constant availability to answer queries has been indistinguishable from the ideal world.

Third — to collaborators in the past and present, (in no particular order) Bart Preneel, Jintai Ding, Albrecht Petzoldt, Mohamed Saied Emam Mohamed, Bart Mennink, Ward Beullens, Wouter Castryck, Frederik Vercauteren, Carl Bootland, Reza Reyhanitabar, Atul Luykx, Aysajan Abidin, Tomer Ashur, Siemen Dhooge, Abdelrahaman Aly, Marcel Tiepelt, for providing whetstone to make my thinking razor sharp — thank you. And thank you again to the master students among these, for graciously pretending that their own bright shine was partly my reflection.

Fourth — to my office mates throughout the years; to the Answerers of the Coffee Call and to the Drinkers of Friday Beers; to the organizers and partakers of all social events and activities; and great colleagues all round — thank you. Your contribution makes COSIC a joy to return to every morning (or, sometimes, afternoon). And to Péla Noë, the secretary whose glue and oil keeps the various parts of COSIC working smoothly together, and to the other non-technical staff for shielding us from administration and paperwork — thank you. And to the

proofreaders (and proof-readers) of the various chapters of this book — thank you. The errors the reader is bound to encounter, were put in afterwards.

Fifth — to my friends from de Reizende Reigers, Atlantis, and Brasa, for helping me with the life part of the work-life balance; and to my parents and brothers, whose endless support has kept me on track throughout my life — thank you.

I would also like to express gratitude toward the Flemish Agency for Innovation by Science and Technology (Agentschap voor Innovatie door Wetenschap en Technologie, IWT), which has since become the Flemish Agency for Innovation and Entrepreneurship (Vlaamse Agentschap voor Innoveren en Ondernemen, VLAIO), whose grant made my research possible.

Abstract

The ongoing construction of large-scale quantum computers gives rise to unique threats. By exploiting the peculiar properties of quantum particles, these computers can solve particular problems exponentially faster than their classical counterparts. Widely-deployed public key cryptosystems such as RSA and ECDH are vulnerable to quantum attacks.

In particular, Shor's celebrated quantum algorithms solve the integer factorization and discrete logarithm problems in polynomial time, thus breaking the public key cryptosystems that rely on them. Moreover, the adversarial model has an important impact on the validity of security proofs. Many classical security proofs fail when quantum adversaries are considered, even if they start from computational problems that are hard for quantum computers.

The design of post-quantum cryptosystems therefore requires a two-pronged approach: On the one hand, in the *mathematical* layer, the foundational hard problems should be computationally expensive on quantum computers as well as on classical ones. On the other hand, in the *provable security* layer, the reduction showing that a successful adversary implies a hard problem solver should rely only on proof techniques that hold for a quantum attacker model.

This dissertation presents a series of contributions to both layers. More specifically, on the mathematical side, the contributions are as follows.

- Chapter 6 § 6.1 presents a new construction for obtaining an efficiently-invertible encryption map from multivariate quadratic (MQ) polynomials. This expands the toolbox of the MQ cryptosystem designer.
- Chapter 6 § 6.4 introduces a new plausibly post-quantum hard problem, called the Short Solutions to Nonlinear Equations (SSNE) Problem, which boasts a better scaling behavior than its progenitors.

In terms of provable security, several independent results are spread out across two papers and the general overview.

- Chapter 6 § 6.3 introduces the notion of *constrained linear signature scheme* and shows that many post-quantum signature schemes are special cases. Moreover, this paper presents a transformation to shrink the public key at the expense of a larger signature, in order to reduce their combined size. This trade-off makes sense in the context of public key infrastructure.
- In the paper of Ch. 7 § 7.2 syntax and a security notion for *noisy key agreement (NKA)* protocols are introduced. We demonstrate that the correct security game for NKA protocols is the adaptation of the decisional Diffie-Hellman problem to the noisy case, which we call *noisy key distinguishing (NKD)*. Moreover, we provide a transformation for obtaining a key encapsulation mechanism (KEM) from an NKA protocol, and we provide a proof of security valid in the quantum-accessible random oracle model.
- To enable a refined reasoning about queries made to the random oracle, the same paper introduces the *aggregate quantum query amplitude* as a measure for the degree to which a quantum adversary makes a particular query. While the notion is implicit in other works, the standalone definition presented here is what enables the refined argumentation.
- Part I § 3.4 presents a comprehensive summary of the state of the art in terms of results related to quantum random oracle model. It puts the aggregate quantum query count at the center where it connects to many other extant results.

Lastly, fusing both mathematical and provable security layers into a coherent whole, the following concrete cryptosystems are proposed.

- Chapter 6 § 6.2 presents a blind signature scheme based on MQ primitives. A blind signature scheme enables the generation of a signature by a signer who remains ignorant of the message that is signed. It is a useful tool for untraceable cash and privacy-preserving protocols.
- Chapter 7 § 7.1 presents a digital signature scheme based on SSNE. This result positively answers the question left open at the end of Ch. 6 § 6.4 which introduced the hard problem but merely conjectured that it was useful for public key cryptography.
- Chapter 8 presents a key encapsulation mechanism relying on sparse integer arithmetic in a Mersenne ring. This relatively new hard problem

is similar in spirit to the lattice-based Ring Learning with Errors (RLWE) but its hardness is independent of the difficulty of lattice reduction. This cryptosystem was submitted to the NIST PQC project [75] without security proof. Its security is established in a rather generic fashion by the results in Ch. 7 § 7.2.

Beknopte samenvatting

De voortdurende constructie van grootschalige kwantumcomputers vormt een unieke bedreiging. Door de eigenaardige eigenschappen van quantumdeeltjes te benutten, kunnen deze computers bepaalde problemen exponentieel sneller oplossen dan hun klassieke tegenhangers. Publieke-sleutel-cryptosystemen zoals RSA en ECDH, die op grote schaal geïmplementeerd zijn, zijn kwetsbaar voor quantumaanvallen.

In het bijzonder lossen de breed gewaardeerde quantumalgoritmen van Shor de integerfactorisatie en discrete logaritmeproblemen op in polynomiale tijd, waardoor de publieke sleutel cryptosystemen die daarop steunen, als gebroken moeten worden beschouwd. Dit model van de tegenstander heeft bovendien een belangrijke invloed op de geldigheid van veiligheidsbewijzen. Veel klassieke veiligheidsbewijzen falen wanneer ze geconfronteerd worden met quantumtegenstanders, zelfs als ze beginnen met een rekenkundig probleem dat moeilijk is voor quantumcomputers.

Het ontwerp van post-quantum cryptosystemen vereist daarom een tweeledige benadering: aan de ene kant, in de *wiskundige* laag, zouden de fundamentele moeilijke problemen rekenkundig duur moeten zijn op zowel quantumcomputers als klassieke. Aan de andere kant, in de laag van *bewijsbare veiligheid*, moet de reductie die aantoont dat een succesvolle tegenstander een oplosser impliceert voor het moeilijke probleem, enkel berusten op bewijstechnieken die gelden voor een quantummodel van de aanvaller.

Dit proefschrift presenteert een reeks bijdragen aan beide lagen. Meer specifiek zijn de bijdragen als volgt aan de wiskundige kant.

- Hoofdstuk 6 § 6.1 presenteert een nieuwe constructie voor het verkrijgen van een efficiënt-inverteerbare afbeelding op basis van multivariate kwadratische (MQ-) veeltermen. Dit breidt de toolbox uit van de ontwerper van MQ-cryptosystemen.

- Hoofdstuk 6 § 6.4 introduceert een nieuw plausibel post-quantum moeilijk probleem, het “Short Solutions to Nonlinear Equations” (SSNE) probleem, dat een beter schalingsgedrag heeft dan zijn voorlopers.

In termen van bewijsbare beveiliging zijn de verschillende onafhankelijke resultaten verdeeld over twee artikelen en het algemene overzicht.

- Hoofdstuk 6 § 6.3 introduceert het begrip *begrensd lineair handtekeningschema* en toont aan dat veel post-quantum digitale handtegekingschema’s speciale gevallen zijn. Bovendien presenteert dit document een transformatie om de publieke sleutel te verkleinen ten koste van een grotere handtekening, om de gecombineerde grootte te reduceren. Deze afweging houdt steek in de context van publieke-sleutel-infrastructuur.
- In het artikel van Hoofdstuk 7 § 7.2 wordt een syntaxis en een veiligheidsbegrip voor *noisy key agreement (NKA)* protocollen geïntroduceerd. We tonen aan dat het juiste beveiligingsspel voor NKA-protocollen de analoge is van het Diffie-Hellman-beslissingsprobleem maar met ruis; we noemen dit probleem *noisy key distinguishing (NKD)*. Bovendien bieden we een transformatie voor het verkrijgen van een sleutel-inkapselingsmechanisme (*key encapsulation mechanism*, KEM) uit een NKA-protocol en we leveren een veiligheidsbewijs dat geldig is in het quantumtoegankelijke random orakelmodel.
- Om een verfijnde redenering mogelijk te maken over de query’s die aan het random orakel gemaakt worden, introduceert datzelfde artikel de *aggregate quantum query amplitude* als een grootheid voor de mate waarin een quantumvijand een bepaalde query maakt. Hoewel het begrip impliciet is in andere werken, is de afzonderlijke definitie hier wat de verfijnde argumentatie mogelijk maakt.
- Deel I § 3.4 presenteert een uitgebreide samenvatting van de stand van de techniek in termen van resultaten gerelateerd aan het quantum random-orakel-model. Het plaatst de aggregate quantum query amplitude in het centrum waar het verbinding maakt met vele andere bestaande resultaten.

Ten slotte worden de volgende concrete cryptosystemen voorgesteld, waarbij zowel wiskundige als bewijsbare veiligheidslagen worden samengevoegd tot een samenhangend geheel.

- Hoofdstuk 6 § 6.2 presenteert een schema voor geblindeerde digitale handtekeningen gebaseerd op MQ-primitieven. Een dergelijk schema maakt het mogelijk voor een ondertekenaar om een handtekening aan te maken voor een bericht waarvan hij onkundig blijft. Het is een handig hulpmiddel voor ontraceerbaar digitaal geld en privacybeschermende protocollen.

- Hoofdstuk 7 § 7.1 presenteert een schema voor digitale handtekeningen op basis van SSNE. Dit resultaat geeft een positief antwoord op de vraag aan het einde van Hoofdstuk 6 § 6.4, dat het moeilijke probleem introduceerde maar slechts vermoedde dat het ook nuttig was voor publieke sleutel cryptografie.

- Hoofdstuk 8 presenteert een sleutel-inkapselingsmechanisme, gebaseerd op spaarse gehele getallen en hun rekenkunde in een Mersenne-ring. Dit relatief nieuw moeilijk probleem lijkt qua geest op het roostergebaseerde Ring Learning with Errors (RLWE), maar de moeilijkheid is onafhankelijk van de moeilijkheid van roosterreductie. Dit cryptosysteem werd ingediend bij het NIST PQC-project [75], maar zonder veiligheidsbewijs. De veiligheid ervan werd op een generieke wijze vastgelegd door de resultaten van Hoofdstuk 7 § 7.2.

Contents

List of Figures

List of Games

List of Hard Problems

Chapter 1

Introduction

Cryptography, in the first sense of the word, is the science of protecting information. The objective of this practice is formulated in terms of precisely defined properties of the information that its protection should guarantee; the collection of these properties may be referred to as *security.* The space of adversaries capable of bypassing or nullifying the protections must be bounded only by assumptions that are realistic. This preference for realism eliminates the need for trust to the greatest possible extent, which is after all the goal implied by the need to protect anything at all. The protections themselves are put into effect by tools specifically designed for the purpose; *cryptography*, in the second sense of the word, refers to the collection of these tools.

Cryptography is often associated with secrecy, which is the security property that aims to prevent the adversary from reading a transmitted message or learning anything about it. However, for the purpose of formalizing precise security properties, this term is rather vague. For instance, the secrecy of the content of a message is known more precisely as *confidentiality*, whereas the secrecy of its authorship is known as *anonymity.* One can even go a step further and require *unobservability*, which keeps secret whether or not the message was sent in the first place. More importantly, secrecy fails to capture the quality of information that renders it immune to modification by third parties; this property is known as *integrity.* More generally, the property of a message that in addition to integrity guarantees that its source is who it claims to be (however *identity* is defined), is known as *authenticity.* Furthermore, some protocols are orders of magnitude more complex than simple message transfer and require accordingly complex definitions of security properties.

Any cryptographic tool or *cryptosystem* comes with an *adversarial model*, which captures the class of adversaries against which the system guarantees the claimed security. Generally, only adversaries with constrained resources are considered, for instance with constraints in terms of computing time, computing power, or number of particular protocol interactions. A natural design strategy is then to guarantee that any violation of security properties necessarily implies a resource constraint violation on the part of the adversary. No guarantee is offered against adversaries not contained in the model, for instance adversaries with access to side channel information such as timing information or power traces; or against active adversaries capable of injecting faults.

For instance, a cryptosystem may be considered secure if breaking it requires solving a computational problem whose optimal solving time is larger than the time frame that is available to the adversary. Relying on the *computational complexity* of certain *hard problems* is indeed a popular strategy, but it generally requires an additional *hardness assumption* as hardness proofs are exceedingly rare in the field of computational complexity. Nevertheless, the benefit of reducing the insecurity of a cryptosystem to a violation of the hardness assumption comes from the *mathematical* statement of the hard problem that is independent of the cryptosystem, thus enabling and inviting independent study and hence a stronger hardness argument. The term *provable security* refers to the property of a cryptosystem having such a security proof, as well as to the study and development of security proofs. In many cases cryptosystems require *interactive hardness assumptions*, where the task is not to solve a non-interactive problem but rather to win an interactive *game*, and in other cases still there is no independent assumption to speak of.

1.1 Symmetric and Public Key Cryptography

The distinction between symmetric and public key cryptography is drawn based on the distribution of secret key material. When all participants are in possession of the same secret information and the adversary is not, then the situation is captured by *symmetric cryptography*. In contrast, in the case of *public key cryptography*, the secret information is distributed asymmetrically, for instance by distributing the encryption key to the public and keeping the decryption key secret. In this model, anyone can encrypt a message that only the intended recipient can decrypt.

The obvious benefit of public key encryption over symmetric encryption is the reduced burden associated with key management. In the symmetric key scenario, a user and each of his intended communication partners must be in possession of a unique key that was shared beforehand. In contrast, in the public key scenario, a user can decide *ad hoc* whom to communicate with. After all, the transmission of the public encryption key and of the preceding request for it can occur over a public channel, assuming a *passive adversary*, *i.e.*, one who eavesdrops only.

A similar asymmetry benefits key management in the setting of an *active adversary*, *i.e.*, one who can alter, block and forge messages. Before the user can encrypt a message under a public key, he must verify that the given public key was indeed generated by the intended recipient and is not the clever forgery of a malicious adversary. To facilitate a straightforward authentication test, the public encryption key can be transmitted along with a *digital signature* which is efficiently verifiable under a public key but can only be generated by the matching secret key. To test the authenticity of *that* public key, the user can verify another signature under another public key, and so on, traversing a tree of signature-public key links whose root is a public key that is already known to the user. Using this *public key infrastructure*, the user can verify the authenticity of an exponential number of public keys by simply storing a small number of root public keys. The best strategy attainable using only symmetric cryptography requires either interacting iteratively with a trusted third party or else obtaining and storing *all* keys of all parties the user might want to communicate with.

Where techniques for symmetric cryptography excel in comparison to their public key counterparts is in their performance. Symmetric primitives like block ciphers and hash functions are orders of magnitude faster than typical public key primitives. This comparatively poor performance on the part of public key algorithms is due to their need to achieve functionality through the preservation of homomorphic properties. The most pertinent use of public key cryptography in practice is to establish a shared symmetric key, after which point the symmetric key is used to secure communication much less expensively. In fact, *key agreement* protocols are public key protocols that are tailored to this use case by virtue of omitting the transmission of public keys and ciphertexts in favor of a pair of protocol contributions and deriving an identical symmetric key from the one party's secret key and the other party's contribution.

The examples covered so far —public key encryption, digital signatures, and key agreement— constitute only a small subset of functionalities classifiable as public key protocols, although they are certainly the most used and most deployed public key schemes. For example, *homomorphic public key encryption* enables operations on ciphertexts that remain meaningful after decryption. *Zero-knowledge proofs* are protocols that enable one party to prove the truth of a statement to another party without revealing anything beyond the fact that the claim is true. *Blind signatures* mimic the physical placement of a signature on an envelope made of carbon paper by digitally enabling a credential issuer and receiver to jointly generate a signature that is unlinkable to the issuer's view of the process. Most generally, *multiparty computation (MPC)* protocols enable any number of participants, each holding a potentially different secret input, to compute the value of a function of their inputs. There is no shortage of complex public key protocols achieving specific functionalities in a way that is more efficient than applying generic MPC. Likewise, there are a wide range of properties of public key schemes beyond encryption and digital signatures that might be desirable in specific contexts, as well as proposed cryptosystems to achieve them.

The previous description benefits from an explanatory example. To this end, one cannot do better than review the textbook RSA cryptosystem [118]. This cryptosystem presents a *trapdoor function*[1]: a function that is easy to evaluate but hard to invert by anyone ignorant of the secret trapdoor information. In the case of RSA, this function is exponentiation in $\mathbb{Z}_n$, the ring of integers modulo a product of large primes $n = pq$. Specifically, given $x \in \mathbb{Z}_n$ and given an exponent $e \in \mathbb{Z}$ it is easy to compute $x^e \bmod n$; but it is difficult to compute x from e and $x^e \bmod n$.

Example 1. Textbook RSA Encryption

- *Key Generation.* Pick two large primes p and q and set $n = pq$; compute $\lambda = \mathsf{lcm}(p-1, q-1)$; pick a random public exponent e and compute $d \equiv e^{-1} \bmod \lambda$. The public key is (n, e), and the private key is d.
- *Encryption.* To encrypt a message $m \in \mathbb{Z}_n$ compute $c \equiv m^e \bmod n$.
- *Decryption.* To decrypt a ciphertext c, compute $m \equiv c^d \bmod n$.

The label "textbook" in "textbook RSA" refers to the fact that the present description may be sufficient to convey intuition about how and why the cryptosystem works, but ultimately falls short of achieving concrete security properties. For instance, since encryption is deterministic, the same plaintext will be mapped to the same ciphertext, which is enough already for an attacker to determine whether the same message was sent twice. Other attacks exploit the following homomorphic property of ciphertexts: $c_1 c_2 \equiv (m_1^e)(m_2^e) \equiv (m_1 m_2)^e \bmod n$ and so by tricking the user into decrypting the ciphertext $c_1 c_2$, the attack obtains the product of plaintexts. In order to implement a rigorously

[1] Trapdoor functions are sometimes also called *trapdoor one-way functions*, a terminology I intentionally avoid because trapdoor functions are not one-way.

secure version of the RSA cryptosystem, we refer to the OAEP construction of Bellare and Rogaway [21], to Shoup's RSA key encapsulation mechanism [124], or to PKCS#1 [88].

The RSA cryptosystem is actually quite unique because it presents a trapdoor function that is *bijective*. Where the injective property is used for decryption, the surjective property can be used for signature generation in a digital signature scheme. This next example additionally requires a hash function $\mathsf{H} : \{0,1\}^* \rightarrow \mathbb{Z}_n$, which informally speaking is a deterministic map of bitstrings of any length to random-looking elements of a target range. Usually this target range is the set of bit strings of length κ but in the case of the RSA signature scheme it is the ring of integers modulo n.

Example 2. RSA Signature Scheme

- *Key Generation.* Pick two large primes p and q and set $n = pq$; compute $\lambda = \mathsf{lcm}(p-1, q-1)$; pick a random public exponent e and compute $d \equiv e^{-1} \bmod \lambda$. The public key is (n, e) and the private key is d.
- *Signature Generation.* To sign a document $m \in \{0,1\}^*$, compute its hash $h = \mathsf{H}(m)$, and compute the signature $s \equiv h^d \bmod n$.
- *Signature Verification.* To verify a signature s on a document m, test whether $s^e \equiv \mathsf{H}(m) \bmod n$.

The RSA cryptosystems, when securely implemented, derive security from the computational hardness of inverting the RSA function $f_e : \mathbb{Z}_n \rightarrow \mathbb{Z}_n, x \mapsto x^e \bmod n$. Currently, the best-performing attack on this problem is the number field sieve to factorize n [92]. This algorithm heuristically runs in time $L_n\left[\frac{1}{3}, \sqrt[3]{\frac{64}{9}}\right] = \exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right)(\ln n)^{\frac{1}{3}}(\ln \ln n)^{\frac{2}{3}}\right)$. For a 3072-bit modulus, this amounts to roughly $2^{138.74}$ elementary operations, or $6.13 \cdot 10^{15}$ billion years on a single 3 gigahertz processor. (For reference: the universe is only 13.8 billion years old, at the time of writing.) Compare this attack complexity with the running time for legitimate users, as measured on my own Intel 2.4 GHz quadcore machine running OpenSSL: 34.0 milliseconds for generating a key pair, 7.0 milliseconds for generating signatures and 3.1 milliseconds for verifying them.

Note that the public and private operations of the RSA cryptosystems can be expressed as elementary group-theoretical operations. RSA is not alone in this reliance on group theory; other widely-deployed systems such as Diffie-Hellman key exchange [106], DSA [74], and their elliptic curve counterparts have it as well. Even the first cryptosystems with fancy properties like homomorphic encryption or blind signatures have the same feature. This early skew towards using group theory for public key cryptography is no accident: group theory provides an abundance of useful one-way homomorphisms, especially compared to competing branches of mathematics for public key cryptography. However, in the context of adversaries capable of performing quantum computations, having an abundance of homomorphic properties seems to be a fatal flaw rather than a selling point.

1.2 Quantum Computers

In the early 1980's Richard Feynman gave a talk [57] in which he made the observation that simulating quantum physics on classical computers seemed like, and likely was, an intractable task. He followed up this observation by conjecturing that *quantum computers*, *i.e.*, physical devices whose inner mechanics relied on quantum phenomena, *would* be good candidates for simulating quantum mechanics. Soon after, David Deutsch formalized the notion of a quantum Turing machine and showed that it was universal: a quantum Turing machine can simulate any quantum mechanical process with small overhead and independently of the substrate [44].

The question then arises, are there natural computational problems (beyond simulating quantum physics or contrived problems) that quantum computers can solve faster than classical computers can? Shor's influential 1994 paper answered this question positively: he presented *polynomial-time quantum* algorithms to solve the integer factorization and discrete logarithm problems [123] — problems for which, to date, no efficient classical algorithms exist. The impact on public key cryptography should be obvious: large enough quantum computers break factorization-based cryptosystems (such as RSA) as well as cryptosystems based on the discrete logarithm (such as elliptic curve cryptosystems).

But are quantum computers realistic? Only time will tell. A once-common criticism is that the presence of noise and decoherence will restrict the power of quantum computations in practice. This criticism is less common today because quantum error-correcting codes have been shown to enable the encoding of a single logical qubit into multiple physical qubits and its error-correction, should it have been disturbed by noise, *without affecting the logical qubit's value* [32, 128]. Consequently, it is possible to sustain a quantum state arbitrarily long and compute quantumly on that state, provided that the additive noise rate remains below a nonzero threshold [6, 84, 85]. These results strongly indicate that

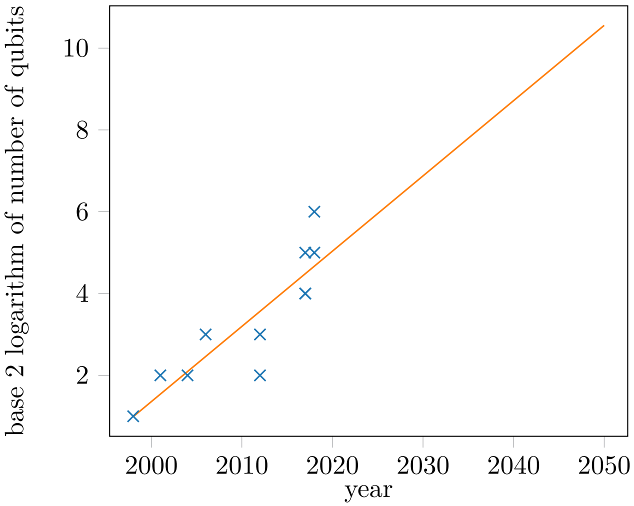


Figure 1.1: Extrapolation of progress on quantum computer construction. Sources: [38, 141, 153, 105, 94, 149, 71, 77, 72, 82, 76].

large-scale quantum computers are, at least in theory, practically feasible, and "merely" a massive engineering challenge.

Many research groups around the world, including Google, IBM, Intel, are working on the construction of quantum computers. The present worldwide revenue of the supercomputer market is estimated at $4 billion [127], and Dave Wecker of Microsoft's QuArC Group estimates that 50% of it is spent on simulation of quantum many-body systems for chemicals, pharmaceuticals and materials science [143]. These are tasks that stand to benefit dramatically from even moderate-scale quantum computers and the potential economic gains are sure to guarantee continued funding for research into their construction for many years to come. Meanwhile, the claims made by these research groups are getting stronger and stronger: a straightforward extrapolation of progress suggests that quantum computers will provide the required 1754 logical qubits[2] to break currently deployed elliptic curve cryptosystems within 35 years. If Moore's law holds for qubits as it does for transistors, this event will occur much sooner.

Despite the appearance of progress, some noteworthy computer scientists such as Gil Kalai remain skeptical about the possibility of scalable quantum

[2] Estimated using the circuit of Roetteler *et al.* [119] to attack NIST standard curve P-192 [74, §D.2.1].

computers [78, 79, 80]. Kalai argues that on quantum computers there will be a strong tendency for errors to synchronize, like metronomes on a floating board. As a result of this synchronization, the errors will entangle and corrupt many qubits at once, instead of the occasional somewhat isolated qubit that quantum error-correcting codes allow for. Preskill responds to this criticism by showing that scalable fault-tolerant quantum computing is possible for a large class of correlated error models [116]. The question remains open whether there are noise models that are compatible with quantum mechanics but make scalable fault-tolerant quantum computation impossible, and if noise of that type is likely to occur in practice.

However, even if the skepsis of the doubters is well-founded, they must associate a nonzero probability, no matter how small, to the physical realization of a scalable quantum computer within 20 years. The exact magnitude of this probability is important because it should be factored in into a security calculation. A quantum-skeptic cryptographer who estimates the odds of scalable quantum computation as inconceivably low as 10^{-20} cannot simultaneously claim a 128 bit security level for a cryptosystem that is known to be vulnerable to an efficient quantum attack.

1.3 Post-Quantum Cryptography

Post-quantum cryptography refers to the science of protecting information against both quantum and classical attacks, as well as to the collection of tools that accomplish this task. The field consists of many branches such as the study of quantum-secure hard problems, the design of concrete cryptosystems relying on them, quantum attack algorithms, provably security against quantum attackers, secure implementations, *et cetera.* Post-quantum cryptography is steadily gaining more and more traction among cryptographers, as evidenced by the attendance of the now-yearly Post-Quantum Crypto conference rising year after year [135, 90, 89]; the approval of the PQCRYPTO and PROMETHEUS projects by the EU [136, 24]; and the PQC project by the US National Institute for Standards and Technology (NIST) which has the express intention of issuing a standard within five years [75].

Unfortunately, the adoption of post-quantum cryptography is not cost-free. The post-quantum hard problems (except for hash inversion) have been studied less than integer factorization and the discrete logarithm problem. Consequently, a post-quantum hard problem inevitably confers a weaker security assurance compared to a pre-quantum alternative due to the greater potential of future improvements on attacks. Additionally, many of the hard problems that hold

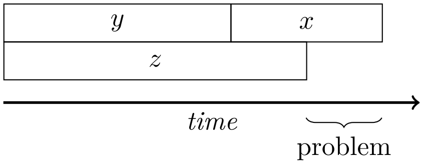


Figure 1.2: Visual representation of Mosca's argument.

promise of resisting attacks on quantum computers require far greater memory and bandwidth, impeding their adoption into cryptosystems for low-cost devices. Additionally, these hard problems sometimes introduce the potential for failure events (*e.g.* decryption failures) despite honest, non-malicious, usage — in which case developing a security proof is a tricky endeavor. Other branches of post-quantum mathematics do not have security proofs to begin with, although this might be merely due to the laziness or lack of intelligence of the researchers that study them. At any rate, there are many challenges to be answered before the end-goal of securing the information flows against quantum attacks in today's economy can be realized.

Nevertheless, there is a compelling argument to be made that developing and deploying post-quantum cryptography is an urgent task rather than a back-up plan to be executed when the time comes. Michele Mosca's most poignant articulation of this argument asks to consider the following time periods [104].

- How long must sensitive data remain cryptographically protected? Call this number x. For instance, ones present financial situation might be hardly relevant ten years from now, whereas ones health profile might be just as sensitive forty years from now.
- How long does it take to deploy new cryptography? Call this number y. How long does it take to replace all bank cards and terminals?
- How long does it take for quantum computers to break current cryptography? Call this number z. The 35 years extrapolation derived above is just one estimate; Mosca himself estimates the probability of quantum computers breaking RSA-2048 by 2031 at 50% [104].

If $x + y > z$, there is a problem. Sensitive information will be exposed to quantum attacks before the updated cryptography is deployed.

The argument is even more compelling in the special case of encryption. An adversary capable of intercepting and storing messages in transit can store them indefinitely. At a future point in time when quantum computers are available, the encrypted messages in the storage database can be decrypted. Therefore, in order to protect the confidentiality of transmissions today against the quantum computers of the future, we must already be using post-quantum encryption.

1.3.1 What Post-Quantum Cryptography is Not

Quantum Cryptography. An important restriction with respect to post-quantum cryptography is its reliance on classical hardware to execute the cryptographic algorithms. An alternative strategy that is appropriately called *quantum cryptography* is to replace the hardware so as to produce quantum phenomena that are then engineered to protect information [23, 53]. While a fascinating subject in its own right, with its own list of promised features, challenges and constraints, quantum cryptography is ultimately very different from (and should not be confused with) post-quantum cryptography precisely because it mandates instantiation on different physical devices. Quantum cryptography is entirely out of the scope of this dissertation.

Symmetric Key. Post-quantum cryptography is chiefly concerned with public key cryptography due to the structure embedded in its hard problems, *i.e.*, the same structure that enables Shor's algorithms and similar quantum attacks. In contrast, symmetric key primitives are, by and large, designed to break any and all structure. As a consequence, Shor's algorithms fail and as far as we can tell no derivative thereof achieves the same exponential speedup against block ciphers such as AES or DES or their common modes of operation. However, quantum computers can speed up the solution of generic search problems. In particular, Grover's algorithm [64] requires $O(\sqrt{N})$ queries to find a single marked element from a set of N. So in order to guarantee a minimum attack complexity of 2^k of a key search attack using Grover's algorithm, it suffices to use $2k$ key bits, assuming the cipher under attack has no exploitable structure. With respect to the k bits required to guarantee the same security level against a classical brute force attack, this security measure amounts to a doubling of the key size.

Scott Fluhrer goes a step further [60] and observes that in contrast to classical brute force searches, Grover's algorithm is inherently sequential. The running time cannot be reduced in exchange for more parallelism except at a very disadvantageous rate. A security measure should take this sequential nature into account and furthermore it should require that an attack run in less than, say, 200 years. With these constraints, an adequate security measure needs only increase in the number of key bits by a relatively small constant.

However, it is not obvious that Grover's algorithm always is the most efficient attack against a symmetric primitive running on a quantum computer. A nice little result by Bart Mennink and myself shows that in the particular case of the XOR of pseudorandom permutations, a popular construction of pseudorandom functions from pseudorandom permutations, there is a quantum attack that actually outperforms Grover [99]. Hosoyamada and Sasaki show similar results for a variety of symmetric key constructions [67, 68]. The commonality between these quantum attacks and those on public key cryptosystems is that the interaction between the attacker and the user (or more precisely: between the attacker and the secret key material) is classical; however, the attacker has a quantum computer at his disposal with which he can accelerate computations. From this point of view, engineering security for symmetric cryptosystems in this regime is properly a branch of post-quantum cryptography. However, the quantum attacks in this context are much less worrisome because they are all still exponential in some security parameter, and by tweaking this parameter appropriately the attacks can be made infeasible without impacting usability too much.

Cryptography on Quantum Computers. What happens if the quantum-enabled attacker *is* allowed quantum access to secret key material? In this context, the security of many modes of operation as well as MAC constructions fail completely [86, 87, 81, 13, 120]. The common basis for these attacks is Simon's quantum algorithm [126], which is a cousin of, and precursor to, Shor's algorithms. The design of symmetric key primitives retaining security even against quantum attackers that interact quantumly with the secret key material is a fascinating subject area. However, these algorithms must potentially be executed on quantum computers to support an advantage over standard symmetric key techniques, whereas for practical cryptography, the target platform is classical hardware. Nevertheless, in the context of white-box cryptography and trusted platform modules, where the user is presented with obfuscated code, this is a relevant attacker model. Additionally, an algorithm that is secure in this context will also be secure in the weaker setting where the interaction must be classical. Therefore, this security model constitutes a valid target for overkill design.

1.4 Outline

This dissertation presents a selection of results relating to mathematical and provable security aspects of post-quantum cryptography obtained over the last couple of years. These results are presented in Part II as papers, most of which have been published in peer-reviewed conference proceedings, but some of which are at the time of writing still unpublished manuscripts.

The purpose of the rest of this general overview is to provide the reader with a crash course on the necessary background with which to read, interpret, and critically assess the papers in Part II. To this end, Chapter 2 covers the necessary concepts of quantum computation, which is the standard computing model for quantum attackers. Next, Chapter 3 covers the basics of provable security, starting with some functionality descriptions and security definitions, followed by a generic explanation on how to show that a security definition is met by a concrete system, and concluding with an enumeration of and discussion about proof techniques in the quantum random oracle model. In Chapter 4 we survey some of the hard problems that promise simultaneously to resist attacks on quantum computers and allow for public key cryptography. Finally, this general overview is brought to a conclusion in Chapter 5 with a short summary and a discussion about open problems and potential research topics.

Chapter 2

Quantum Computation

The advantage of quantum computers over classical ones derives from *interference in configuration space*. This concept joins two phenomena that have classical counterparts.

Interference is the process by which multiple wave sources generate patterns through cancellation and reinforcement, as opposed to the uniform non-patterns associated with one or zero wave sources. At any given point, two arriving waves exist in *superposition*: together they form a single waveform whose amplitude is given by the sum of the components' amplitudes. If the two waves are in phase, the amplitudes have the same sign and are reinforced; if they are out of phase the sign is opposite and the amplitudes are canceled. Interference patterns are exhibited by all waves that we know of.

Configuration space is the set of all possible configurations in a probabilistic process. A single coin has two configurations: face up or face down. Ten coins have 2^{10} configurations. Probability theory requires that the sum of all configurations' probabilities equals one. Configurations are identifiable with events but in the context of computation it is helpful to think of them as potential states because the next computational step can depend on the previous state and alter the resulting distribution differently. From this perspective, probabilistic computations amount to manipulations of a probability density distribution. In order to be valid, these manipulations must retain the property of probability distributions that they integrate to one.

Turning to quantum mechanics, the Schrödinger equation describes the evolution of a wave in configuration space. The amplitude of this wave in a particular configuration is identifiable with that configuration's probability, except the amplitude is a complex number whereas probabilities are real numbers between zero and one. In particular, complex numbers can cancel whereas positive ones cannot. As a result, quantum processes exhibit interference in configuration space, in contrast to classical ones. Instead of all configurations' amplitudes summing to one, their squared norms sum to one. Phrased differently, quantum operations preserve the Euclidean length of unit-length state vectors.

Mysterious qualities have been ascribed to the phenomenon of quantum measurement and entanglement. Albert Einstein famously referred to the implied faster-than-light transmission of effect on borne probabilities as spooky action at a distance [52]. Roger Penrose argues that measurement is an inherently uncomputable phenomenon and may be the origin of consciousness [15]. In the many-world interpretation of quantum mechanics popularized by Everett, quantum measurements do not exist — they can be explained as the result of entanglement with quantum particles that exist outside of the considered system [69]. However, despite their counter-intuitive consequences, what quantum measurement and entanglement describe is not so different from classical processes with unknown variables described by observation and correlation. Before a classical system is observed, its state is drawn from a probability distribution. Observing the system enables the observer to collapse this probability distribution to a single point, in accordance with the observed variable. Measuring only a part of the system partially collapses the distribution to a refinement that is in accordance with the partial observation. For a pair of correlated coins, the observer of one coin at one end of the universe will know *instantly* whether the other coin at the other end of the universe is heads or tails. What separates probabilistic processes from quantum processes is that quantum amplitude distributions seem to *exist* — whereas classical probability distributions might be merely an abstraction invented by humans to cope with a lack of information. The keyword here is "might" because there is no way of distinguishing the world in which classical probability distributions exist as a physical entity from the world in which they do not.

This characterization of quantum mechanics suggests a dangerously simple —but perfectly valid— description of quantum computation: *quantum computation is probabilistic computation that preserves the ℓ_2-norm of the system's state instead of its ℓ_1-norm.* A formal proof of this fact is presented by Lucien Hardy [65]. A much more accessible and fun to read text [2] by Scott Aaronson tackles the related question, what is so special about the ℓ_1 and ℓ_2 norms, that Nature would choose to preserve these metrics rather than others? The following summary follows the inimitable approach of Nielsen and Chuang by building quantum computation from the ground up, starting with the postulates [108]. While this summary does cover the essentials, it is not complete. For a comprehensive treatment the Nielsen and Chuang book is the go-to resource.

2.1 State Vector Formulation

A *qubit* is a physical carrier of a unit of quantum information, in the same sense that a flip-flop is a physical carrier of a unit of classical information, this unit being called a bit. The classical mechanism that allows one to identify a flip-flop or a classical memory register with the value contained therein does not translate to the quantum world. In particular, one cannot copy quantum information from one carrier to another without changing it in general; this principle is known as the No Cloning Theorem [147, 45]. Therefore, it is important to make the distinction between the physical substrate, and its state at a given point in time. We refer to a collection of qubits jointly used for a particular purpose as a *quantum register*.

> **Postulate 1.** The state of a quantum register of k qubits is given by a state vector $|\psi\rangle \in \mathcal{H} \subset \mathbb{C}^{2^k}$ of 2^k complex numbers which has unit length in the ℓ_2-norm.

The notation $|\cdot\rangle$ is called *ket* notation; it stands in contrast to $\langle\cdot|$ which is a *bra* and denotes the same vector's conjugate transpose. The space $\mathcal{H}$ where the state vectors live is a Hilbert space, meaning that it is a vector space that is equipped with an inner product, which in this case allows for the aesthetically pleasing bra-ket notation $\langle\cdot|\cdot\rangle$. This notation is sometimes also called *Dirac notation*, after its inventor.

It is clear that a complete description of a quantum system requires not just a vector but also a basis for the Hilbert space. The most convenient *computational basis* is given by $\{|b_0\rangle, |b_1\rangle, \dots, |b_{2^k-1}\rangle\}$ where $|b_i\rangle$ represents the padded binary expansion of the number $i \in \{0, \dots, 2^k - 1\}$. Often times the $|b_i\rangle$ will be substituted for something more descriptive like $|15\rangle$ or $|a\rangle$ to refer to the computational basis vector associated with the padded binary expansion of the number fifteen or with the bitstring a.

Generic quantum systems are not described by computational basis vectors. In this case $|\psi\rangle = \sum \alpha_i |b_i\rangle$ is said to be a *superposition* of all bitstrings b_i whose *amplitude* α_i is nonzero. Whether a quantum system is in superposition or not, depends on the basis with which its state vector is considered.

> **Postulate 2.** Closed quantum systems evolve via the action of unitary matrices on the state vector. In particular, if $|\psi\rangle$ and $|\phi\rangle$ describe the same system at different points in time, then there is some unitary matrix $U \in \mathbb{C}^{2^k \times 2^k}$ such that $|\psi\rangle = U|\phi\rangle$.

Unitary matrices preserve the ℓ_2-norm of vectors they act on; in fact, this is one way to define unitarity. An alternate definition is the description of the inverse of a unitary matrix as its complex conjugate transpose $U^\dagger$, *i.e.*, $UU^\dagger = U^\dagger U = I$. The invertibility of unitaries translates to the reversibility of quantum computation.

> **Postulate 3.** Quantum measurement is defined with respect to a collection $\{M_m\}$ of measurement operators acting on $\mathcal{H}$, one for each possible event m. The probability of observing event m is given by $\Pr[m] = \langle\psi|M_m^\dagger M_m|\psi\rangle$; after observing this event the state of the system is given by $\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}$. The collection of measurement operators satisfies completeness: $\sum_m M_m^\dagger M_m = I$.

An important special case of measurement is *measurement in the computational basis*. In this case $M_m = |b_m\rangle\langle b_m|$ and the state after measuring the bitstring m is given simply by $|b_m\rangle$. There are other special cases of measurement such as projective measurements or positive operator-valued measure (POVM) measurements, but ultimately these are all equivalent to the application of some unitary transformation followed by a measurement in the computational basis.

> **Postulate 4.** The state $|\psi_{AB}\rangle$ of the composition of two quantum systems A and B with states $|\psi_A\rangle$ and $|\psi_B\rangle$, respectively, is given by the *tensor product* $|\psi_{AB}\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$.

The tensor product, or *Kronecker* product, of two column vectors $\mathbf{a} \in \mathbb{C}^m$ and $\mathbf{b} \in \mathbb{C}^n$ is simply the vector of mn complex elements identical to the elements of the matrix $\mathbf{ab}^\mathsf{T}$, enumerated in some particular order. The tensor product of unit-length vectors (in the ℓ_2-norm) is automatically unit-length as well. The symbol $\otimes$ is overloaded to apply to the state spaces as well: $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$.

Out of convenience, the $\otimes$ symbol is often dropped. The expressions, $|a\rangle \otimes |b\rangle$, $|a\rangle|b\rangle$ and $|a, b\rangle$ denote the same meaning.

In addition to these four postulates of quantum mechanics, a foundation for quantum computation requires a mechanism for translating the description of an algorithm into a sequence of applications of the postulates. Such a foundation is provided by the circuit model of quantum computation. In this model, a quantum algorithm describes a quantum circuit, consisting of quantum gates operating on a quantum register. Each gate operates only on a small number of qubits and is equipped with a unitary matrix describing its behavior. The operation of the entire circuit is given by the composition and tensor product of all gates' unitary matrices. Section 2.3 describes a set of quantum gates.

2.2 Density Operator Formulation

The state vector description of quantum mechanics already gives a complete mathematical framework for analyzing quantum algorithms. So why bother with another one? The answer is two-fold: First, often times the physical device that produces quantum states is not perfectly reliable and rather than always outputting a given state exactly, its output is a distribution of states. While the state vector formulation is well-equipped to handle a single state $|\psi\rangle$, it is rather cumbersome to adapt it to apply to a probability ensemble of states $\{p_i, |\psi_i\rangle\}$ where with probability p_i the state is given by $|\psi_i\rangle$. Second, in many applications one does not care about a large portion of the quantum system and only a select few qubits are relevant for the present concern. The density operator formulation offers an elegant framework to describe *partial* quantum systems.

A *density operator* or *density matrix* of a pure state $|\psi\rangle$ is given by $|\psi\rangle\langle\psi|$. The density operator of a probability ensemble of states $\{p_i, |\psi_i\rangle\}$ is given by $\sum_i p_i|\psi_i\rangle\langle\psi_i|$. While it is useful to think of density operators as somewhat redundant matrix versions of state vectors, it is worth noting that there are equivalent postulates without references to state vectors, that nevertheless provide a complete description of quantum mechanics.

> **Postulate 1.** A quantum system of k qubits is completely described by its *density matrix* $\rho \in \mathbb{C}^{2^k \times 2^k}$ such that $\mathsf{Tr}(\rho) = 1$ and such that for any vector $|\varphi\rangle \in \mathcal{H}$, $\langle\varphi|\rho|\varphi\rangle \geq 0$.

Recall that the *trace* of a square matrix, denoted by $\mathsf{Tr}(\cdot)$, is simply the sum of its diagonal elements.

> **Postulate 2.** Closed systems evolve via the action of a unitary matrix $U \in \mathbb{C}^{2^k \times 2^k}$ that sends the system's density operator ρ to $\sigma = U\rho U^\dagger$.

> **Postulate 3.** Quantum measurement is defined with respect to a collection $\{M_m\}$ of measurement operators, one for each possible event m. When applied to a system with density operator ρ, the probability of observing event m is given by $\Pr[m] = \mathsf{Tr}(M_m^\dagger M_m \rho)$ and after observing m the system is described by the density operator $\frac{M_m \rho M_m^\dagger}{\mathsf{Tr}(M_m \rho M_m^\dagger)}$. The collection $\{M_m\}$ satisfies completeness: $\sum_m M_m^\dagger M_m = I$.

> **Postulate 4.** The density operator ρ_{AB} of the composition of two pure quantum systems A and B with density operators ρ_A and ρ_B is given by their tensor product $\rho_{AB} = \rho_A \otimes \rho_B$.

A quantum system is *pure* when it is not a non-trivial probability ensemble of different states. In other words, it is pure when exactly one p_i is one and all the others are zero. This is formalized without reference to state vectors as follows: a system described by density operator ρ is pure if and only if $\mathsf{Tr}(\rho^2) = 1$.

The Kronecker product $L \otimes R \in \mathbb{C}^{km \times ln}$ of two matrices $L \in \mathbb{C}^{k \times l}$ and $R \in \mathbb{C}^{m \times n}$ is the following block matrix, whose blocks are scalar multiples of R. Here $L_{i,j}$ denotes the element at row i and column j of the matrix L with indexation starting at zero.

$$L \otimes R = \left(\begin{array}{c|c|c} L_{0,0}R & \cdots & L_{0,l-1}R \\ \hline \vdots & \vdots\vdots\vdots & \vdots \\ \hline L_{k-1,0}R & \cdots & L_{k-1,l-1}R \end{array}\right) \tag{2.1}$$

To obtain the density operator of a subsystem A of a composite system $A + B$, one applies the *partial trace operator* to "trace out" B. Let ρ_{AB} be the density operator for the system $A + B$. Then the *reduced density operator* that describes subsystem A is given by $\rho_A = \mathsf{Tr}_B(\rho_{AB})$ where for any $|a_1\rangle, |a_2\rangle \in \mathcal{H}_A$ and $|b_1\rangle, |b_2\rangle \in \mathcal{H}_B$,

$$\mathsf{Tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) = \mathsf{Tr}(|b_1\rangle\langle b_2|)|a_1\rangle\langle a_2| = \langle b_2|b_1\rangle |a_1\rangle\langle a_2| \ . \tag{2.2}$$

The density operator formulation has another selling point, namely its ability to capture the difference between two quantum states or ensembles into a single quantity called the *trace distance*. The trace distance between two states or ensembles described by density operators ρ_1 and ρ_2 is simply half of the trace norm of the difference of the matrices. The trace norm of a matrix $\rho \in \mathbb{C}^{n \times n}$ is given by $\mathsf{Tr}(\sqrt{\rho^\dagger \rho})$ and so the trace distance is $\mathrm{TD}(\rho_1, \rho_2) =$

$\frac{1}{2}\mathsf{Tr}(\sqrt{(\rho_1 - \rho_2)^\dagger(\rho_1 - \rho_2)})$. The arguments of the $\mathrm{TD}(\cdot,\cdot)$ operator can also be kets or named registers, but in this case the density operator of the given ket or the reduced density operator of the given register is meant. From a computational perspective, the trace distance, like its classical analogue, the *statistical distance*, captures the advantage of a computationally unbounded adversary in distinguishing two ensembles.

2.3 Quantum Circuits

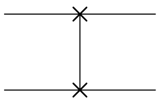
A circuit is a directed acyclic graph whose nodes are gates and whose edges are wires. The wires contain values and the gates compute a function of its input wires' values. In the case of quantum circuits, it is misleading to think of a circuit being laid out in space because that would imply that every point of the wire has the same value. Instead, quantum circuits are laid out in time. Every wire represents a qubit and these qubits may hold different quantum states at different time slices. The gates therefore have as many inputs as outputs, and come with unitary matrices that describe the effect on the affected qubits.

Out of convention, time flows forward from left to right. Single lines represent qubits or registers of qubits and double lines represent either classical information or quantum registers containing classical information.

The following list of gates covers some of the most-used quantum gates, but is by no means exhaustive. Indeed, one can build new gates by composing smaller ones.

Swap. When it is possible to identify a wire with a bit, it is tempting to draw extra wires to move the bits around and generate the right configuration of inputs to a particular gate. However, qubits are not spread out evenly across wires but are instead localized in space, even if this location is given by a wave function. In order to engineer the right configuration of input qubits to a subsequent quantum gate, it might be necessary to physically move them, or if their positions are fixed, to cause them to interact to switch values. Both operations are captured by the swap gate, whose diagram and properties are shown in Fig. 2.1.

Toffoli. A Toffoli gate, also known as a controlled-controlled-not gate, flips the third qubit if and only if the first two are set. It can be used to simulate classical and-gates and, given the availability of two qubits that are set to $|1\rangle$, classical

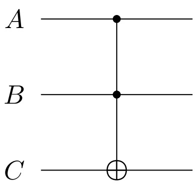


$$|a, b\rangle \mapsto |b, a\rangle$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 2.1: Swap gate: diagram, function description, and unitary matrix.

not-gates. It is therefore universal with respect to classical computations. The diagram and functional description is shown in Fig. 2.2.

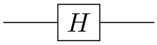


a	b	c	a	b	c
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0
	in			out	

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 2.2: Toffoli gate: diagram, truth table, and unitary matrix.

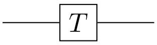
Hadamard. A Hadamard gate is the quantum analogue of a coin toss, except instead of letting the coin land and assume a definite state, face up or face down, it is left in mid-toss. It is the most straightforward way to put a qubit into a superposition of $|0\rangle$ and $|1\rangle$. The diagram and description is shown in Fig. 2.3.



$$|a\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^a}{\sqrt{2}}|1\rangle \qquad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Figure 2.3: Hadamard gate: diagram, function description, and unitary matrix.

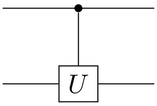
Phase Shift. In some cases, for instance in the quantum Fourier transform, it is useful to manipulate the phase of a qubit only if it is set to 1. A $\pi/8$ gate, also called a T gate, rotates this phase by $\pi/4$ radians[1], but in principle this angle can be arbitrary. The diagram and description is shown in Fig. 2.4.



$$|a\rangle \mapsto \begin{cases} |0\rangle & \text{if} \quad a = 0 \\ e^{i\pi/4}|1\rangle & \text{if} \quad a = 1 \end{cases} \qquad \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Figure 2.4: T gate: diagram, function description, and unitary matrix.

Controlled-Unitary. A controlled-unitary gate consists of a circuit, which can be any composition of gates, and a control. The circuit is applied to the indicated register if the control qubit is set; otherwise nothing happens. The diagram and description is shown in Fig. 2.5.



$$|c\rangle \otimes |a\rangle \mapsto \begin{cases} |0\rangle \otimes |a\rangle & \text{if} \quad c = 0 \\ |1\rangle \otimes (U|a\rangle) & \text{if} \quad c = 1 \end{cases} \qquad \left(\begin{array}{c|c} I & 0 \\ \hline 0 & U \end{array}\right)$$

Figure 2.5: Controlled-unitary gate: diagram, function description, and unitary (block) matrix.

Measurement. Measurement is how information is extracted from the quantum system. Upon measurement, the state collapses to classical information in accordance with the measured value; this explains the double arrow. The diagram is shown in Fig. 2.6.

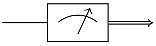


Figure 2.6: Measurement gate: diagram.

2.4 General Remarks

Complexity. The circuit model of quantum computation suggests three simple characterizers of complexity of quantum algorithms, namely a) required number of qubits, b) circuit size, and c) circuit depth. While these indicators do offer a good first approximation, they can be misleading because in practice quantum

[1] Or alternatively, it rotates the phase associated with $|0\rangle$ by $-\pi/8$ radians and the phase associated with $|1\rangle$ by $+\pi/8$ radians, hence the name.

computation is inherently noisy: qubits decohere over time and gates only apply an approximation of the unitary matrix they purport to apply. Consequently, there is a distinction between physical qubits, referring to the physical particles that contain the actual noisy quantum amplitudes, and logical qubits, the units of quantum information in the next layer of abstraction. Practical construction of quantum computers will involve quantum error correction performed by the physical qubits to simulate clean, perfect qubits. Depending on the substrate used for the physical layer, the overhead of quantum error correction can be several orders of magnitude.

Oracles. Quantum computers can compute any function classical computers can, despite the requirement that quantum computations be invertible. It turns out there is a rather simple trick to turn any computable function into a function that is computable reversibly. Let $\mathsf{H} : \{0,1\}^* \rightarrow \{0,1\}^\star$ be a computable function from bitstrings of any length to bitstrings of any length. Then the unitary transformation U_H that operates on registers Q and R (possibly short for "query" and "response") and sends $|q,r\rangle$ to $|q, r \oplus \mathsf{H}(q)\rangle$ is invertible — indeed, it is its own inverse. This is the standard construction of oracle-algorithms, *i.e.*, quantum algorithms that have black box access to a subprocedure with a given description but that is unknown to the algorithm itself. For example, H may represent a hash function that is modeled as a random function, and like in the classical case, the intuition that the algorithm knows nothing about the description of H is captured by the oracle interface. The algorithm sends two of its registers to the black box, the black box applies its unitary transformation, and the two registers are sent back.

Measurement and entanglement. Measurement is indistinguishable from entanglement with qubits that are traced out. To see this, consider the simple example sketched in Fig. 2.7. Consider the effect of the left and right hand side circuits on the register A, which at the start contains the state $\alpha|0\rangle + \beta|1\rangle$. In the circuit on the left, the measurement collapses the state to $|0\rangle$ with probability $||\alpha||^2$ and $|1\rangle$ with probability $||\beta||^2$, concisely described as the density matrix $||\alpha||^2|0\rangle\langle 0| + ||\beta||^2|1\rangle\langle 1|$. The circuit on the right sends the input

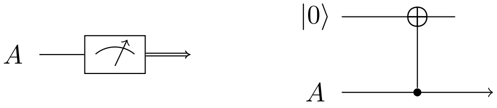


Figure 2.7: Equivalence between measurement and outside qubits.

state $\alpha|0,0\rangle + \beta|1,0\rangle$ to $\alpha|0,0\rangle + \beta|1,1\rangle$, which may be described by the density matrix $\rho = \alpha\alpha^\dagger|0,0\rangle\langle 0,0| + \beta\beta^\dagger|1,1\rangle\langle 1,1|$. Tracing out the top qubit ("T") gives $\mathsf{Tr}_T(\rho) = \|\alpha\|^2|0\rangle\langle 0| + \|\beta\|^2|1\rangle\langle 1|$, or exactly the same density operator that comes out of the circuit on the left. This observation allows one to transform any quantum circuit that contains measurement gates into one with more qubits but whose measurement gates are located at the end.

No Cloning Theorem. It is impossible to clone an unknown quantum state. This can be immediately derived from the unitarity of operators. Suppose to the contrary that there is a unitary matrix U that maps $|\phi\rangle \otimes |0\rangle \mapsto |\phi\rangle \otimes |\phi\rangle$ for all $|\phi\rangle \in \mathcal{H}$. Then choose another ket $|\psi\rangle \in \mathcal{H}$ and observe that $\langle\phi|\psi\rangle = (\langle\phi|\otimes\langle 0|)(|\psi\rangle\otimes|0\rangle) = (\langle\phi|\otimes\langle 0|)U^\dagger U(|\psi\rangle\otimes|0\rangle) = (\langle\phi|\otimes\langle\phi|)(|\psi\rangle\otimes|\psi\rangle) = \langle\phi|\psi\rangle^2$. The equation $\langle\phi|\psi\rangle = \langle\phi|\psi\rangle^2$ cannot be satisfied for all kets $|\phi\rangle, |\psi\rangle \in \mathcal{H}$, take for example a pair of kets that are 45° apart. Therefore, such a unitary matrix cannot exist. A slightly more complex argument shows that the same holds for any combination of unitary transformations and measurement.

Nevertheless, in some special cases, cloning information is possible. For example the unitary matrix that maps $|a,b\rangle \mapsto |a, b \oplus a\rangle$ copies the bitstring a when $b = 0$. However, the point is that the left-hand register's reduced density operator changes as a result of this operation. What is being copied is the classical bitstring a, and not the quantum ket $|a\rangle$. It is possible to copy classical information, but it is impossible to copy quantum information. Any operation that would extract information from an unknown quantum state necessarily changes it.

Chapter 3

Provable Security

How does one prove that a cryptosystem is secure? To answer that question, it must first be clear what is meant by the opposite, *i.e.*, what makes a cryptosystem insecure. Specifically, one must define which events constitute a security violation, or *attack*. Additionally, one must specify the adversarial model, *i.e.*, the class of adversaries the security statement is supposed to cover. Given these two elements, one can proceed to state propositions such as "for all adversaries that fit the model, the attack fails" and prove them by showing that their negations imply a contradiction.

Often times the adversarial model contains only polynomial-time algorithms; this restriction captures the intuition that an attack should be efficient in order to be valid. In this case a security statement and proof can additionally rely on a *computational hardness assumption*. The derived contradiction then shows that either an efficient attack does not exist, or else that the considered hardness assumption is false. If that assumption pertains to a well-established mathematical problem that is and has been studied independently from its cryptographic applications, then the hard-earned belief in that problem's hardness is leveraged in support of the cryptosystem's security.

The adversarial model in the context of post-quantum cryptography is restricted to polynomial-time algorithms capable of performing quantum computations offline. That is to say, any messages exchanged with other participants in the protocol that is under attack consist of classical information. In contrast, the computations between interactions may be quantum, and the attacker may even keep quantum memory across interactions. Any function can be evaluated in a superposition of inputs, provided that the attacker possesses the complete function description.

A security definition takes the form of program code[1] describing either a two-player game between the adversary and a challenger or a protocol in which the adversary is one of many participants [22]. The adversary itself is treated as a black box; its code is not defined and it is only invoked abstractly in the way that a subprocedure is invoked. The adversary may retain a secret and even quantum state across invocations; in this case the program code must record it and pass it as an argument to the adversary at the next call. The program code outputs a single bit, indicating whether the attack was successful (1) or not (0). The cryptosystem is secure if the program outputs 1 only with a negligible probability, over all the random coins involved.

A security proof then consist of a sequence of patches to the program code. Each patch is accompanied by an argument showing that the output distribution changes only by a negligible amount. After all patches have been applied, the program code is identical to the description of a problem whose hardness is assumed, preferably up front. The various stages of the program code are referred to as games; this patchwork methodology of security proofs in cryptography is known as a sequence of games approach [125].

3.1 Asymptotic and Concrete Notions.

The previous description of security definitions and proofs make reference to the notion of *negligible* quantities. Formally, a function $\mathsf{negl} : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ is negligible if and only if it drops faster than any polynomial's reciprocal. Conversely, a function $\mathsf{noti} : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ is *noticeable* if it drops slower than some positive polynomial's reciprocal. Formally:

$$\forall p(x) \in \mathbb{R}_{\geq 0}[x] \,.\, \exists N \in \mathbb{N} \,.\, \forall n > N \,.\, \mathsf{negl}(n) < \frac{1}{p(n)} \ , \tag{3.1}$$

$$\exists p(x) \in \mathbb{R}_{\geq 0}[x] \,.\, \exists N \in \mathbb{N} \,.\, \forall n > N \,.\, \mathsf{noti}(n) > \frac{1}{p(n)} \ . \tag{3.2}$$

[1] Actually, many security definitions in the literature do not present code but a complex probability expression. However, without loss of generality, any security definition can be translated into pseudocode.

Additionally, a probability is *overwhelming* if its distance from 1 is negligible. In the context of security definitions and proofs, the argument of noticeable and negligible functions is generally speaking the *security parameter* λ.

While very intuitively accessible, asymptotic security does have its disadvantages. For instance, the square root of a negligible quantity is still negligible, but a 2^{-128} probability of successful attack is a far greater concern than if the same probability is only 2^{-256}. The previous definitions can be used to capture *whether* a cryptosystem is secure, but we often wish to know *how much* security it offers. The *concrete security framework*, pioneered by Bellare and Rogaway [18] aims to answer this question by capturing security losses in explicit and exact terms called *insecurity functions* that grow with the resources expended by the adversary and capture the amount of security lost as a function of these resources.

For instance, the one-wayness insecurity function, which is defined as $\mathsf{InSec}_{\mathsf{H}}^{\mathsf{OW}}(Q) \triangleq \mathsf{max}_{\mathsf{A}} \Pr[\mathsf{H}(\mathsf{A}^{\mathsf{H}}(\mathsf{H}(x))) = \mathsf{H}(x) \,|\, x \xleftarrow{\$} \{0,1\}^\lambda]$, captures the maximum success probability across all adversaries A with Q queries and unbounded time to find an inverse of $\mathsf{H}(x)$ under the function H, provided as an oracle. Here x is a random input and $\mathsf{H}(x)$ is its matching image, and the adversary also wins if he outputs a different preimage $x' \neq x$ as long as $\mathsf{H}(x') = \mathsf{H}(x)$. The acronym OW stands for the one-wayness game, which is captured by the probability expression. When $\mathsf{H} : \{0,1\}^\lambda \to \{0,1\}^\lambda$ is a random function and only classical queries are allowed, then $\mathsf{InSec}_{\mathsf{H}}^{\mathsf{OW}}(Q) = (Q+1)/2^\lambda$.

Suppose there is a sequence-of-games proof that involves two games, G_1 and G_2, and suppose moreover that the event $\mathcal{E}$, "$\mathsf{G}_1^{\mathsf{A}}$ outputs 1 but $\mathsf{G}_2^{\mathsf{A}}$ outputs 0" occurs only when the adversary queries H on a preimage to $\mathsf{H}(x)$. Then there is an extractor algorithm E that simulates G_1 or G_2 only to look at the list of queries made by A to H and resulting responses; if this list contains a preimage to $\mathsf{H}(x)$ then E outputs it and halts, and if it does not then E outputs $\perp$ and halts. Naturally, E's success probability is bounded by $\mathsf{InSec}_{\mathsf{H}}^{\mathsf{OW}}(Q)$ — but this is also a bound on the probability of event $\mathcal{E}$. This translates to a concrete bound on the difference in output distributions of G_1 and G_2, namely

$$|\Pr[\mathsf{G}_1 \text{ outputs } 1] - \Pr[\mathsf{G}_2 \text{ outputs } 1]| \leq \mathsf{InSec}_{\mathsf{H}}^{\mathsf{OW}}(Q) \ . \tag{3.3}$$

3.2 Functionalities

A public key functionality follows a syntax that describes its usage. The purpose of this syntax is to abstract away the mathematical foundations that make the cryptosystem work and that make it secure. Additionally, the security definition is presented in terms of the provided syntax. Here are some of the most basic public key functionalities along with common security definitions. This list is far from exhaustive.

3.2.1 Digital Signature Scheme.

A digital signature scheme allows a user to bind himself to a document in a way that makes later repudiation impossible, similar to physically signing a contract or note except digitally. A digital signature simultaneously provides authenticity and integrity: the source of the signature must be the holder of the secret key that matches the public key, and the signature is not valid for any other message than the one that was signed.

A digital signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is a triple of polynomial-time algorithms with the following properties.

- KeyGen takes a security level λ (provided in unary notation); and outputs two values: sk and pk, the secret key and the public key, respectively.
- Sign takes a secret key sk and a document d; and outputs a signature sig.
- Verify takes a public key pk, a document d, and a signature sig; and outputs 0 or 1.
- The scheme is *correct*, *i.e.*, whenever a secret key is used to sign a document, the resulting signature is valid with respect to the matching public key with overwhelming probability. Formulaically:

$$\forall d \in \{0,1\}^* \, . \tag{3.4}$$

$$\Pr[\mathsf{Verify}(pk, d, \mathsf{Sign}(sk, d)) = 1 \mid sk, pk \leftarrow \mathsf{KeyGen}(1^\lambda)] \geq 1 - \mathsf{negl}(\lambda) \ .$$

Realistic security definitions involve *chosen message attacks (CMA)*, *i.e.*, the adversary A is allowed to query a signature oracle on a message d of his choosing. This oracle models the capacity of an attacker to trick the user into signing something.

In the *universal unforgeability under chosen message attack (UUF-CMA)* game, the adversary is presented with a single message that he must find a signature to. The signature oracle refuses to answer if this message was queried.

In the *existential unforgeability under chosen message attack (EUF-CMA)* game, the adversary gets to choose which message he forges a signature for. However, if this message was one of the queries to the signature oracle then the adversary will be penalized. In other words: he only wins if he forges a signature on an entirely new message.

Game 3.1: UUF-CMA

```
$sk, pk \leftarrow \mathsf{KeyGen}(1^\lambda)$
$m \xleftarrow{\$} \{0,1\}^{\mathsf{poly}(\lambda)}$
define $\mathsf{S}(d)$ as:
    if $d = m$ then:
        return $\bot$
    else:
        return $\mathsf{Sign}(sk, d)$
$sig \leftarrow \mathsf{A}(pk, m)$
return $\mathsf{Verify}(pk, m, sig)$
```

Game 3.2: EUF-CMA

```
$sk, pk \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathcal{D} \leftarrow \varnothing$
define $\mathsf{S}(d)$ as:
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{d\}$
    return $\mathsf{Sign}(sk, d)$
$m, sig \leftarrow \mathsf{A}(pk)$
return $[\![\mathsf{Verify}(pk, m, sig) = 1 \wedge m \notin \mathcal{D}]\!]$
```

Security is defined with respect to an UUF-CMA or EUF-CMA insecurity function, namely by requiring them to be negligible functions. Formally, the definitions are as follows.

Definition 1 (UUF-CMA security of digital signature schemes)**.** *A digital signature scheme $\mathcal{S}$ is secure in the UUF-CMA model if for all polynomial-time adversaries* A *the insecurity* $\mathsf{InSec}_{\mathcal{S}}^{\mathsf{UUF\text{-}CMA}}(\mathsf{A}) \triangleq \Pr[\mathsf{UUF\text{-}CMA}^{\mathsf{A}}(1^\lambda) = 1]$ *is negligible in λ,* i.e.*,* $\mathsf{InSec}_{\mathcal{S}}^{\mathsf{UUF\text{-}CMA}}(\mathsf{A}) \leq \mathsf{negl}(\lambda)$*, where* UUF-CMA *is shown in Fig. 3.1.*

Definition 2 (EUF-CMA security of digital signature schemes)**.** *A digital signature scheme $\mathcal{S}$ is secure in the EUF-CMA model if for all polynomial-time adversaries* A *the insecurity* $\mathsf{InSec}_{\mathcal{S}}^{\mathsf{EUF\text{-}CMA}}(\mathsf{A}) \triangleq \Pr[\mathsf{EUF\text{-}CMA}^{\mathsf{A}}(1^\lambda) = 1]$ *is negligible in λ,* i.e.*,* $\mathsf{InSec}_{\mathcal{S}}^{\mathsf{EUF\text{-}CMA}}(\mathsf{A}) \leq \mathsf{negl}(\lambda)$*, where* EUF-CMA *is shown in Fig. 3.2.*

Two other chosen message attack games go by the acronym SUF-CMA. The *selective unforgeability under chosen message attack* game is a hybrid between UUF-CMA and EUF-CMA whereby the adversary is allowed to choose the message he forges a signature for, but this message must be fixed before the signature oracle is queried. A universal forger implies a selective forger, which in turn implies an existential forger. The *strong (existential) unforgeability under chosen message attack* game is a relaxation of EUF-CMA where the list $\mathcal{D}$ records the message-and-signature pairs of all queries, instead of just the messages. The strong unforgeability adversary wins if he produces a new signature, possibly on an already-signed message; in contrast, the existential unforgeability adversary must produce a new message with signature. An existential forger implies a strong existential forger, meaning that the strong unforgeability game is the strongest notion. However, it is not clear that this stronger notion is necessary; most of the time, EUF-CMA is sufficient. For instance, the NIST call for proposals states that EUF-CMA captures what will be considered relevant attacks [75]. Nevertheless if an attack is discovered that works only in the strong unforgeability model, it will be a cause for concern.

3.2.2 Key Encapsulation Mechanism.

Public key encryption is much more expensive than symmetric encryption, and consequently public key encryption is usually only used for securely transporting symmetric keys. (One important exception is homomorphic encryption.) If the purpose is key establishment anyway, then transporting keys may be overkill; a shared symmetric key may also be computed from mutual protocol contributions so long as the passive adversary cannot compute it also. The *key encapsulation mechanism (KEM)* formalism captures this rigorously. A key encapsulation mechanism $\mathcal{K} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is a triple of polynomial-time algorithms with the following properties.

- KeyGen takes a security level λ (provided in unary notation); and outputs two values, sk and pk, the secret and the public key, respectively.
- Enc ("encapsulate") takes a public key pk and generates a ciphertext $ctxt$ and a symmetric key k.
- Dec ("decapsulate") takes a secret key sk and a ciphertext $ctxt$ and outputs a symmetric key k.
- The mechanism is *correct*, *i.e.*, whenever a secret key decapsulates a ciphertext, decapsulation produces the same symmetric key that was produced during encapsulation provided it was encapsulated with the matching public key. Formulaically:

$$\Pr[\mathsf{Dec}(sk, ctxt) = k \,|\, sk, pk \leftarrow \mathsf{KeyGen}(1^\lambda);\ ctxt, k \leftarrow \mathsf{Enc}(pk)] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{3.5}$$

The security notions for KEMs are indistinguishability games: the adversary has to decide whether a given ciphertext decapsulates to the given key, or whether the ciphertext decapsulates to something else and the given key was drawn uniformly at random from the symmetric key space SKSpace. The adversary is in possession of the public key and therefore he can generate ciphertexts as he pleases. The label "chosen-plaintext attack" is confusing in the context of KEMs because there are no plaintexts. Nevertheless the notion is analogous to the likewise-named notion for public key encryption (PKE) schemes. In the chosen-ciphertext attack, the adversary has the additional capacity to query a decapsulation oracle.

Game 3.3: IND-CPA for KEMs

```
1. $sk, pk \leftarrow \mathsf{KeyGen}(1^\lambda)$
2. $b \xleftarrow{\$} \{0,1\}$
3. $k_0 \xleftarrow{\$} \mathsf{SKSpace}$
4. $c, k_1 \leftarrow \mathsf{Enc}(pk)$
5. $b' \leftarrow \mathsf{A}(pk, k_b, c)$
6. return $[\![b = b']\!]$
```

Game 3.4: IND-CCA for KEMs

```
1. $sk, pk \leftarrow \mathsf{KeyGen}(1^\lambda)$
2. $b \xleftarrow{\$} \{0,1\}$
3. $k_0 \xleftarrow{\$} \mathsf{SKSpace}$
4. $c, k_1 \leftarrow \mathsf{Enc}(pk)$
5. define $\mathsf{D}(q)$ as:
6.     if $q = c$ return $\perp$
7.     else return $\mathsf{Dec}(sk, q)$
8. $b' \leftarrow \mathsf{A}^{\mathsf{D}(\cdot)}(pk, k_b, c)$
9. return $[\![b = b']\!]$
```

Definition 3 (IND-CPA security of KEMs). *A KEM $\mathcal{K}$ is secure in the IND-CPA model if for all polynomial-time adversaries* D *the advantage* $\mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CPA}}(\mathsf{D}) \triangleq |\Pr[\mathsf{IND\text{-}CPA}^{\mathsf{D}}(1^\lambda) = 1] - \frac{1}{2}|$ *is negligible in* λ, i.e., $\mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CPA}}(\mathsf{D}) \leq \mathsf{negl}(\lambda)$*, where the game* IND-CPA *is shown in Game 3.3.*

Definition 4 (IND-CCA security of KEMs). *A KEM $\mathcal{K}$ is secure in the IND-CCA model if for all polynomial-time adversaries* D *the advantage* $\mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathsf{D}) \triangleq |\Pr[\mathsf{IND\text{-}CCA}^{\mathsf{D}}(1^\lambda) = 1] - \frac{1}{2}|$ *is negligible in* λ, i.e., $\mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathsf{D}) \leq \mathsf{negl}(\lambda)$*, where the game* IND-CPA *is shown in Game 3.4.*

3.2.3 Zero-Knowledge Proofs.

Zero-knowledge proofs are an indispensable tool in the design of cryptographic protocols because they enable one participant to prove to others that his protocol contribution is honest and correctly formed despite its encryption. The term *zero-knowledge* refers to the fact that the verifier, after engaging with and being convinced by the prover, has obtained zero knowledge about the the claim that is proven beyond the fact that it is true. One can think of a zero-knowledge proof as the encryption of a proof — it is no less valid, but in contrast to proofs in mathematics, even mathematicians cannot decipher them. Non-interactive zero-knowledge proofs form a popular design methodology to generate digital signature schemes.

Formally, an *interactive proof system* $\Pi = (\mathsf{P}, \mathsf{V})$ for a language $\mathcal{L} \in \mathbf{NP}$ is a protocol between two polynomial time algorithms, called the *prover* P and *verifier* V, respectively, both of which receive a string $\ell \in \{0,1\}^*$ for input. The prover has an additional secret input, namely the witness $v \in \{0,1\}^*$ that certifies that $\ell \in \mathcal{L}$, *i.e.*, $\mathcal{R}_{\mathcal{L}}(\ell, v) = 1$. An execution of the protocol is denoted by $\langle \mathsf{V}(\ell) \leftrightarrow \mathsf{P}(v, \ell)\rangle$, the verifier's output by $\mathsf{out}_{\mathsf{V}}(\langle \mathsf{V}(\ell) \leftrightarrow \mathsf{P}(v, \ell)\rangle)$, and this output is 1 if he *accepts* and 0 if he *rejects*. The *transcript* $T \leftarrow \langle \mathsf{P}(v, \ell) \leftrightarrow \mathsf{V}(\ell)\rangle$ consists of all messages sent between the two parties. A *zero-knowledge proof system* satisfies three properties:

1. *Completeness.* For every $\ell \in \mathcal{L}$ and matching witness v, P convinces V with high probability:

$$\begin{aligned} &\forall \ell \in \{0,1\}^*, v \in \{0,1\}^* \,.\, \mathcal{R}_{\mathcal{L}}(\ell, v) = 1 \\ &\qquad \Longrightarrow \Pr[b = 1 \,|\, b \leftarrow \mathsf{out}_{\mathsf{V}}(\langle \mathsf{V}(\ell) \leftrightarrow \mathsf{P}(v, \ell)\rangle)] \geq 1 - \varepsilon \ . \end{aligned} \tag{3.6}$$

In this expression ε represents the *completeness error* and should be a negligible function of $|\ell|$.

2. *Soundness.* For every $\ell \notin \mathcal{L}$ no prover B is likely to convince the verifier:

$$\forall \ell \notin \mathcal{L} \,.\, \forall \mathsf{B} \,.\, \Pr[b = 1 \,|\, b \leftarrow \mathsf{out}_{\mathsf{V}}(\langle \mathsf{B}(\ell) \leftrightarrow \mathsf{V}(\ell)\rangle)] \leq \sigma \ . \tag{3.7}$$

The quantity σ represents the *soundness error* and should be small but not necessarily negligible.

2*. *Witness-extractability, or knowledge-soundness.* In addition to being a zero-knowledge proof system, Π is a proof system for *proofs of knowledge* if there is a polynomial-time extractor machine E who, given black-box access to any sufficiently successful prover B, can compute the witness v

with noticeable probability $\xi \geq \mathsf{noti}(|\ell|)$.

$$\exists \mathsf{E} \,.\, \forall \mathsf{B} \,.\, \Pr[\mathsf{out}_{\mathsf{V}}(\langle \mathsf{V}(\ell) \leftrightarrow \mathsf{B}(\ell) \rangle) = 1] \geq \varsigma \\ \implies \Pr[\mathcal{R}_{\mathcal{L}}(\ell, v) = 1 \,|\, v \leftarrow \mathsf{E}^{\mathsf{B}}(\ell)] \geq \xi \ . \quad (3.8)$$

Phrased differently, if the *probability of extraction* ξ is not noticeable, then B's success probability is upper-bounded by the *knowledge error* ς, which should also be small but not necessarily negligible.

In the post-quantum setting, B and E^{B} are allowed to be quantum algorithms. Proof systems satisfying this lifted property of *quantum-witness-extractability* generate *quantum proofs of knowledge* (QPoK) [138]. Black box oracle access for quantum computers is defined as follows. The prover's computations before and between sending and receiving messages are described by a sequence of invertible quantum circuits $\mathfrak{B}_1, \ldots, \mathfrak{B}_N$ acting on a secret quantum register S which is initially set to some quantum input $|\Psi\rangle$. The extractor can apply these circuits as well as their inverses but has no access to S. All interaction happens by writing and reading information to and from a designated message register. These messages follow the format of the proof system and are thus classical.

3. *Honest-verifier zero-knowledge.* There is a polynomial-time simulator S capable of producing a transcript $T \leftarrow \mathsf{S}(\ell)$ of the protocol without knowledge of the witness v such that T is indistinguishable from authentic transcripts. Indistinguishability is satisfied when all polynomial-time distinguishers D have at most a negligible advantage, *i.e.*, $\mathsf{Adv}_{\Pi}^{\mathsf{ZK}}(\mathsf{D}) \leq \mathsf{negl}(|\ell|)$, where

$$\mathsf{Adv}_{\Pi}^{\mathsf{ZK}}(\mathsf{D}) \triangleq \big|\Pr[\mathsf{D}(T) = 1 \,|\, T \leftarrow \langle \mathsf{P}(v, \ell) \leftrightarrow \mathsf{V}(\ell) \rangle] - \\ \Pr[\mathsf{D}(T) = 1 \,|\, T \leftarrow \mathsf{S}(\ell)]\big| \ . \quad (3.9)$$

The protocol additionally satisfies *special* honest-verifier zero-knowledge if the simulator S cannot choose the verifier's messages. Specifically, the messages from the verifier in the transcript $T \leftarrow \mathsf{S}(\ell, c_1, \ldots, c_N)$ are exactly $c_1, \ldots, c_N$, where N is the number of messages sent by the verifier.

In contrast to digital signature schemes, PKEs, and KEMs, zero-knowledge proof systems have *two* insecurity functions: the *zero-knowledge advantage* $\mathsf{Adv}_{\Pi}^{\mathsf{ZK}}(\mathsf{D})$, and the *soundness error* σ or, when applicable, the *knowledge error* ς. Both functions must be negligible in a practical instantiation. Proof systems with non-negligible soundness or knowledge errors can still be useful as a building block to build larger proof systems where these quantities are negligible.

3.3 Security Reductions

A lot of the pieces are in place for a demonstration proving the security of the RSA signature scheme of Example 2. Recall that this cryptosystem derives security from the hardness of inverting the RSA function $f_e : \mathbb{Z}_n \rightarrow \mathbb{Z}_n, x \mapsto x^e \mathsf{mod}\, n$. Moreover, it requires a hash function $\mathsf{H} : \{0,1\}^* \rightarrow \mathbb{Z}_n$, which for the purposes of the proof is modeled as a *random oracle*, *i.e.*, a function drawn uniformly at random from $\{f \mid f : \{0,1\}^* \rightarrow \mathbb{Z}_n\}$ and presented as an oracle. The proof of Example 3 is a reformulation of that of Coron [41]. It features a single game transition: Game_1 is the EUF-CMA game, and Game_2 is the RSA inversion problem. A more complex proof will have several more games. Nevertheless, this example suffices to illustrate many relevant aspects.

This proof also serves as an excellent opportunity to introduce the syntax and semantics of the python-like *dictionary* notion, which I use elsewhere as well. Formally, a dictionary is a variable representing a list of (*key*, *value*) pairs such that for every *key* there is at most one matching *value*. If $\mathcal{D}$ is a dictionary, then $\mathcal{D}[k]$ represents the unique value v such that (k, v) is in this list. The notation $\mathcal{D}[k] \leftarrow v$, or $\mathcal{D}[k] \xleftarrow{\$} \mathcal{S}$, inserts the pair (k, v) where v is either given explicitly or drawn uniformly at random from $\mathcal{S}$, into the list. If necessary, the prior element where k was the key is removed. The set of key values is denoted by $\mathcal{D}.\mathsf{keys}()$, thus enabling a concise expression to determine if the list contains a pair where k is the key: $k \in \mathcal{D}.\mathsf{keys}()$.

Several remarks about the theorem and proof of Example 3 are in order.

- *Running time.* The proof ignores the running time of the simulator B, but this is actually a crucial concern. If B's running time were much larger than that of A, it could be argued that B's ability to invert the RSA function was the result of his larger running time and not of A's capacity to break the signature scheme, thus nullifying the argument for security. Nevertheless, it is clear from inspection of Example 3 that B incurs only a small linear overhead over the running time of A. In other security proofs, the running time may require explicit attention.
- *Classical random oracle model.* The hash function H is modeled as a random oracle. However, any given concrete hash function used in practice

Example 3. RSA Signature Scheme — Security Proof

Theorem 1. *Let* A *be a winning adversary against the RSA Signature Scheme of Example 2 with* Q_s *signature queries and* Q_h *hash queries in the EUF-CMA and random oracle models. Then there is an algorithm* B *such that*

$$\mathsf{InSec}^{\mathsf{EUF\text{-}CMA}}_{RSA\text{-}sig}(\mathsf{A}) \le \left(1+\frac{1}{Q_s}\right)^{Q_s} \cdot (Q_s+1) \cdot \Pr[\mathsf{B}^{\mathsf{A}}(x,n,e) = f^{-1}(x)] \ . \tag{3.10}$$

(Note that this bound is independent of Q_h*.)*

Proof. The input to the algorithm B is the RSA public key (n, e) as well as the image x for which B must find $f_e^{-1}(x)$, which is its output. The strategy B employs is as follows: he runs the EUF-CMA game and invokes A as part of it, thus enabling him to leverage A's winning probability to his own advantage. A is allowed to make queries to a signing oracle $\mathsf{S}(\cdot)$ and to a random oracle $\mathsf{H}(\cdot)$; B must answer these queries without the matching secret key to the public key (n, e).
To overcome this difficulty, B maintains two dictionaries $\mathcal{G}$ and $\mathcal{H}$, both of which are initially empty. $\mathcal{H}$ represents the query-response pairs of the random oracle H, whereas $\mathcal{G}$ stores, for every such (*query*, *response*) pair, either the pair (*query*, f^{-1}(*response*)) or (*query*, $x \cdot f^{-1}$(*response*)) where f is the RSA function and x is the given image. Additionally, B maintains a set $\mathcal{D}$ which is also initially empty, but later represents the set of documents queried by A to the signing oracle S. B answers A's oracle queries as follows. The parameter p will be determined later.

```
define $\mathsf{H}(q)$ as:
    if $q \notin \mathcal{H}$.keys() then:
        $\mathcal{G}[q] \xleftarrow{\$} \mathbb{Z}_n$
        $u \xleftarrow{\$} [0;1]$
        if $u > p$ then:
            $\mathcal{H}[q] \leftarrow x \cdot \mathcal{G}[q]^e \bmod n$
        else:
            $\mathcal{H}[q] \leftarrow \mathcal{G}[q]^e \bmod n$
    return $\mathcal{H}[q]$
```

```
define $\mathsf{S}(d)$ as:
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{d\}$
    if $d \notin \mathcal{G}$.keys() then:
        $\mathcal{G}[d] \xleftarrow{\$} \mathbb{Z}_n$
        $\mathcal{H}[d] \leftarrow \mathcal{G}[d]^e \bmod n$
    return $\mathcal{G}[d]$
```

Queries to the random oracle H are answered in accordance with the dictionary $\mathcal{H}$. If a new query-response pair is needed for the query q,

Example 3 — continued

then B samples $\mathcal{G}[q]$ at random and sets $\mathcal{H}[q]$ to $x \cdot \mathcal{G}[q]^e \bmod n$ with probability $1 - p$ and to $\mathcal{G}[q]^e \bmod n$ with probability p. The exception is when a new query-response pair is needed in the course of answering a signature oracle query; in this case $\mathcal{H}[q]$ is set to $\mathcal{G}[q]^e \bmod n$ with certainty.

At this point, B simulates the adversary by invoking $\mathsf{A}^{\mathsf{H},\mathsf{S}}(n, e)$, *i.e.*, simulating A on the input (n, e) with oracle access to $\mathsf{H}(\cdot)$ and $\mathsf{S}(\cdot)$ as described above. The simulation fails if the adversary A first makes a random oracle query that triggers line 6 and then queries the signature oracle on the same input, because the returned signature will be invalid. Call this event $\mathcal{F}$. Since the outputs of $\mathsf{H}(\cdot)$ are uniform, $\Pr[\neg\mathcal{F}] = p^{Q_s}$ and

$$\Pr[\mathsf{A}\, \mathit{success} \wedge \neg\mathcal{F}] = \mathsf{InSec}^{\mathsf{EUF\text{-}CMA}}_{\mathit{RSA\text{-}sig}}(\mathsf{A}) \cdot p^{Q_s} \ . \tag{3.11}$$

In the event of adversarial success, A outputs a pair (m, sig) such that $\mathsf{H}(m) = \mathit{sig}^e \bmod n$ and $m \notin \mathcal{D}$. Without loss of generality, $m \in \mathcal{H}.\mathsf{keys}()$, because otherwise B can query $\mathsf{H}(m)$ himself. At the point of A's termination, the condition $m \notin \mathcal{D}$ implies that $\mathcal{H}[m]$ must have been set by line 6 or line 8 of $\mathsf{H}(q)$. If it was line 6 then the returned value sig satisfies $\mathit{sig}^e = x \cdot \mathcal{G}[m]^e \bmod n$ or equivalently, $\mathit{sig}^e \cdot (\mathcal{G}[m]^{-1})^e \bmod n$, meaning that $\mathit{sig} \cdot \mathcal{G}[m]^{-1} = f_e^{-1}(x)$. Therefore, B outputs this value and then his success probability is bounded via

$$\Pr[\mathsf{A}\, \mathit{success} \wedge \neg\mathcal{F}] = \Pr[\neg\mathcal{F} \wedge \mathsf{H}(m) = \mathit{sig}^e] \tag{3.12}$$

$$= \Pr[\neg\mathcal{F} \wedge \mathit{sig}^e = x \cdot \mathcal{G}[m]^e] + \Pr[\neg\mathcal{F} \wedge \mathit{sig}^e = \mathcal{G}[m]^e] \tag{3.13}$$

$$= \frac{1}{1-p} \cdot \Pr[\mathsf{B}^{\mathsf{A}}(x, n, e) = f^{-1}(x)] \ . \tag{3.14}$$

The last equality holds because, as the outputs of H are identically distributed, the event $\mathit{sig}^e = \mathcal{G}[m]^e$ is $p/(1-p)$ times as likely as the event $\mathit{sig}^e = x \cdot \mathcal{G}[m]^e$. The latter implies that B wins. Therefore,

$$\mathsf{InSec}^{\mathsf{EUF\text{-}CMA}}_{\mathit{RSA\text{-}sig}}(\mathsf{A}) = \left(\frac{1}{p}\right)^{Q_s} \cdot \frac{1}{1-p} \cdot \Pr[\mathsf{B}^{\mathsf{A}}(x, n, e) = f^{-1}(x)] \ . \tag{3.15}$$

The theorem statement follows from choosing the value for $p \in [0; 1]$ that minimizes this expression, *i.e.*, $p = \frac{Q_s}{Q_s+1} = 1 - \frac{1}{Q_s+1}$. □

cannot be chosen uniformly at random from the space of functions with the right domain and range. From a rigorous perspective then, the proof is proving the wrong thing: it is proving the security of some abstract construction rather than the one that is used in practice. However, this sleight of hand is justifiable to some extent if the best attack on the given hash function is no better than a generic attack on a real random oracle. Also, in the proof, the random oracle is simulated by *a*) maintaining a list of query-response pairs, and *b*) sampling the responses lazily, *i.e.*, when they are needed and not sooner. While both points lead to a range of effective arguments in security proofs, they rely on the query consisting of classical information only. In a post-quantum context where the adversary can make queries in superposition and receive superposition responses in return, these techniques fail. Section 3.4 elaborates on the *quantum random oracle model*, which addresses this concern.

- *Tightness.* The bound is not tight: there is a gap between the best possible attack and the given insecurity, owing to the factor Q_s. If the signature scheme is expected to generate, say, $Q_s = 2^{28}$ signatures, and if the RSA modulus takes some 2^{128} time steps to factor, then the bound shows "only" 100 bits of security. Nevertheless, as far as security bounds go, a linear security degradation is acceptable. A square-root degradation is not uncommon, particularly in the context of post-quantum provable security or as a result of the Forking Lemma [19]. A major task in provable security is to find better proof techniques to establish tighter bounds, or to tweak constructions so as to enable a tighter bound. Nevertheless, loose bounds do not indicate the existence of an attack that meets the bound but certainly do indicate the nonexistence of attacks running in polynomial time, so even loose bounds are still asymptotically sound security bounds.

- *Random self-reducibility.* The proof relies on the fact that the returned results of the hash oracle $\mathsf{H}(\cdot)$ are all uniformly distributed. While this is true, it derives from a property of the underlying RSA function called *random self-reducibility*, and not from the proof itself. Random self-reducibility is the property of a class of problems that enables translating a given instance into another, random instance. The RSA inversion problem is most illustrative: given the instance x, its inverse can be found from r and the inverse of $r^e x$. As long as the solver chooses r at random, the instance $r^e x$ is uniformly random. It is what guarantees that the adversary cannot behave differently with respect to hash queries where he is being tricked into solving the RSA inversion problem. As this security proof relies on random self-reducibility, it does not apply to generic hash-and-sign constructions unless they feature random self-reducibility as well.

Generally speaking, post-quantum hash-and-sign signature schemes do not have random self-reducibility.

3.4 Quantum Random Oracle Model

To develop post-quantum cryptography, it is not sufficient to exchange pre-quantum hard problems for post-quantum hard problems and still employ the same construction strategy. The reason is that quantum attacks do not target only the hard problem; they target the security proof as well. This fact is particularly evident in the case of proofs in the random oracle model.

The random oracle is an indispensable tool for the construction and provable security of cryptographic functions and protocols [58, 20]. Informally, random oracles represent truly random functions, and therefore accurately capture the ideal situation in which the adversary knows nothing about the function's value for inputs that were not evaluated. In addition to EUF-CMA proofs like that of Example 3, random oracles are used in the all-or-nothing OAEP construction [21], the Fujisaki-Okamoto transform for generating CCA-secure cryptosystems from CPA ones [61], and transforms for obtaining signature schemes and non-interactive zero-knowledge proofs from interactive ones [58, 59].

However, this list of examples contains only classical, pre-quantum systems. Security proofs based on the random oracle model tend to break down in the context of adversaries capable of performing computations on quantum computers. This was first observed by Boneh *et al.* [28], who argue that however well a hash function approximates a random oracle, it must also be accessible to the quantum attacker and therefore it must *receive and answer superposition queries.* The classical random oracle model therefore fails to capture security against quantum adversaries. Instead, Boneh *et al.* recommend a *quantum random oracle model.* In this model, all parties are presented with query-access to an oracle that computes a random function $\mathsf{H} \xleftarrow{\$} \{f \mid f : \{0,1\}^* \rightarrow \{0,1\}^\star\}$ of arbitrary-length bit strings to arbitrary-length bit strings, selected at the start of the protocol. To query the oracle, the adversary sends it two registers (Q, R). The oracle then maps $|q, r\rangle \mapsto |q, r \oplus \mathsf{H}(q)\rangle$ before returning the registers to where they came from.

Unfortunately, several powerful proof strategies that work in the classical random oracle model break down in the quantum random oracle model.

- *Adaptive programmability.* Adaptive programmability refers to the capability of the simulator to change the input-output behavior of the random oracle while the protocol is running. Classically, the outputs of unqueried inputs may be considered undefined as they have not been selected yet. Quantumly, however, the adversary can query the superposition of all bitstrings and the result should be a superposition of all matching responses, thus fixing every input-output pair at once.

- *Query collection.* Classically, the simulator can keep a list of query and response pairs as they come. After the simulation is done and the simulator needs to know the matching preimage to a given hash, he can simply browse the list of queries. Quantumly, this strategy fails because of the No-Cloning Theorem: unless the query represents classical information, the adversary cannot copy any information from it without affecting its state. Consequently, an adversary that is not being simulated may behave differently from the same adversary when it is being simulated by a simulator that is trying to extract the queries.
- *Lazy sampling.* In the classical setting, outputs to yet-unqueried inputs may be sampled dynamically, *i.e.*, no sooner than when they are necessary. This enables the simulator to reflect on the previously received queries —from all simulation oracles— and select outputs that conform to a consistent adversarial view. The key point is that the correct value of the outputs may depend on previous query values. Quantumly, however, the entire list of input-output maps must be fixed at the onset of the simulation.
- *Rewinding.* Some proof techniques require collecting the output of an adversarial computation, then rewinding the adversary to some intermediate point, and then replaying it but relative to a different random oracle. Collecting the output and then rewinding presents a challenge in and of itself thanks to the No Cloning Theorem. More importantly, replaying the adversary relative to a different random oracle is in conflict with the requirement that the entire list of input-output pairs be fixed at the time of the first query.

On the up side, the same paper by Boneh *et al.* [28] presents a positive result for *history-free* reductions. The technical definition is rather cumbersome for its length and specificity, but informally it requires that the random oracle answers queries independently of its history. In other words, no part of the simulator's memory is allowed to change as a result of answering queries. From this point of view, maybe the term *memory-free* would have been a more descriptive choice of words. Random oracle proofs that are history-free, or memory-free, do hold in the quantum computing model.

3.4.1 Providing Oracle Access.

The first question to ask is how a polynomial-time simulator can provide access to a random function that potentially fixes an exponentially-large list of query-response pairs at the point of first query. Unruh answers this question as follows [140]. The simulator S, in addition to whatever other return value, outputs a description of the circuit for H. The simulated adversary A then has oracle access to H over the course of its computation. Since S runs in polynomial time, the description of H can be at most polynomial in size whereas random functions require an exponentially large description (probably). So H cannot have the same distribution as a truly random function. However, this is not a problem because A is only allowed to make a polynomial number, say $\hat{Q}_{\mathsf{H}}$ of queries. The question is not whether H is distinguishable from a random function, but whether H is distinguishable from a random function *given oracle access and at most $\hat{Q}_{\mathsf{H}}$ queries*. A complementary result by Zhandry shows that $2\hat{Q}_{\mathsf{H}}$-wise independent functions are perfectly indistinguishable from random functions from at most $\hat{Q}_{\mathsf{H}}$ quantum queries [152]. Since random polynomials of degree at most $2\hat{Q}_{\mathsf{H}} - 1$ are $2\hat{Q}_{\mathsf{H}}$-wise independent, the most straightforward strategy for S is to simply choose such a random polynomial of degree at most $2\hat{Q}_{\mathsf{H}} - 1$ and output a circuit for its evaluation as the description of H.

Personally, I find this approach inelegant. First, it requires that S knows $\hat{Q}_{\mathsf{H}}$ or at least an upper bound on this number. However, it is not clear that this upper bound can be computed by S if it only has black-box access to A, who might decide dynamically to make more queries. Second, a random polynomial over the field $\mathbb{F}_{2^\lambda}$ lends naturally to a random function $\{0,1\}^\lambda \rightarrow \{0,1\}^\lambda$, but careful construction is required if the desired function signature is instead $\{0,1\}^\lambda \rightarrow \{0,1\}^\kappa$ with $\kappa > \lambda$. Third, in many cases the random oracle must be programmed to give certain responses to certain queries. Finding a random bounded-degree polynomial subject to these constraints requires expensive interpolation and constitutes a needless simulation overhead.

Instead, I propose the following approach whereby the simulator S also has access to a random oracle H' — a different one but with the same function signature. Then S must produce an *interface* to H for A, which is a description of a circuit that computes H but relative to H', and that moreover takes into account the necessary programmed responses. Suppose for example that S has compiled a dictionary $\mathcal{D}$ of to-be-programmed query-response pairs. He can then provide A with the following interface to H.

```
define $\mathsf{H}(q)$ as:
    if $q \in \mathcal{D}.\mathsf{keys}()$ then:
        return $\mathcal{D}[q]$
    else:
        return $\mathsf{H}'(q)$
```

Formally, the simulator S is required to output this description of H to a separate tape before the simulation of the adversary A begins.

3.4.2 Aggregate Quantum Query Amplitude

In the classical random oracle model, it is often useful to consider the list of queries and their matching responses, and argue about the probability of particular queries or responses being members of this list. In the quantum random oracle, this list is ill-defined because a single query might contain a superposition of all possible queries, each with a negligible absolute amplitude. However, it turns out it is possible to salvage the spirit behind arguments involving the probability of particular queries being made at some point in an adversarial computation. This leads to the definition of the *aggregate quantum query amplitude* as a metric for the degree to which members of a list $\mathcal{S}$ of possible bit strings have been queried.

Definition 5 (aggregate quantum query amplitude [134])**.** *Let* A^{H} *be a quantum algorithm with oracle access to* H *making* $\hat{Q}$ *queries. In particular,* A *consists of* $\hat{Q}+1$ *unitary transforms* $U_0, \ldots, U_{\hat{Q}}$ *operating on a triple of quantum registers* S, Q, R*, and interleaved with unitary circuits* H *operating only on* Q, R *and sending* $|q, r\rangle \mapsto |q, r \oplus \mathsf{H}(q)\rangle$*. Let* ρ_k^Q *represent the reduced density matrix with respect to* Q *immediately after query* k*, with query indexation starting at zero. Then the* aggregate quantum query amplitude $\hat{a}_{\mathcal{S}}$ *associated with a set* $\mathcal{S}$ *of potential queries is*

$$\hat{a}_{\mathcal{S}} \triangleq \sum_{k=0}^{\hat{Q}-1} \sqrt{\sum_{s \in \mathcal{S}} \langle s | \rho_k^Q | s \rangle} \ . \tag{3.16}$$

In the same paper where Reza Reyhanitabar, Bart Preneel, and I define the notion, we provide lemmata for easy usage. The first two bound the aggregate quantum query amplitude for larger, respectively smaller, sets. The third shows that the aggregate quantum query amplitude is an upper bound on the trace distance (and hence maximum distinguishing probability) of the final state of an oracle algorithm with respect to an oracle whose outputs differ only in a set $\mathcal{S}$. Proofs can be found in the original paper [134].

Lemma 1. *For any two sets* $\mathcal{S}_1, \mathcal{S}_2 \subseteq \{0,1\}^*$, $\hat{a}_{\mathcal{S}_1} \leq \hat{a}_{\mathcal{S}_1 \cup \mathcal{S}_2}$.

Lemma 2. *For any two sets* $\mathcal{S}_1, \mathcal{S}_2 \subset \{0,1\}^*$, *if* $\hat{a}_{\mathcal{S}_1} \leq 1$ *and* $\hat{a}_{\mathcal{S}_2} \leq 1$ *then* $\hat{a}_{\mathcal{S}_1 \cup \mathcal{S}_2} \leq \hat{a}_{\mathcal{S}_1} + \hat{a}_{\mathcal{S}_2}$.

Lemma 3. *Let* D *be a quantum distinguisher making at most* $\hat{Q}$ *queries to one of two oracles* $\mathsf{H}_0, \mathsf{H}_1$, *whose outputs differ only on a set* $\mathcal{S}$ *of inputs. Then the trace distance of the distinguishers' final states is bounded by*

$$\mathrm{TD}(\mathsf{D}^{\mathsf{H}_1}(), \mathsf{D}^{\mathsf{H}_2}()) \leq 2\hat{a}_{\mathcal{S}} \ . \tag{3.17}$$

3.4.3 One-way to Hiding Lemma

Random oracle proofs often rely on the adversary's ignorance of responses to queries that were not made. It turns out that this intuition can be lifted to the quantum random oracle model. Unruh's One-Way to Hiding Lemma [139] formalizes the argument by introducing an extractor machine who waits until a randomly chosen query-and-response interaction and measures the query register at that point. An adversary that *does* know the given response can only learn it from making a query, and so it gives rise to a successful query extractor. The following lemma, which explicitly relates to the aggregate quantum query amplitude, states the lemma in terms of an oracle algorithm trying to determine which of two almost-identical oracles it has access to. This is in contrast to Unruh's formulation, which states the lemma in terms of an algorithm tasked with determining whether its input (x, y) is consistent with respect to the single oracle H, *i.e.*, whether $\mathsf{H}(x) \stackrel{?}{=} y$. Nevertheless, the first step in Unruh's proof is to translate the lemma into a distinguishing task with respect to two almost-identical oracles.

Lemma 4 (multi-target one-way to hiding [134])**.** *Let* H_0 *and* H_1 *be identical oracle functions except when their input belongs to a set* $\mathcal{S}$, *and let* A *be a quantum adversary that makes at most* $\hat{Q}_{\mathsf{H}}$ *queries to either* H_0 *or* H_1. *Let* E *be the following algorithm: select* $b \xleftarrow{\$} \{0,1\}$ *and* $k \xleftarrow{\$} \{0, \ldots, \hat{Q}_{\mathsf{H}} - 1\}$ *at random, simulate* $\mathsf{A}^{\mathsf{H}_b}$ *until the kth query, measure the query register in the* *computational basis, and output the result. Then*

$$\left(\frac{1}{2\hat{Q}_{\mathsf{H}}}\mathrm{TD}(\mathsf{A}^{\mathsf{H}_0}(), \mathsf{A}^{\mathsf{H}_1}())\right)^2 \leq \left(\frac{\hat{a}_{\mathcal{S}}}{\hat{Q}_{\mathsf{H}}}\right)^2 \leq \Pr[\mathsf{E}^{\mathsf{A},\mathsf{H}_0,\mathsf{H}_1}() = s \in \mathcal{S}] \ . \tag{3.18}$$

The left inequality is simply a restatement of Lemma 3. The inequality on the right follows from a straightforward description of the probability that E outputs some $s \in \mathcal{S}$ in terms of the reduced density operator ρ_k^Q of the state of A with respect to the query register Q at query k. Namely:

$$\Pr[\mathsf{E}^{\mathsf{A},\mathsf{H}_0,\mathsf{H}_1}() = s \in \mathcal{S}] = \sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \sum_{s\in\mathcal{S}} \Pr[\mathsf{E}^{\mathsf{A},\mathsf{H}_0,\mathsf{H}_1}() = s \wedge \mathsf{E}\, \textit{chooses}\, k] \tag{3.19}$$

$$= \sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \sum_{s\in\mathcal{S}} \langle s|\rho_k^Q|s\rangle \cdot \frac{1}{\hat{Q}_{\mathsf{H}}} \ , \tag{3.20}$$

which in conjunction with the following application of Jensen's inequality gives the lemma statement:

$$\hat{a}_{\mathcal{S}} = \sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \sqrt{\sum_{s\in\mathcal{S}} \langle s|\rho_k^Q|s\rangle} = \hat{Q}_{\mathsf{H}} \sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \frac{1}{\hat{Q}_{\mathsf{H}}} \sqrt{\sum_{s\in\mathcal{S}} \langle s|\rho_k^Q|s\rangle} \tag{3.21}$$

$$\leq \hat{Q}_{\mathsf{H}} \sqrt{\sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \frac{1}{\hat{Q}_{\mathsf{H}}} \sum_{s\in\mathcal{S}} \langle s|\rho_k^Q|s\rangle} \ . \tag{3.22}$$

3.4.4 Preimage Search

One of the best-used properties of random oracles is the difficulty of finding preimages satisfying certain criteria. Three games in particular capture this intuition. Informally:

- *One-Wayness.* The adversary is given a list of targets $\mathcal{Y} = \{y_1, \ldots, y_p\}$ and his task is to find a preimage x such that $\mathsf{H}(x) \in \mathcal{Y}$. One-wayness is used *e.g.* to achieve the *hiding* of information if an adversary, capable of learning the information despite its being hidden, can be made to break one-wayness.

- *Second Preimage Resistance.* The adversary is given a list of first preimages $\mathcal{X} = \{x_1, \ldots, x_p\}$ and his task is to find another preimage x such that $x \notin \mathcal{X}$ but for some $i \in \{1, \ldots, p\}$, $\mathsf{H}(x) = \mathsf{H}(x_i)$. This captures the security requirement for, *e.g.*, Merkle trees: an adversary who produces a new authentication path has found a second preimage for some node.
- *Marked Element Search.* The adversary is given access to a marking function $\mathsf{mark} : \mathsf{Domain}(\mathsf{H}) \times \mathsf{Range}(\mathsf{H}) \rightarrow \{0, 1\}$ that determines if a given input-output pair is valid, and his task is to find an input x such that $\mathsf{mark}(x, \mathsf{H}(x)) = 1$. This captures the security of, *e.g.*, the Fiat-Shamir transform for making interactive proofs non-interactive. For a given commitment, a fraudulent adversary can answer only a small fraction of challenges. When the challenge is determined as the hash of the commitment, then the adversary must find a commitment that leads to a challenge he can answer.

In fact, the three games can be used to define the insecurity functions of a hash function family $\mathcal{H} = \{\mathsf{H}_0, \mathsf{H}_1, \ldots, \mathsf{H}_{k-1}\} \subset \{f \mid f : \{0,1\}^* \rightarrow \{0,1\}^\star\}$. When $\mathcal{H} = \{f \mid f : \{0,1\}^* \rightarrow \{0,1\}^\star\}$ then the same insecurity applies to "the" random oracle. Formal descriptions of the games are shown in Games 3.5, 3.6, and 3.7. The acronyms abbreviate *single-function, multi-target one-wayness* (SM-OW), *single-function, multi-target second preimage resistance* (SM-SPR), and *marked element search* (MES).

Game 3.5: SM-OW

```
$\mathsf{H} \xleftarrow{\$} \mathcal{H}$
for $i$ from 1 to $p$ do:
    $x_i \xleftarrow{\$} \mathsf{Range}(\mathsf{H})$
    $y_i \leftarrow \mathsf{H}(x_i)$
$x' \leftarrow \mathsf{A}^{\mathsf{H}}(y_1, \ldots, y_p)$
return $[\![\exists i \,.\, \mathsf{H}(x') = y_i]\!]$
```

Game 3.6: SM-SPR

```
$\mathsf{H} \xleftarrow{\$} \mathcal{H}$
for $i$ from 1 to $p$ do:
    $x_i \xleftarrow{\$} \mathsf{Range}(\mathsf{H})$
$x' \leftarrow \mathsf{A}^{\mathsf{H}}(x_1, \ldots, x_p)$
return $[\![\exists i \,.\, \mathsf{H}(x') = \mathsf{H}(x_i) \wedge x' \neq x_i]\!]$
```

Game 3.7: MES

```
$\mathsf{H} \xleftarrow{\$} \mathcal{H}$
$x' \leftarrow \mathsf{A}^{\mathsf{H},\mathsf{mark}}()$
return $\mathsf{mark}(x', \mathsf{H}(x))$
```

Definition 6 (one-wayness insecurity). *The one-wayness insecurity of a hash function family $\mathcal{H}$ is defined as the maximum success probability in the SM-OW game (Game 3.5) with p targets across all unbounded adversaries* A *making at most $\hat{Q}$ queries:*

$$\mathsf{InSec}^{\mathsf{SM\text{-}OW}}_{\mathcal{H}}(\hat{Q}, p) \stackrel{\triangle}{=} \mathsf{max}_{\mathsf{A}} \Pr[\mathsf{SM\text{-}OW}^{\mathsf{A}}() = 1] \ . \tag{3.23}$$

Definition 7 (second preimage resistance insecurity). *The second preimage resistance insecurity of a hash function family $\mathcal{H}$ is defined as the maximum success probability in the SM-SPR (Game 3.6) game with p targets across all unbounded adversaries* A *making at most $\hat{Q}$ queries:*

$$\mathsf{InSec}^{\mathsf{SM\text{-}SPR}}_{\mathcal{H}}(\hat{Q}, p) \stackrel{\triangle}{=} \mathsf{max}_{\mathsf{A}} \Pr[\mathsf{SM\text{-}SPR}^{\mathsf{A}}() = 1] \ . \tag{3.24}$$

Definition 8 (marked element search insecurity). *The marked element search insecurity of a hash function family $\mathcal{H}$ and a marking function* mark *is defined as the maximum success probability in the MES game (Game 3.7) across all unbounded adversaries* A *making at most $\hat{Q}$ queries:*

$$\mathsf{InSec}^{\mathsf{MES}}_{\mathcal{H},\mathsf{mark}}(\hat{Q}) \stackrel{\triangle}{=} \mathsf{max}_{\mathsf{A}} \Pr[\mathsf{MES}^{\mathsf{A},\mathsf{mark}}() = 1] \ . \tag{3.25}$$

The first to show an upper bound on the insecurity of preimage search in the quantum-accessible oracle model was Unruh [140]. Instead of counting the number of targets or of marked elements, this result is articulated in terms of the *ratio* of the number of targets to elements in the output space, or of marked elements. The following paraphrases and re-proves Unruh's result, starting from the aggregate quantum query amplitude and casts it into the language of insecurity functions.

Definition 9 (Bernoulli function search). *Let $\mathcal{B}_\gamma$ be a Bernoulli distribution of functions* $\mathsf{B} : \{0,1\}^* \to \{0,1\}$ *such that every* $\mathsf{B}(x)$ *is independently distributed with* $\Pr_{\mathsf{B},x}[\mathsf{B}(x) = 1] = \gamma$*. The Bernoulli function search (BFS) insecurity is the maximum probability of finding an x such that* $\mathsf{B}(x) = 1$ *across all unbounded adversaries with at most $\hat{Q}$ quantum queries:*

$$\mathsf{InSec}^{\mathsf{BFS}}_{\gamma}(\hat{Q}) \stackrel{\triangle}{=} \max_{\mathsf{A}} \Pr[\mathsf{B}(\mathsf{A}^{\mathsf{B}}()) = 1] \ . \tag{3.26}$$

Lemma 5 (insecurity of Bernoulli function search).

$$\mathsf{InSec}^{\mathsf{BFS}}_{\gamma}(\hat{Q}) \leq 2(\hat{Q} + 1)\sqrt{\gamma} \ . \tag{3.27}$$

Proof. Let $\mathsf{N} : \{0,1\}^* \to \{0,1\}, x \mapsto 0$ be the constant zero function. We assume the existence of a BFS adversary A making $\hat{Q}$ queries and use it to build a distinguisher D between the oracles B and N. Let $\mathcal{S} = \{x \,|\, \mathsf{B}(x) = 1\}$. The distinguisher D simulates A, obtains the candidate preimage x, and queries his oracle on this value and returns the result. The success probability when the oracle is N is

$$\Pr_{\mathsf{B}\sim\mathcal{B}}[\mathsf{D}^{\mathsf{A},\mathsf{N}}() = 0] = 1 \ , \tag{3.28}$$

because no possible return value x from A can make $\mathsf{N}(x) = 1$. When the oracle is B then the success probability is

$$\Pr_{\mathsf{B}\sim\mathcal{B}}[\mathsf{D}^{\mathsf{A},\mathsf{B}}() = 1] = \mathsf{InSec}_{\gamma}^{\mathsf{BFS}}(\hat{Q}) \ , \tag{3.29}$$

because that is the probability that A returns a value $x \in \mathcal{S}$, whose membership in $\mathcal{S}$ guarantees that $\mathsf{B}(x) = 1$. So in summary,

$$|\Pr_{\mathsf{B}\sim\mathcal{B}}[\mathsf{D}^{\mathsf{A},\mathsf{B}}() = 1] - \Pr_{\mathsf{B}\sim\mathcal{B}}[\mathsf{D}^{\mathsf{A},\mathsf{N}}() = 1]| = \mathsf{InSec}_{\gamma}^{\mathsf{BFS}}(\hat{Q}) \ , \tag{3.30}$$

where the probabilities are taken both over the randomness involved in the selection of B, and over the random tape of D (and hence A). In fact, it pays to separate the two sources of randomness. Since the selection of B is independent from the random tapes, this gives:

$$\mathsf{InSec}_{\gamma}^{\mathsf{BFS}}(\hat{Q}) = \sum_{\mathsf{B}} \Pr[\mathsf{B}] \cdot |\Pr[\mathsf{D}^{\mathsf{A},\mathsf{B}}() = 1] - \Pr[\mathsf{D}^{\mathsf{A},\mathsf{N}}() = 1]| \ . \tag{3.31}$$

Each term in the right hand side of Eqn. 3.31 is in turn bounded by the trace distance of D's final state across both worlds. This enables a bound on this quantity via Lemma 3:

$$\sum_{\mathsf{B}} \Pr[\mathsf{B}] \cdot |\Pr[\mathsf{D}^{\mathsf{A},\mathsf{B}}() = 1] - \Pr[\mathsf{D}^{\mathsf{A},\mathsf{N}}() = 1]| \leq \sum_{\mathsf{B}} \Pr[\mathsf{B}] \cdot \mathrm{TD}(\mathsf{D}^{\mathsf{A},\mathsf{B}}(), \mathsf{D}^{\mathsf{A},\mathsf{N}}()) \tag{3.32}$$

$$\leq \sum_{\mathsf{B}} \Pr[\mathsf{B}] \cdot 2\hat{a}_{\mathcal{S}} \tag{3.33}$$

$$= \sum_{\mathsf{B}} \Pr[\mathsf{B}] \cdot 2 \sum_{k=0}^{\hat{Q}} \sqrt{\sum_{s\in\mathcal{S}} \langle s|\rho_k^Q|s\rangle} \ . \tag{3.34}$$

The middle sum runs from $k = 0$ to $k = \hat{Q}$ because D makes one query more than A. Remember that $\mathcal{S} = \mathcal{S}(\mathsf{B}) = \mathcal{S}_\mathsf{B}$ is determined by B. Continue with switching the summation order:

$$= 2\sum_{k=0}^{\hat{Q}} \sum_{\mathsf{B}} \Pr[\mathsf{B}] \cdot \sqrt{\sum_{s \in \mathcal{S}_\mathsf{B}} \langle s|\rho_k^Q|s\rangle} \tag{3.35}$$

$$\leq 2\sum_{k=0}^{\hat{Q}} \sqrt{\sum_{\mathsf{B}} \Pr[\mathsf{B}] \cdot \sum_{s \in \mathcal{S}_\mathsf{B}} \langle s|\rho_k^Q|s\rangle} \tag{3.36}$$

$$= 2\sum_{k=0}^{\hat{Q}} \sqrt{\sum_{\mathsf{B}} \Pr[\mathsf{B}] \cdot \sum_{x \in \{0,1\}^{l_Q}} \mathsf{B}(x) \langle x|\rho_k^Q|x\rangle} \tag{3.37}$$

$$= 2\sum_{k=0}^{\hat{Q}} \sqrt{\sum_{x \in \{0,1\}^{l_Q}} \langle x|\rho_k^Q|x\rangle \sum_{\mathsf{B}} \Pr[\mathsf{B}] \cdot \mathsf{B}(x)} \tag{3.38}$$

$$= 2\sum_{k=0}^{\hat{Q}} \sqrt{\sum_{x \in \{0,1\}^{l_Q}} \langle x|\rho_k^Q|x\rangle \cdot \mathrm{E}[\mathsf{B}(x)]} \tag{3.39}$$

$$= 2\sum_{k=0}^{\hat{Q}} \sqrt{\sum_{x \in \{0,1\}^{l_Q}} \langle x|\rho_k^Q|x\rangle \cdot \gamma} \tag{3.40}$$

$$= 2\sum_{k=0}^{\hat{Q}} \sqrt{\gamma \sum_{x \in \{0,1\}^{l_Q}} \langle x|\rho_k^Q|x\rangle} \tag{3.41}$$

$$= 2\sum_{k=0}^{\hat{Q}} \sqrt{\gamma} \tag{3.42}$$

$$= 2(\hat{Q} + 1)\sqrt{\gamma} \ . \tag{3.43}$$

The inequality is an application of Jensen's inequality. Here l_Q is the number of qubits in the query register Q, and the squared-amplitudes associated with all possible l_Q-bit basis vectors sum to one because of the law of total probability. □

If $\mathcal{H} = \{f \,|\, f : \{0,1\}^* \to \{0,1\}^\star\}$ is the random oracle function family, then the function $\mathsf{B} : \{0,1\}^* \to \{0,1\}$,

$$x' \mapsto \begin{cases} [\![\exists i \,.\, \mathsf{H}(x') = y_i]\!] & \text{in the case of SM-OW,} \\ [\![\exists i \,.\, \mathsf{H}(x') = \mathsf{H}(x) \,\wedge\, x' \neq x_i]\!] & \text{in the case of SM-SPR,} \\ \mathsf{mark}(x', \mathsf{H}(x)) & \text{in the case of MES,} \end{cases}$$

is very close to the distribution $\mathcal{B}$ of Lemma 5. To see this, observe that in every case its value depends on $\mathsf{H}(x')$, which is sampled at random with H. The ratio of the number of $\mathsf{H}(x')$ that satisfy the predicate to $\#\mathsf{Range}(\mathsf{H})$ is exactly γ. Therefore, an adversary solving SM-OW, SM-SPR, or MES is *simultaneously* solving BFS. We can therefore use the upper bound on BFS insecurity to upper bound the adversary's success probability of its original game:

$$\mathsf{InSec}^{\mathsf{SM\text{-}OW}}_{\mathcal{H}}(\hat{Q}, p) \leq \mathsf{InSec}^{\mathsf{BFS}}_{p/\#\mathsf{Range}(\mathsf{H})}(\hat{Q}) \tag{3.44}$$

$$\mathsf{InSec}^{\mathsf{SM\text{-}SPR}}_{\mathcal{H}}(\hat{Q}, p) \leq \mathsf{InSec}^{\mathsf{BFS}}_{p/\#\mathsf{Range}(\mathsf{H})}(\hat{Q}) \tag{3.45}$$

$$\mathsf{InSec}^{\mathsf{MES}}_{\mathcal{H},\mathsf{mark}}(\hat{Q}) \leq \mathsf{InSec}^{\mathsf{BFS}}_{p/\#\mathsf{Range}(\mathsf{H})}(\hat{Q}) \ . \tag{3.46}$$

In the last equation, and more generally in the context of MES, $p = \mathsf{max}_x \, \#\{y \,|\, \mathsf{mark}(x, y) = 1\}$ is the maximum number of outputs that could make a given input into a marked element.

Note that this reduction is different in terms of form from standard reductions in provable security. There we expect to transform one adversary that solves, *e.g.*, SM-OW (analogous arguments hold for SM-SPR and MES), into another adversary that solves BFS *for a given function* B. Presently, B is not a given argument to the reduction, but determined not just by H but also by the chosen values for x_i. We are saved, however, by working in the random oracle model. The argument to the reduction is not a concrete sample B but a distribution $\mathcal{B}$. While the concrete samples H and $\{x_i\}_i$ determine a concrete sample B, the distributions they are drawn from induce a distribution $\mathcal{B}'$ and the reduction holds in the random oracle model if $\mathcal{B}' = \mathcal{B}$.

Another important note is that the induced distribution $\mathcal{B}'$ is not a Bernoulli function distribution as per Definition 9. It is true that for a given sample B, $\Pr_x[\mathsf{B}(x)] = \gamma$. However, for a given B, the values $\mathsf{B}(x)$ are not independent for different x because the number $\#\mathcal{S} = \#\{x \,|\, \mathsf{B}(x) = 1\}$ is bounded away from zero — in fact, if no collisions occur, this number is exactly p. In contrast, there is a nonzero probability of sampling the constant zero function $\mathsf{N} : x \mapsto 0$ from a true Bernoulli function distribution $\mathcal{B}_\gamma$.

However, the inequalities 3.44—3.46 *do* hold for conditional probability distributions where $\#\mathcal{S}$ is fixed or where this number follows a given distribution. Moreover, the proof of Lemma 5 can be made to work even for such a conditional probability distribution. The crucial transition is between Eqns. 3.39 and 3.40, which requires that $\mathrm{E}_{\mathsf{B}}[\mathsf{B}(x)] = \gamma$ for all $x \in \{0,1\}^{l_Q}$. This equality clearly holds for a Bernoulli function distribution $\mathcal{B}_\gamma$ because

$$
\begin{aligned}
\mathrm{E}_{\mathsf{B},x}\left[\mathsf{B}(x)\right] &= \mathrm{E}_{\mathsf{B}}\left[\sum_x \Pr_x[x]\mathsf{B}(x)\right] && (3.47)\\
&= \mathrm{E}_{\mathsf{B}}\left[\#\{x \,|\, \mathsf{B}(x) = 1\}/2^{l_Q}\right] && (3.48)\\
&= \mathrm{E}_{\mathsf{B}}\left[\Pr_x\left[\mathsf{B}(x) = 1\right]\right] && (3.49)\\
&= \mathrm{E}_{\mathsf{B}}[\gamma] = \gamma \ , && (3.50)
\end{aligned}
$$

and since this value is independent of x, $\mathrm{E}_{\mathsf{B},x}[\mathsf{B}(x)] = \mathrm{E}_{\mathsf{B}}[\mathsf{B}(x)]$.

However, if $\mathrm{E}_{\mathsf{B}}[\mathsf{B}(x)] \neq \gamma$ but still independent of x, then the same derivation of Lemma 5 holds provided that one replaces γ with $\mathrm{E}_{\mathsf{B}}[\mathsf{B}(x)]$ or whichever symbol is used to denote this value. This motivates the following symbol abuse.

Definition 10 (BFS, amended)**.** *Let $\mathcal{B}_\gamma$ be a distribution of functions $\mathsf{B} : \{0,1\}^* \to \{0,1\}$ where for all $x \in \{0,1\}^*$ we have $\mathrm{E}[\mathsf{B}(x)] = \gamma$. The BFS insecurity for $\mathcal{B}_\gamma$ is defined as the maximum probability of finding an x such that $\mathsf{B}(x) = 1$ across all unbounded adversaries given at most $\hat{Q}$ quantum queries to $\mathsf{B} \sim \mathcal{B}_\gamma$:*

$$\mathsf{InSec}^{\mathsf{BFS}}_{\gamma}(\hat{Q}) \overset{\triangle}{=} \max_{\mathsf{A}} \Pr[\mathsf{B}(\mathsf{A}^{\mathsf{B}}()) = 1] \ . \qquad (3.51)$$

With this redefinition of the symbol $\mathsf{InSec}^{\mathsf{BFS}}_{\gamma}(\hat{Q})$, the bounds of Lemma 5 and of Eqns. 3.44, 3.45 and 3.46 hold without reservation. One may imagine the "B" to stand for "Boolean" to stress the distinction between Defs. 9 and 10, or once again for "Bernoulli" to hide it.

Hülsing *et al.* have a somewhat stronger result [70]. They start[2] with the original BFS distribution $\mathcal{B}_\gamma$ of Def. 9, apply a theorem by Zhandry [151, Thm. 7.2] to it, and obtain

[2] In fact, the description of the distribution D_λ on [70, page 9] is technically speaking distinct from $\mathcal{B}_\gamma$ of Def. 9, but the authors have confirmed in private communication that it was meant to be identical. Indeed, the proof of Thm. 2 only works for $\mathcal{B}_\gamma$.

Theorem 2 ([70])**.** *For all adversaries* A *with at most* $\hat{Q}$ *quantum queries to a oracle mapping* $\{0,1\}^n \to \{0,1\}$*, and for the family of distributions* $\mathcal{B}_\gamma$ *of Def. 9,*

$$\Pr_{\mathsf{B}\sim\mathcal{B}_\gamma}[\mathsf{B}(\mathsf{A}^{\mathsf{B}}()) = 1] \leq 8\gamma(\hat{Q}+1)^2 \ . \tag{3.52}$$

Of course, by restricting to this family of distributions, Hülsing *et al.* run into the problem identified earlier that the induced distribution is different from the one defined. They circumvent this obstacle by restricting attention to the random oracle function family $\mathcal{H} = \{f \,|\, f : \{0,1\}^m \to \{0,1\}^n\}$, and to the regime where $p \ll 2^n \ll 2^m$. Under these assumptions, Hülsing *et al.* provide reductions showing that,

$$\mathsf{InSec}^{\mathsf{SM\text{-}OW}}_{\mathcal{H}}(\hat{Q},p), \mathsf{InSec}^{\mathsf{SM\text{-}SPR}}_{\mathcal{H}}(\hat{Q},p) \in \Theta((\hat{Q}+1)p/2^n) \ , \tag{3.53}$$

where the Landau notation hides quantities that are negligible in the regime $p \ll 2^n \ll 2^m$. Beullens, Preneel, and I show that for the same random oracle function family and for any marking function that marks at most p outputs for a given input, *i.e.*, $p = \mathsf{max}_x \#\{y \,|\, \mathsf{mark}(x,y) = 1\}$, $\mathsf{InSec}^{\mathsf{MES}}_{\mathcal{H},\mathsf{mark}}(\hat{Q}) \leq \mathsf{InSec}^{\mathsf{SM\text{-}OW}}_{\mathcal{H}}(\hat{Q},p)$ [26]. And so $\mathsf{InSec}^{\mathsf{MES}}_{\mathcal{H},\mathsf{mark}}(\hat{Q}) \in O((\hat{Q}+1)p/2^n)$ as well.

The bounds of Unruh and of Hülsing *et al.* cannot both be true, can they? Strictly speaking, no contradiction is implied. Both bounds are compatible with the intuition derived from Grover's algorithm that the success probability is large only for a number of iterations that is on the order of $1/\sqrt{\gamma}$. From a closer inspection of Grover's algorithm, one would expect the success probability to rise with $\hat{Q}^2$ and in multiples of γ, *but only for small values for both.* The Hülsing *et al.* bound applies only if γ is sufficiently small, and moreover only if $2^m \gg 2^n$. Inside this regime, Hülsing *et al.*'s bounds are preferable. Outside of this regime, those of Unruh are. It remains an interesting open problem to determine the degree to which other regimes Hülsing *et al.*'s bounds can be lifted.

3.4.5 Preimage-awareness.

Another important technique enables the simulator to know the preimage of a given image. He can then proceed, for instance, to invert a commitment function and compute the witness in a zero-knowledge proof of knowledge, or answer decryption queries despite being ignorant of the secret key in an IND-CCA game. In the classical world, the simulator needs only look at the list of queries made by the adversary, and search for the query that yields the given response. In the quantum world this list cannot exist. However, the simulator can present the adversary with a trapdoored random oracle that is indistinguishable from an authentic one but for which the simulator can efficiently compute the list of candidate preimages. In fact, the possibility of achieving preimage-awareness vindicates to some degree the random-polynomial approach (as opposed to the interface-approach).

In particular, a random polynomial $p \in \mathbb{F}_{2^\kappa}[x]$ of degree at most $2\hat{Q} - 1$ is $2\hat{Q}$-wise independent and thus perfectly indistinguishable from a uniformly random function from $\{f \,|\, f : \{0,1\}^\kappa \rightarrow \{0,1\}^\kappa\}$ by any adversary that is restricted to at most $\hat{Q}$ queries. However, the simulator, who knows the coefficients, can efficiently factor the polynomial $p(x) - y$ and obtain a list of at most $2\hat{Q}$ candidate preimages to the image y. This approach was first used by Unruh [140].

The simulator does incur a simulation overhead as a result of this technique: he has to factor polynomials and test all elements in a list of candidates. A rigorous concrete proof must take this time cost into account. Asymptotically, both operations can be done in polynomial time. Moreover, both tasks can be formulated in a manner that is independent of the hard problems or cryptosystem underlying the construction; as a result, the extra time spent on answering queries is unlikely to hide extra time spent attacking the hard problem or cryptosystem. It may be argued then, that it is safe to ignore this time overhead.

A greater drawback of this technique is that it is restricted to length-preserving random oracles. Otherwise the list of candidate preimages explodes and can no longer be computed in polynomial time.

Chapter 4

Hard Problems

4.1 Multivariate Quadratic

Informally, the MQ problem asks to find a satisfying assignment to the variables in a list of multivariate quadratic polynomials over a finite field. The problem is known to be NP-hard in the worst case as well as empirically hard on average when the number of equations m is approximately equal to the number of variables n. It serves as the hard problem in a host of post-quantum cryptosystems [83, 109, 49, 51, 35]. Formally, the problem is stated as follows.

Hard Problem 4.1: MQ Problem

Parameters: number of equations m, number of variables n, field size q.

Given: a list $\mathbf{P} \in (\mathbb{F}_q[\mathbf{x}]_{\leq 2})^m$ of m polynomials of degree at most 2 in n variables $(x_1, \ldots, x_n) = \mathbf{x}^\mathsf{T}$ over a finite field $\mathbb{F}_q$.

Task: Find a solution $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathbf{P}(\mathbf{x}) = \mathbf{0}$.

The matching hardness assumption is essentially one-wayness of evaluation of random MQ systems, where "random" means selecting every coefficient uniformly at random from $\mathbb{F}_q$. Formally, the *MQ Assumption* states that if $m = n$, for all quantum polynomial-time adversaries S, the success probability is negligible, *i.e.*, $\mathsf{Succ}^{\mathsf{OW}}_{\mathcal{MQ}_{m,n}}(\mathsf{S}) \leq \mathsf{negl}(n)$, where

$$\mathsf{Succ}^{\mathsf{OW}}_{\mathcal{MQ}_{m,n}}(\mathsf{S}) \triangleq \Pr[\mathcal{P}(\mathbf{x}_1) = \mathbf{0} \,|\, \mathcal{P} \xleftarrow{\$} (\mathbb{F}_q[\mathbf{x}]_{\leq 2})^m \,;\, \mathbf{x}_0 \xleftarrow{\$} \mathbb{F}_q^n \,; \\ \mathbf{x}_1 \leftarrow \mathsf{S}(\mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{x}_0))] \ . \quad (4.1)$$

In this expression $\mathsf{S}(\mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{x}_0))$ represents the output of $\mathcal{S}$ when given a complete description of the list of polynomials $\mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{x}_0)$ as input.

4.1.1 Algebraic Attack

The best attack against generic instances of the MQ problem consists of a mixture of guessing variables and computing Gröbner bases of the ideals spanned by the resulting lists of polynomials. To see why a Gröbner basis might be useful for solving the problem, recall the following definitions.

Definition 11 (polynomial ideal)**.** *A* polynomial ideal $\mathcal{I}$ *is the algebraic span of a list of polynomials* $p_1(\mathbf{x}), \ldots, p_m(\mathbf{x}) \in \mathbb{F}_q[\mathbf{x}]$*:*

$$q(\mathbf{x}) \in \mathcal{I} \quad \Leftrightarrow \quad \exists \alpha_1(\mathbf{x}), \ldots, \alpha_m(\mathbf{x}) \in \mathbb{F}_q[\mathbf{x}] \,.\, q(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i(\mathbf{x}) p_i(\mathbf{x}) \ . \quad (4.2)$$

For convenience we write $\mathcal{I} = \langle p_1, \ldots, p_m \rangle$*.*

Definition 12 (monomial ordering; leading term, monomial, and coefficient)**.** *A* monomial ordering *is a relation* $\succ$ *on the all monomials of* $\mathbb{F}_q[\mathbf{x}]$ *satisfying:*

(i) totality*: for every pair of monomials* $l, r \in \mathbb{F}_q[\mathbf{x}]$ *either* $l \succ r$, $l = r$, *or* $l \prec r$*;*

(ii) if $l \succ r$ *then for any monomial* $m \in \mathbb{F}_q[\mathbf{x}]$, $ml \succ mr$*; and*

(iii) well-ordering*: every non-empty subset of monomials of* $\mathbb{F}_q[\mathbf{x}]$ *has a smallest element under* $\succ$*.*

A monomial order determines the largest term of a polynomial $p(\mathbf{x})$*, we write this* leading term $\boldsymbol{lt}(p(\mathbf{x}))$*. Its coefficient is the* leading coefficient $\boldsymbol{lc}(p(\mathbf{x}))$ *and its monomial is the* leading monomial $\boldsymbol{lm}(p(\mathbf{x}))$*.*

Definition 13 (Gröbner basis). *A* Gröbner basis $\mathbf{G}$ for an ideal $\mathcal{I}$ with respect to a monomial ordering $\succ$ *is a list of polynomials* $g_1(\mathbf{x}), \ldots, g_k(\mathbf{x}) \in \mathbb{F}_q[\mathbf{x}]$ *such that* $\langle g_1(\mathbf{x}), \ldots, g_k(\mathbf{x}) \rangle = \mathcal{I}$ *and such that for every polynomial* $q(\mathbf{x}) \in \mathcal{I}$*, there is a polynomial* $g_i(\mathbf{x})$ *in the list* $\mathbf{G}$ *with a leading monomial that divides that of* $q(\mathbf{x})$*. Symbolically, this condition is*

$$\forall q(\mathbf{x}) \in \mathcal{I} \,.\, \exists i \in \{1, \ldots, k\} \,.\, \boldsymbol{lm}(g_i(\mathbf{x})) | \boldsymbol{lm}(q(\mathbf{x})) \ . \tag{4.3}$$

A Gröbner basis generalizes the echelon form of linear systems of equations. For linear equations in echelon form, any additional linear equation is either linearly independent from the previous ones, or else it can be reduced to zero by adding scalar multiples of each previous equation. Similarly, for Gröbner bases, any additional polynomial is either algebraically independent from the basis elements (and thus outside the ideal) or else it can be reduced to zero by adding scalar multiples of the basis elements. By identifying a polynomial $p(\mathbf{x})$ with the equation $p(\mathbf{x}) = 0$ and vice versa, one extends Gröbner bases to systems of polynomial equations. The analogy with echelon form motivates the following analogue of the reduced echelon form.

Definition 14 (reduced Gröbner basis). *A* reduced Gröbner basis $\mathbf{G}$ *is a Gröbner basis* $g_1(\mathbf{x}), \ldots, g_k(\mathbf{x}) \in \mathbb{F}_q[\mathbf{x}]$ *such that a) all* $\boldsymbol{lc}(g_i) = 1$*; and b) for all* $g_i(\mathbf{x})$ *and all terms* $t(\mathbf{x})$ *of* $g_i(\mathbf{x})$*,* $t(\mathbf{x}) \notin \langle \boldsymbol{lt}(\mathbf{G} \backslash \{g_i(\mathbf{x})\}) \rangle$*.*

And just as solutions can be read out from a list of linear equations put into reduced echelon form, so too can solutions be read out from a reduced Gröbner basis. This is particularly obvious in the case of a lexicographical monomial ordering where $x_i^a x_{i+1}^b \succ x_{i+1}^c$ for any $a, c, b \in \mathbb{N} \backslash \{0\}$. Then the reduced Gröbner basis is triangular: $\forall j \, \exists i \,.\, i \leq j \wedge g_1(\mathbf{x}), \ldots, g_i(\mathbf{x}) \in \mathbb{F}_q[x_1, \ldots, x_j]$. So a sequence of greatest common divisor calculations, univariate polynomial factorizations to find roots, and back-substitutions, generates a complete and consistent assignment to all the variables. In the case of generic monomial orderings, the trick is to incrementally refine the ordering and update the Gröbner basis accordingly. Every step makes the first iteration of this elimination and back-substitution procedure possible. For details the reader is referred to [43, Ch. 3].

In cryptographic applications, a nice representation of the set of all solutions, or a complete enumeration of all its members, is rarely important. Instead, an attacker wins if he finds *just one* solution. Even if he has to find a *specific* solution, where *specific* means something that is not easily expressible in terms of algebraic equations, the complexity of finding an arbitrary solution may be indicative of the complexity of finding the specific one.

With this in mind, it makes sense to restrict attention to ideals whose solution set is zero-dimensional. This restriction is without loss of generality: an attacker can always fix some variables' values randomly until the number of variables equals the number of equations. Unless the system exhibits non-trivial algebraic dependencies, this makes the system of equations determined.

Another technique that reduces the complexity of a Gröbner basis calculation is to adjoin the field equations $x_i^q - x_i = 0$ to the system of equations. The polynomial on the left hand side evaluates to 0 in every element of $\mathbb{F}_q$, so this adjoining does not destroy solutions. Conversely, this polynomial is nonzero in some extension field elements $z \in \mathbb{F}_{q^e}$, but the attacker is not looking for solutions of that form to begin with. The reason why this technique helps is that $x_i^q - x_i$ can be used to reduce the degree of polynomials in the computation, particularly if q is small. For large q, adjoining the field equations is akin to adding dead weight.

In terms of cryptanalytic applications, the state of the art of Gröbner basis-like algorithms is a position shared jointly by F_4/F_5, MXL$_3$, and PWXL. All three methods explicitly relate the problem at hand to linear algebra. This relation relies on two observations. First, any list of polynomials $\mathbf{P} = (p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})) \in (\mathbb{F}_q[\mathbf{x}]_{\leq d})^m$ can be identified with a so-called *Macaulay matrix* $M_{\mathbf{P}} \in \mathbb{F}_q^{m \times \binom{n+d+1}{n}}$ containing the polynomials' coefficients. In particular, each row of the Macaulay matrix corresponds to one polynomial, and each column of the Macaulay matrix corresponds to a monomial. For example, the following system of polynomial equations is identifiable with the Macaulay matrix below.

$$\left\{ \begin{array}{r} p_1(\mathbf{x}) = x_1x_2 + x_3^2 + x_1 + x_3 + 1 = 0 \\ p_2(\mathbf{x}) = x_1^2 + x_2x_3 + x_3^2 + 1 = 0 \\ p_3(\mathbf{x}) = x_2^2 + x_2x_3 + x_3^2 + z = 0 \end{array} \right\} \tag{4.4}$$

$$\updownarrow$$

$$\begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{cccccccccc} x_1^2 & x_1x_2 & x_1x_3 & x_2^2 & x_2x_3 & x_3^2 & x_1 & x_2 & x_3 & 1 \end{array} \\ \left(\begin{array}{cccccccccc} 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{array} \right) \end{array} \tag{4.5}$$

Second, any system $\mathbf{P}$ of polynomial equations can be extended by adjoining new equations obtained from multiplying old ones by monomials. While this may increase the maximum degree across all polynomials, and thus widen the Macaulay matrix, it will also generate new rows. A subsequent reduction to echelon form may bring the polynomial system into Gröbner basis, or if a single solution is desired, the echelon form may produce a linear triangular system that determines the assignments to the variables. In fact, the original Buchberger's algorithm [31] for computing a Gröbner basis can be seen to be a specialization of these principles: the S-polynomial $S(p_i, p_j) = \frac{\mathsf{lcm}(\boldsymbol{lm}(p_i), \boldsymbol{lm}(p_j))}{\boldsymbol{lt}(p_i)} \cdot p_i - \frac{\mathsf{lcm}(\boldsymbol{lm}(p_i), \boldsymbol{lm}(p_j))}{\boldsymbol{lt}(p_j)} \cdot p_j$ is exactly a linear combination of extensions of p_i and p_j and the remainder of this S-polynomial modulo the other polynomials is exactly what is computed by the echelon reduction. Nevertheless, Buchberger's algorithm is just one way to specialize these principles; the F_4/F_5, MXL$_3$ and PWXL algorithms share this linear algebra perspective but differ in important respects.

F_4/F_5. Where Buchberger's algorithm adjoins one extended polynomial at a time, Faugère's F_4 algorithm [55] adjoins batches of S-polynomials before reducing all of them simultaneously. An important factor affecting complexity is the strategy by which the batch of S-polynomials is chosen. A straightforward constraint is to restrict attention in every step to those S-polynomials of lowest degree, as this guarantees that the Macaulay matrix is never larger than it needs to be. This degree is referred to as the *step degree.* The F_5 algorithm [54] presents another optimization. First, the algorithm computes a sequence of Gröbner bases: one for (p_1), then for (p_1, p_2), and so on. Second, every polynomial in the computation is stored with a signature that details how it was obtained from the original list. This pair of modifications enables a very stringent criterion for selecting critical pairs from which to compute S-polynomials; in particular, this criterion guarantees that no time is spent on redundantly reducing an S-polynomial to zero provided the system is *regular* (see below). While both F_4 and F_5 can be implemented with sparse polynomials, the implementation in the Magma computer algebra system [29] uses dense linear algebra. Moreover, Faugère indicates that the sparsity is lost in the course of large computations [34, § 3].

MXL$_3$. The MXL$_3$ algorithm [100] works specifically for ideals with zero-dimensional varieties[1], or equivalently, for determined systems of polynomial equations. Where the F_4/F_5 algorithms are extremely selective in their choices of which polynomials to extend and adjoin, the XL family of algorithms [42, 148] employs a rather brute strategy. All polynomials are extended via multiplication by all monomials such that the resulting degrees are equal to the current *working degree.* At this point, the Macaulay matrix is brought into reduced row-echelon form. If there are univariate polynomials, they are factored and a root is selected and back-substituted. Otherwise, the working degree is incremented. Eventually, all variables receive an assignment. What makes the subfamily of Mutant-XL algorithms [46, 101, 9] special is the attention devoted to *mutants*, *i.e.*, algebraic combinations of starting polynomials resulting in an unexpected

[1] A *variety* is the set of all solutions to all polynomials in the ideal.

degree drop. Since their degree is lower than the working degree, they can be extended *again*, thus providing more material with which to reduce the next polynomial. The novelty of MXL_3 is that the univariate factorization is dropped. As a result, MXL_3 actually outputs a proper Gröbner basis rather than a single solution, although it is only guaranteed to work if the ideal in question defines a zero-dimensional variety.

PWXL. The WXL algorithm [103] drops Gaussian elimination altogether in favor of a sparse linear system solver — in particular, the Wiedemann algorithm [144] or a blockwise generalization thereof due to Coppersmith [39]. After extending the original polynomials to working degree d, some random rows of the Macaulay matrix are dropped so as to make it square. If this square matrix A is non-singular, then $2\binom{n+d+1}{n}$ matrix-vector and vector-vector products suffice to compute the matrix's minimal polynomial, after which a rearranging of the intermediate results and summing with appropriate weights yields the solution $\bar{\mathbf{x}} = (x_1^d, x_1^{d-1}x_2, \ldots, x_n)^\mathsf{T}$ to the linear system of equations $A\bar{\mathbf{x}} = \mathbf{b}$. If A is singular, then try another random selection of rows. If A remains consistently singular then increment the working degree d. The "P" in PWXL indicates that the matrix-vector products are computed in a parallel fashion. While PWXL has a lower complexity than F_4/F_5 and MXL_3 asymptotically speaking, in practice it occasionally terminates at a higher working degree. This makes for a larger running time in practice.

Clearly, the various algorithms for performing an algebraic attack are related — and so are their complexities. They all boil down to performing sparse linear algebra on an extended Macaulay matrix. The complexity is therefore determined by the size of this matrix and, by proxy, the degree to which the polynomials are extended. Determining this degree is therefore an important aspect of estimating the complexity of algebraically solving a system of polynomial equations.

Degree of Regularity. These paragraphs recycle text from my answer in response to a question on Stack Exchange [146].

There are a couple of definitions in the literature that each aim to capture an aspect of the degree d to which a system of polynomial equations must be extended before linear algebra on its Macaulay matrix will yield a solution. Some of them are confusingly referred to as *the* degree of regularity, despite denoting logically different notions. For random MQ systems, we are interested in the *index of regularity* or the *degree of semi-regularity.*

Index of Regularity. The index of regularity is defined using the Hilbert polynomial and sequence of an ideal $\mathcal{I}$ [43, Ch.9 Sect.3.]. Denote the set of polynomials in $\mathcal{I}$ of degree s or lower as $\mathcal{I}_{\leq s}$ and the same for $\mathbb{F}_q[\mathbf{x}]_{\leq s}$. The Hilbert function $HF_{\mathcal{I}} : \mathbb{N} \to \mathbb{N}$ of an ideal $\mathcal{I}$ is defined as $HF_{\mathcal{I}}(s) = \mathsf{dim}(\mathbb{F}_q[\mathbf{x}]_{\leq s}/\mathcal{I}_{\leq s})$ and it follows immediately that $HF_{\mathcal{I}}(s) = \mathsf{dim}(\mathbb{F}_q[\mathbf{x}]_{\leq s}) - \mathsf{dim}(\mathcal{I}_{\leq s})$. For sufficiently large s, the Hilbert function of $\mathcal{I}$ is identical to a polynomial $HP_{\mathcal{I}}(s) = \sum_{i=0}^{d} b_i \binom{s}{d-i}$ for some $b_i \in \mathbb{Z}$ and $b_0 \in \mathbb{N}\backslash\{0\}$, called the Hilbert polynomial. The *index of regularity* is the smallest s_0 such that for all $s \geq s_0$, $HF_{\mathcal{I}}(s) = HP_{\mathcal{I}}(s)$. This value is also called the *Hilbert regularity* [145].

Degree of Semi-Regularity. A sequence of polynomials $(p_1(\mathbf{x}), \ldots, p_m(\mathbf{x}))$ is *regular* if $g \cdot p_i \in \langle p_1, \ldots, p_{i-1}\rangle \implies g \in \langle p_1, \ldots, p_{i-1}\rangle$. Regular systems capture the worst case of polynomial systems in terms of their solving complexity. The *Hilbert series* of an ideal $\mathcal{I}$ is defined as the formal power series $HS_{\mathcal{I}}(z) = \sum_{s=0}^{\infty} HF_{\mathcal{I}}(s) z^s$. The Hilbert series of the ideal $\mathcal{I} = \langle p_1, \ldots, p_m\rangle$ spanned by a regular sequence of homogeneous polynomials $(p_1, \ldots, p_m)$ is given by $HS_{\mathcal{I}}(z) = \frac{\prod_{j=1}^{m}(1-z^{\mathsf{deg}(p_j)})}{(1-z)^n}$. It is known [16] that the degree of the highest-degree elements in a degree-reverse lexicographical Gröbner basis is bounded (up to a linear change of variables) by the Macaulay bound: $\sum_{i=1}^{m}(\mathsf{deg}(p_i)-1)+1$. This bound can be used to estimate the complexity of Gröbner basis algorithms for regular (*i.e.*, worst-case) systems. If $m = n$, the sequence is regular if and only if $HS_{\mathcal{I}}$ is a polynomial [1]. This means that for some bound s_0 and all $s \geq s_0$, $HF_{\mathcal{I}}(s) = 0$ and so $HP_{\mathcal{I}}(s) = 0$. In this case, $s_0 = \mathsf{deg}(HS_{\mathcal{I}}) + 1$ is exactly the index of regularity.

Unfortunately, regular systems do not exist when m is larger than n. In this case, one must assume the ideal defines a zero-dimensional variety, and given that this is the case one can adapt the definition of regular sequences as follows. A list of polynomials $(p_1, \ldots, p_m)$ is *d-regular* if for all $g \in \mathbb{F}_q[\mathbf{x}]$ with $\mathsf{deg}(g) < d - \mathsf{deg}(p_i)$, $g \cdot p_i \in \langle p_1, \ldots, p_{i-1}\rangle \implies g \in \langle p_1, \ldots, p_{i-1}\rangle$. The list $(p_1, \ldots, p_m)$ is *semi-regular* if and only if it is s_0-regular, where s_0 is the index of regularity [16]. For a semi-regular system the Hilbert series $HS_{\mathcal{I}}(z)$ will *not* be a polynomial but it can *always* be written as a formal power series (*i.e.* a polynomial with an unlimited number of terms). In this case s_0 is the degree of the first term in this formal power series whose coefficient is zero or negative. Treating random systems of quadratic polynomial equations as semi-regular seems to be empirically justified, but there is no proof that random systems are indeed semi-regular with high probability.

Complexity. The first step of an algebraic attack is to choose random assignments to variables until the resulting system has as many equations as free variables. However, when the number of variables is more than twice the number of equations, we can do even better. Thomae and Wolf show that an MQ system with n variables and m equations can be reduced to another MQ system but with $m - \lfloor \frac{n}{m} \rfloor + 1$ equations and as many variables [137]. Moreover, in the case of random MQ systems, guessing more than $n - m$ variables causes a drop in the degree of semi-regularity that compensates for the cost of having to retry if the guess was incorrect.

In the case of random systems, the expression for the degree of semi-regularity is preferable to alternative degree of regularity notions. To this end, assume that random lists of polynomials behave the same way that regular sequences of homogeneous polynomials do. Recall that the Hilbert Series of such a sequence of polynomials $p_1, \ldots, p_m \in \mathbb{F}_q[\mathbf{x}]$ is given by

$$HS(z) = \frac{\prod_{i=1}^{m}(1 - z^{\mathsf{deg}(p_i)})}{(1-z)^n} \ . \tag{4.6}$$

The degree of semi-regularity is the degree of the first term in this power series whose coefficient is zero or negative. When $q = 2$, the modified series

$$HS'(z) = \frac{(1+z)^n}{\prod_{i=1}^{m}(1 + z^{\mathsf{deg}(p_i)})} \tag{4.7}$$

must be used instead [17]. Note that n should be substituted with $n - k$ as k variables are guessed first. As k increases, the degree of semi-regularity decreases, but at the expense of the probability of making the correct guess.

Having determined the degree of regularity, the third step is to compute the complexity of doing linear algebra on a Macaulay matrix whose polynomials have this degree. Let $d_{reg}(k)$ denote this degree. The number of monomials of degree d is $\binom{d+n-1}{n-1}$, as is easily visualized via the stars and bars argument. For example, if $n = 4$ and $d = 5$, then there are $n + d - 1$ positions for d stars and $n - 1$ bars.

$$x_1^3 x_3 x_4 \qquad \longleftrightarrow \qquad \underbrace{\star \ \star \ \star}_{x_1} \ \mid \underbrace{\ }_{x_2} \mid \ \underbrace{\star}_{x_3} \ \mid \ \underbrace{\star}_{x_4}$$

The number of monomials of degree d *or less* is $\binom{d+n}{n}$. One can always adjoin an extra homogenizing variable z that is multiplied with every term until it is of the requisite degree; this number is therefore the same as the number of monomials in $n + 1$ variables of degree d.

After guessing k variables, the width of the extended Macaulay matrix is $N = \binom{d_{reg}(k)+n-k}{n-k}$. Gaussian elimination in this matrix requires N^3 field operations. If fast matrix multiplication techniques are used, the exponent is $\alpha \approx 2.373$ [62]. For the Wiedemann method the exponent is 2 but the complexity accrues another factor $\binom{n-k}{2}$. In particular, this method requires $2\binom{d_{reg}(k)+n-k}{n-k}$ matrix-vector and inner product multiplications, where the size of the vectors is $\binom{d_{reg}(k)+n-k}{n-k}$ and the matrix has at most $\binom{n-k+3}{2}$ nonzero elements, thus making for a complexity of $O(\binom{n-k+3}{2}\binom{d_{reg}(k)+n-k}{n-k}^2)$. The Landau notation hides the constant associated with the blockwise aspect as well as terms deriving from non-bottleneck processing.

The k guessed variables correspond to a solution with probability q^{-k}. On a quantum computer, one can Groverize this guessing to find a correct solution after only $q^{k/2}$ iterations, where one iteration requires performing the entire Gröbner basis algorithm. So an estimate of the total complexity is given by

$$C_{\mathcal{MQ}}(k) = O\left(q^{k/2} \cdot \binom{n-k+3}{2}\binom{d_{reg}(k)+n-k}{n-k}^2\right) \quad , \qquad (4.8)$$

or rather, the minimum of this quantity for various k.

Figure 4.1 plots the complexity of an algebraic attack on a system of quadratic equations with various values for the field size q and for the number of equations m. The number of variables is chosen as $n = m$ because this parameter choice leads to the hardest to solve system.

4.1.2 Isomorphism of Polynomials

Another hard problem that pops up frequently in the context of MQ cryptography is the isomorphism of polynomials (IP) problem [110]. Informally, the task is to find a pair of linear or affine transforms that, composed on either side of one given multivariate quadratic polynomial map, yields the other given multivariate quadratic polynomial map. A formal definition follows.

There are many subtle variants. The *decision* variant asks only to decide whether such a pair (T, S) exists. The *homogeneous* variant allows only terms of degree exactly two, and moreover requires dropping the constant part of the affine transforms, *i.e.*, $T \in \mathsf{GL}_m(\mathbb{F}_q)$ and $S \in \mathsf{GL}_n(\mathbb{F}_q)$, because otherwise the problem is easy. Furthermore, in the *isomorphism of polynomials with one secret (IP1S)* problem, $T = \mathsf{Id}$; and the *morphism of polynomials (MP)* [111] considers generic

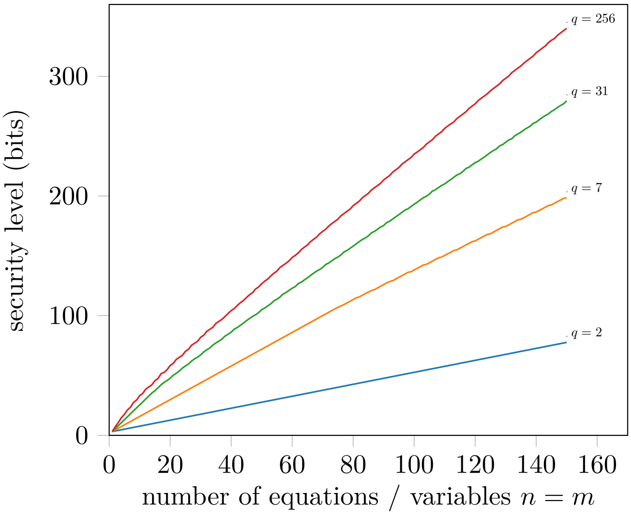


Figure 4.1: Complexity of Gröbner basis attack.

Hard Problem 4.2: IP Problem

Parameters: number of polynomials m, number of variables n.

Given: two lists $\mathbf{F}, \mathbf{P} \in (\mathbb{F}_q[\mathbf{x}]_{\leq 2})^m$ of m polynomials of degree at most 2 in n variables $(x_1, \ldots, x_n) = \mathbf{x}^\mathsf{T}$ over a finite field $\mathbb{F}_q$ such that for some invertible affine transformations $T \in \mathsf{AGL}_m(\mathbb{F}_q)$ and $S \in \mathsf{AGL}_n(\mathbb{F}_q)$, $\mathbf{P} = T \circ \mathbf{F} \circ S$.

Task: Find a pair $(T, S) \in \mathsf{AGL}_m(\mathbb{F}_q) \times \mathsf{AGL}_n(\mathbb{F}_q)$ such that $\mathbf{P} = T \circ \mathbf{F} \circ S$.

matrices S and T, not necessarily invertible and possibly not even square. Most importantly, in the *extended isomorphism of polynomials (EIP)* problem[2], the solver is not given two lists of polynomials, but one. Instead of finding a pair of affine transformation that turns the given system of polynomials into another one, the transformations should turn the system of polynomials into one with a particular structure — structure that may be used by the secret key holder to efficiently find inverses to given images, as reflected by Fig. 4.2. If the EIP problem is hard for a given structural mechanic for computing inverses, then it may be argued that the given public key is indistinguishable from a random

[2] The first mention of this problem I could find was in Petzoldt's dissertation [113, §2.3.2.] but essentially all bipolar MQ cryptosystems rely on the hardness of this problem — even the ones that came before.

MQ map. This generates a trapdoor function, because from the point of view of the adversary, inversion is hard.

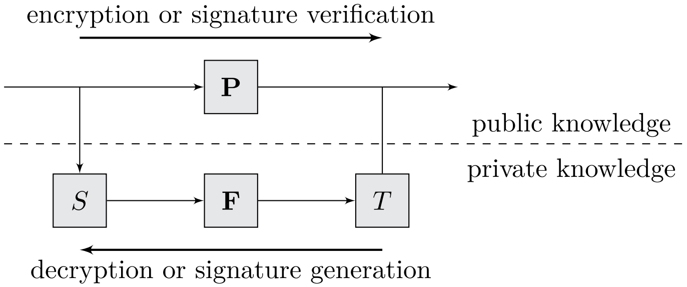

encryption or signature verification
P
public knowledge
private knowledge
S
F
T
decryption or signature generation

Figure 4.2: Bipolar construction for multivariate quadratic cryptosystems.

To date, the best attack on IP is due to Bouillaguet *et al.* [30], which has a heuristic complexity of $\sim q^{n/2}$. However, there are many caveats. For instance, as q rises, at some point an algebraic search for (S, T) whereby the coefficients of these matrices are variables will outperform this attack; at this point the complexity is largely independent of q. When $m = 1$ the problem is trivial because quadratic forms admit a canonical representation; also when $m = 2$ an algorithm by Plût *et al.* solves the problem in polynomial time [115]. In the case of the EIP problem, the best attack depends on the particular strategy for computing inverses, because this strategy induces the structure on the polynomials that might make EIP easy. Indeed, one of the most generic open questions in MQ cryptography is how to generate MQ maps that enable efficient inverse computation but for which EIP is hard.

4.2 Lattices

A lattice is the discrete analogue of an infinite subspace. As such, they pop up in various places in the context of discrete algebra and number theory. A lattice $\mathcal{L}$ is given by a set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$ called a spanning set or, if they are linearly independent, a basis. A basis for a given lattice is not unique, and some are more useful than others.

Definition 15 (lattice)**.** *A* lattice $\mathcal{L} \subset \mathbb{R}^m$ *is the set of integer linear combinations of a spanning set or basis* $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$*:* $\mathcal{L} = \{\sum_{i=1}^n z_i \mathbf{b}_i \,|\, z_i \in \mathbb{Z}\}$.

Lattice basis reduction is the non-trivial task of finding another basis for the same lattice but composed of short vectors, usually in the ℓ_2 norm. Lattice reduction algorithms such as LLL [93] and BKZ [122] are the go-to tool for attacking number-theoretic cryptosystems where small portions of information are leaked, such as factorizing of RSA moduli if a part of one of the prime factors is known [40], or computing the secret key from ECDSA signatures where the nonces are partially known [107].

However, in high dimensions, *i.e.*, several hundreds or more, lattice reduction seems *hard* — even for quantum computers. The canonical lattice problem, the approximate shortest vector problem (SVP_γ), of finding a short vector whose length is at most a given factor γ off from the shortest nonzero vector, is **NP**-hard[3] for constant approximation factors [8], and empirically infeasible for polynomial ones.

Hard Problem 4.3: SVP_γ

Parameters: an approximation factor $\gamma \in \mathbb{R}_{>0}$.

Given: a lattice $\mathcal{L}$.

Task: find a vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\|_2 \leq \gamma \cdot \lambda_1(\mathcal{L})$ where $\lambda_1(\mathcal{L}) = \mathsf{min}_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}}(\|\mathbf{v}\|_2)$.

It stands to reason then, that the cryptosystems whose most effective attack is lattice reduction in high dimension achieve post-quantum security.

4.2.1 SIS and LWE

The Short Integer Solution (SIS) and Learning with Errors (LWE) hard problems are popular source material for the generation of public key cryptosystems for at least two reasons. First, they are expressible in the language of simple linear algebra. Second, they both enjoy a worst case to average case reduction. This reduction guarantees average case hardness, assuming that the underlying lattice problem is hard.

Informally, the SIS problem asks to find a short solution to a under-determined system of linear equations. Conversely, the LWE problem asks to find a solution to an over-determined system of noisy linear equations, *i.e.*, equations that hold up to some small noise.

[3] This **NP**-hardness result holds with respect to *randomized* reductions. Standard **NP**-hardness results hold for deterministic reductions.

In the present context of lattice problems, and generally in contexts where a vector's *length* is important, the field $\mathbb{F}_q$ over which the equations are defined, is a prime field. This allows one to identify integers with the coordinates of the vector and thus to define its length in a straightforward way.

Hard Problem 4.4: SIS Problem

Parameters: dimensions m and n with $n > m$, length-bound $\beta \in \mathbb{R}_{>0}$.

Given: a matrix $A \in \mathbb{F}_q^{m \times n}$.

Task: find a vector $\mathbf{x} \neq 0$ such that $A\mathbf{x} = 0 \bmod q$ and such that $\|\mathbf{x}\|_2 \leq \beta$.

Hard Problem 4.5: LWE Search Problem

Parameters: a discrete Gaussian distribution ψ over $\mathbb{F}_q$ of "small" elements.

Given: query-access to a sample-generator G that outputs samples $(\mathbf{a}_i, b_i)$ where $b_i = \mathbf{a}_i^\mathsf{T}\mathbf{s} + e_i \bmod q$ with $\mathbf{a}_i \xleftarrow{\$} \mathbb{F}_q^n$ and $e_i \sim \psi$ and for some unknown but constant $\mathbf{s} \in \mathbb{F}_q^n$.

Task: find $\mathbf{s}$.

The link with lattices is readily observed. The set of solutions to $A\mathbf{x} = 0 \bmod q$ is a lattice; SIS solutions are short vectors in this lattice. Likewise, the vectors $(\mathbf{a}_i, b_i - e_i)$ lie in a lattice. These lattice points are hidden precisely by the added noise e_i, but the adversary who manages to separate the noise from the lattice point for enough samples can rapidly recover the secret vector $\mathbf{s}$. The link with lattices is even more apparent from the worst case to average case reductions. Ajtai [7] shows that an algorithm that solves *random* SIS instances can be made to efficiently solve a *given* instance of SVP_γ for a polynomial approximation factor, *i.e.*, with $\gamma = n^c$ for some constant c. Regev [117] shows that, when $q > 2n$, an algorithm that solves LWE can be used by a quantum algorithm to efficiently solve a given instance of SVP with approximation factor $\tilde{O}(n/\alpha)$ where $\alpha \in (0, 1)$ is a parameter related to the distribution ψ of small elements.

The description of SIS and LWE instances does consist of large matrices of roughly $m \times n$ random coefficients, where in the case of LWE, m is the number of samples queried or queriable by the solver. However, it is by no means clear that any security is lost by switching to structured matrices, such as cyclic or nega-cyclic matrices for every $n \times n$ block. A user can therefore get away with storing only the first row or column as the other elements can be inferred from this. Algebraically, this corresponds to arithmetic in the polynomial ring $\mathbb{Z}_q[x]/\langle x^n \pm 1\rangle$; the names for the corresponding problems have converged to Ring-SIS and Ring-LWE [95, 112, 129, 96]. Endowing this algebra again with a module-structure further generalizes the problems to Module-SIS and Module-LWE [91]. This last pair of variants have the bandwidth advantage of its immediate ring-based predecessors, while salvaging to some extent the potential security gains associated with unstructured lattices.

4.2.2 Lattice Reduction

LLL. Following standard practice in the context of the LLL algorithm, we consider basis vectors as row vectors: $\mathbf{b}_i \in \mathbb{Z}^{1\times m}$. A given basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ for a lattice $\mathcal{L}$ may be identified with a *basis matrix* $B \in \mathbb{Z}^{n\times m}$ obtained by stacking the row-vectors on top of each other. In this case $\mathcal{L} = \{\mathbf{z}B \,|\, \mathbf{z} \in \mathbb{Z}^{1\times n}\}$. Two basis matrices B_1 and B_2 define the same lattice if and only if there is a unimodular matrix, *i.e.*, with integer elements and determinant ±1, such that $B_1 = UB_2$. An equivalent formulation of the task of lattice basis reduction is to find a unimodular matrix such that the resulting basis matrix is small. The *volume*, or *determinant*, of a lattice is an invariant defined as, for any basis matrix B, $\mathsf{Vol}(\mathcal{L}) = \sqrt{\mathsf{det}(BB^{\mathsf{T}})}$. Incidentally, this quantity is also the volume of the fundamental parallelepiped, and this correspondence leads to the observation that a somewhat short basis must also be somewhat orthogonal and vice versa.

Recall the Gram-Schmidt method for obtaining an orthogonal basis $(\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ from a given basis $(\mathbf{b}_1, \ldots, \mathbf{b}_n) \in (\mathbb{R}^{1\times m})^n$. The process is inductively defined as computing $\mu_{i,j} = \frac{(\mathbf{b}_i^* \cdot \mathbf{b}_j)}{(\mathbf{b}_i^* \cdot \mathbf{b}_i^*)}$ for $1 \leq i \leq j \leq n$ and $\mathbf{b}_j^* = \mathbf{b}_j - \sum_{i=1}^{j-1} \mu_{i,j}\mathbf{b}_i^*$, starting with $\mathbf{b}_1^* = \mathbf{b}_1$. Drop the $\mu_{i,j}$ appropriately in the (j, i) position of a lower-triangular square matrix M, and observe that $MB = B^*$, B^* has orthogonal rows, and $\mathsf{det}(M) = 1$. However, neither B^* nor M are guaranteed to be integer matrices, and so B^* will not span the same lattice. The next best thing is to round the coefficients $\mu_{i,j}$ to the nearest integer and find the new basis vectors $\hat{\mathbf{b}}_j = \mathbf{b}_j - \sum_{i=1}^{j} \lfloor \mu_{i,j} \rceil \hat{\mathbf{b}}_i$ with $\hat{\mathbf{b}}_1 = \mathbf{b}_1$.

The celebrated LLL algorithm [93] combines this rounded Gram-Schmidt procedure with a criterion for swapping the order of an adjacent pair of basis vectors. In particular, the LLL algorithm procedurally computes the rounded Gram-Schmidt "orthogonalization" $(\mathbf{b}_1, \ldots, \mathbf{b}_{k-1})$ for k going from 2 to $n + 1$. However, it only proceeds to the next increment of k if the condition $\|\mathbf{b}_k^* + \mu_{k,k-1}\mathbf{b}_{k-1}^*\|_2 \geq \eta \|\mathbf{b}_{k-1}^*\|_2$ is satisfied; otherwise $\mathbf{b}_{k-1}$ and $\mathbf{b}_k$ are swapped and k is decremented —unless it is already 2— and the new $\mathbf{b}_{k-1}$ is Gram-Schmidt reduced instead. When $k = n + 1$ the algorithm terminates. The η

in the swapping criterion is a parameter that is usually set to 0.99 in practice, and polynomial running time is guaranteed when it lies in the interval $(0.25, 1)$.

The unreasonable effectiveness of the LLL algorithm for small-scale Diophantine problems —which include plenty of practical cryptanalyses— stems from the rather innocuous LLL bound on produced basis vectors. The shortest vector produced by LLL satisfies $\|\mathbf{b}_1\| \leq \left(\frac{4}{4\eta-1}\right)^{(n-1)/4} \cdot \mathsf{Vol}(\mathcal{L})^{1/n}$. Compare this bound to the Minkowski bound which guarantees that there is a nonzero lattice vector $\mathbf{v}$ whose length is bounded by $\|\mathbf{v}\| \leq \sqrt{n} \cdot \mathsf{Vol}(\mathcal{L})^{1/n}$. If the dimension n is small enough such that the relative difference $\left(\frac{4}{4\eta-1}\right)^{(n-1)/4} / \sqrt{n}$ is much smaller than $\mathsf{Vol}(\mathcal{L})^{1/n}$, then LLL will find the shortest solution with overwhelming probability.

The story is completely different for large n, because then the difference between the LLL bound and the Minkowski bound explodes. In order to find short vectors, one has to switch to another lattice basis reduction algorithm.

BKZ and Core SVP Hardness. The Block Korkin-Zolotarev (BKZ) algorithm [122] combines LLL with calls to an SVP oracle. At each iteration, the SVP is computed in the projected sub-lattice of dimension equal to or less than the block dimension b, spanned by the next b working basis vectors, or fewer if there are not so many independent vectors left. If this SVP solution is equal to the next basis vector, the algorithm increments a counter and shifts the window of vectors spanning the sub-lattice by one. Otherwise the SVP solution is inserted into the basis, LLL is run, and the counter is set to zero and the window shifted back to the start. In this way, the BKZ algorithm progressively builds a basis that is reduced in a much stronger sense than the outputs of the LLL algorithm. However, this improved basis comes at the expense of an exponential running time. The BKZ 2.0 algorithm [37] provides a number of improvements to make the SVP oracle faster, and additionally comes with a parameter determining the number of iterations. In practice this number is set to something feasible, trading running time for quality of the output basis.

The reliance on the SVP oracle spurred the authors of the celebrated NewHope cryptosystem [12] to propose a pessimistic estimation of the complexity of lattice problems, which has since seen widespread adoption [10]. The summary here applies some simplifications. The name “core SVP” stems from the fact that the argument considers the complexity of only one SVP oracle query; the number of times such a query is made within the BKZ algorithm is ignored. The type of algorithm that solves the SVP is assumed to be a sieve: sieve type algorithms have asymptotically better running time, though in practice the alternative of enumeration performs better. The complexity of the sieve for classical attackers is estimated at roughly $2^{0.292b}$; an attacker capable of using quantum computations can leverage Grover search and drop this to $2^{0.265b}$. In a more paranoid situation, the attacker can perform faster-than-quantum computations but as sieve algorithms still require explicitly building lists of $2^{0.2075b}$ items, one can use this number as a lower bound on the complexity.

The next question is, for which value of the block size b will BKZ output the desired short vector? From the point of view of the attacker, this short vector should be either equivalent to the secret key or else capable of undermining the security of the cryptosystem in another way. In either case, for a well-designed system, this vector will be short enough to force the attacker to choose a rather large b to find it, and in turn to force him to run a very expensive SVP solver. Let $\mathbf{s}$ denote this secret short vector.

The quality of a reduced lattice basis, such as those output by the LLL or BKZ algorithms, can be characterized by the *root-Hermite factor* δ [63], which is defined via $\|\mathbf{b}_1\|_2 = \delta^m \, \mathsf{Vol}(\mathcal{L})^{1/m}$, where $\mathbf{b}_1$ is the shortest nonzero vector in the basis. Under the geometric series assumption [121], the Gram-Schmidt vectors of the output of BKZ have length $\|\mathbf{b}_i^*\| = \delta^{m-2i+1} \cdot \mathsf{Vol}(\mathcal{L})^{1/m}$. Moreover, Chen [36] gives an asymptotic limit for δ under the same assumption: as n approaches infinity, $\delta \approx \left(\frac{b}{2\pi e}(\pi b)^{\frac{1}{b}}\right)^{\frac{1}{2(b-1)}}$. The secret short vector $\mathbf{s}$ will be found if its projection onto the last b Gram-Schmidt vectors is shorter than $\mathbf{b}^*_{m-b}$. Approximating the size of this projected vector as $\sqrt{\frac{b}{m}} \cdot \|\mathbf{s}\|$, this leads to the criterion for success

$$\sqrt{\frac{b}{m}} \cdot \|\mathbf{s}\| \leq \delta^{2b-m+1} \cdot \mathsf{Vol}(\mathcal{L})^{1/m} \quad . \tag{4.9}$$

For SIS and LWE problems, the lattice is generally q-ary, meaning that for all $\mathbf{v} \in \mathbb{Z}^m$ the membership question $\mathbf{v} \overset{?}{\in} \mathcal{L}$ is determined by $\mathbf{v} \bmod q$. For q-ary lattices with prime q and of dimension n and embedding dimension m, the volume is given by $\mathsf{Vol}(\mathcal{L}) = q^{m-n}$. Figure 4.3 plots the quantum complexity as a function of the lattice dimension n and the modulus q. The remaining free parameters are fixed to typical values: the embedding dimension is $m = 2n$ and length of the secret short vector is $\sqrt{mn/2\pi}$ corresponding to a standard deviation of $\sigma = \sqrt{\frac{n}{2\pi}}$ for the LWE distribution ψ [11].

The top three lines halt abruptly because in those cases no block size b can satisfy Eqn. 4.9. One possible perspective on the cause of this phenomenon is

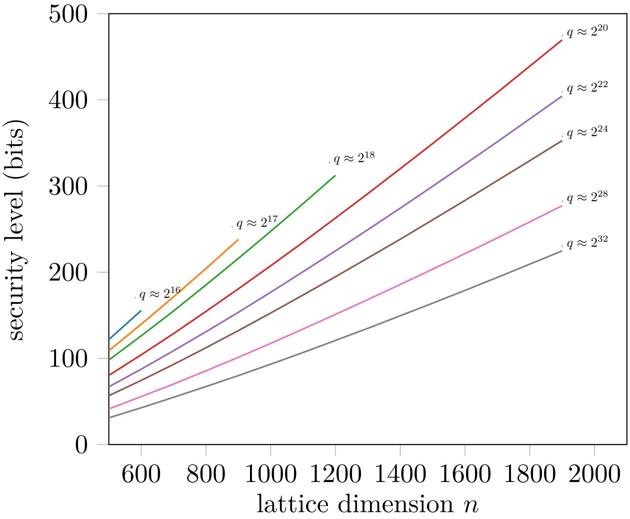


Figure 4.3: Complexity of lattice reduction attack.

that the rule of thumb equating the standard deviation to $\sigma = \sqrt{\frac{n}{2\pi}}$ results in a rather large value, which then causes the right hand side to always be larger than its left hand counterpart; indeed, the NewHope parameters specify $\sigma = \sqrt{8} \ll \sqrt{\frac{n}{2\pi}} = \frac{1024}{2\pi} \approx 163.0$. The constraint $\sigma \geq \sqrt{\frac{n}{2\pi}}$ stems from a requirement to resist the Arora-Ge linearization attack [14]. However, this attack only applies when the attacker has access to an unlimited number of samples, which is not the case for an attack on the NewHope cryptosystem. An interesting open question is therefore whether the NewHope security estimation can be lifted to the regime where Arora-Ge does apply, or which alternative should be used there instead.

4.3 Other Hard Problems

Algorithms for computing Gröbner bases and reducing lattice bases are versatile tools in the toolbox of the algebraic cryptanalyst. Their complexities are limiting factors on the parameter selection for various cryptosystems. This raises the tantalizing possibility of post-quantum hard problems for which both basis strategies either fail completely or are so infeasible that something else is the limiting factor.

This is obviously the case for the other branches of post-quantum cryptography. In particular:

- *Hash-based cryptography* relies on the collision resistance and second preimage resistance properties of hash functions to generate signature schemes such as SPHINCS [25]. While these hash functions can be attacked algebraically, the very high degree polynomials make for an infeasible Gröbner basis computation. There is no lattice to speak of. In practice, the best performing attacks are either symmetric cryptanalysis or generic black box attacks.
- *Code-based cryptography* relies on the difficulty of decoding noisy codewords of a random error-correcting code. The classic example is the McEliece cryptosystem [98]. It is possible to identify a lattice with these cryptosystems but with regards to lattice-based cryptography there are two important differences. First, the secret is a short vector with respect to the Hamming weight rather than the Euclidean norm, meaning that all nonzero coefficients may be arbitrarily large[4]. Second, the dimension is typically an order of magnitude larger than the lattice dimension for lattice-based cryptosystems. Both differences conspire to make lattice reduction wildly infeasible and in practice combinatorial methods such as information set decoding [97] are the bottleneck attack.
- *Isogeny-based cryptography* relies on the difficulty of finding isogenies between elliptic curves over finite fields, and features homomorphisms that makes key exchange possible [56, 33]. There is no lattice to speak of. An algebraic attack first needs to decide on the targeted representation of the isogenies. If the fractional map representation is targeted, a Gröbner basis attack would have to use an exponential number of variables. Otherwise, if it is targeting the coefficients for the torsion subgroup generators P and Q, it would have to find a way to mix elements from different algebras. The former task is infeasible, the second is ill-defined.

In the course of my research, I have paid particular attention to two rather new hard problems which in my estimation belong in this list. In both cases, it is possible to identify a lattice with the space of solutions, although not all lattice points correspond to solutions. However, in both cases this lattice contains parasitical solutions — lattice vectors that are *shorter* than the sought-after secret. This presence of parasitical solutions makes a lattice reduction procedure irrelevant as it is destined to find vectors that are too small. Among the vectors in the lattice that are of the right size, there are too many to choose from, and only an insignificant proportion of them correspond to the secret.

[4] But since these lattices are q-ary also, “arbitrarily large” means at most $\mathsf{max}(\{0, \dots, q-1\})$.

4.3.1 Short Solutions to Nonlinear Equations.

The Short Solutions to Nonlinear Equations (SSNE) problem was introduced by myself and Bart Preneel at NuTMiC 2017 as the logical merger of the MQ and SIS problems [133]. Algebraic attacks fail because Gröbner basis algorithms cannot distinguish between solutions and short solutions, whereas lattice reduction fails because it cannot distinguish between solutions and non-solutions. It is possible to identify a region of parameter space where —conjecturally, but quite plausibly— brute force search is the best performing attack.

Hard Problem 4.6: SSNE Problem

Parameters: number of polynomials m, number of variables n, length bound $\beta \in \mathbb{R}_{>0}$.

Given: a list $\mathbf{P} \in (\mathbb{F}_q[\mathbf{x}])^m$ of m non-affine[a] polynomials in n variables $(x_1, \ldots, x_n) = \mathbf{x}^\mathsf{T}$ over a prime field $\mathbb{F}_q$.

Task: find a solution $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathbf{P}(\mathbf{x}) = \mathbf{0} \bmod q$ and such that $\|\mathbf{x}\|_2 \leq \beta$.

[a] In this context, non-affine means: at least one of the polynomials has degree at least two.

We identify 6 design principles to take into account to ensure the problem is hard. The amendment to principle 2 is adopted from a follow-up paper on obtaining zero-knowledge proofs and signature schemes from SSNE [131], and this amendment makes principle 4 superfluous. The principles for targeting κ bits of security against classical computers are:

1. $\beta \geq \kappa$;

2′. $m(\log_2 q - \log_2 \beta) \geq \kappa$;

3. $\|\mathbf{x}\|_2^2 \geq q$ for all solutions $\mathbf{x}$;

5. $\mathsf{rank}(W^\mathsf{T} + W) \geq \mathsf{dim}(V(\mathbf{P}))$ if the length constraint is generalized to $\mathbf{x}^\mathsf{T} W \mathbf{x} \leq \beta^2$;

6. $o > m \implies \frac{n-o+m}{m+1} \log_2 q \geq \lambda/n + \log_2 \beta$, where $o = \mathsf{max}_o$ s.t. $m(o+1)/2 \leq n$ and $o < n$.

The non-linearity of the equations, along with design principle 3, is an essential property in order to thwart lattice reduction attacks. Polynomial equations in the variables $\mathbf{x} = (x_1, \ldots, x_n)^\mathsf{T}$ can always be considered as linear equations in the extended vector of variables $(1, x_1, \ldots, x_n, x_1^2, x_1x_2, \ldots)$. If there is a solution to the polynomial equations such that the extended vector is small, then lattice reduction algorithms can find it. However, design principle 3 guarantees that this extended vector is larger than q; since this is not at all short, lattice basis reduction will find other vectors instead — vectors that do not correspond to solutions to the polynomial equations. In principle, this constraint can be relaxed to require only that $\|\mathbf{x}\|_2$ is significantly larger than the length of solutions that may be expected from the Gaussian heuristic. It remains an open question, however, whether this relaxation comes with a material benefit from the designer's point of view.

4.3.2 Sparse Integers in a Mersenne Ring.

Let $n = 2^p - 1$ be a Mersenne prime. Arithmetic in the ring $\mathbb{Z}/n\mathbb{Z}$ is somewhat Hamming weight preserving: $\textsc{hw}(a + b) \leq \textsc{hw}(a) + \textsc{hw}(b)$ and $\textsc{hw}(a \times b) \leq \textsc{hw}(a) \times \textsc{hw}(b)$ where $\textsc{hw} : \mathbb{Z}/n\mathbb{Z} \rightarrow \mathbb{Z}$ denotes the Hamming weight of the integers' binary expansion. This somewhat homomorphic property suggests the possibility for a noise-based key encapsulation mechanism or public key encryption scheme, similar to the noisy schemes put forward by the lattice and coding theory branches of post-quantum cryptography.

These observations were first made by Aggarwal *et al.* in a paper published on IACR ePrint [4] where they also propose a cryptosystem similar in spirit to NTRU [66] but relying on the arithmetic of this Mersenne ring. A later version of that ePrint paper [5] updates the design strategy from NTRU-like to resembling a noisy Diffie-Hellman protocol, matching the authors' submission to the NIST project called Mersenne-756839 [3]. Independently, I developed essentially the same noisy Diffie-Hellman based cryptosystem and submitted it under the moniker "Ramstake" [130].

The most salient feature of the noisy Diffie-Hellman protocol, shown in Fig. 4.4, is its simplicity. The algebra involved is just multiplication and addition of integers; no vectors or matrices or polynomials are involved. Even modular reduction is child's play: in the ring of integers modulo a Mersenne number n, reduction is performed by splitting the binary expansion into chunks of p bits, and summing the chunks. If the Hamming weight bound ω is sufficiently small, *i.e.*, an order of magnitude smaller than p, then Alice and Bob will agree approximately on the same number. In particular, the binary expansions of $E_A = acG + ad$ and $E_B = acG + bc$ are roughly $4\omega^2$ bits apart. However, going from approximately equal secrets to exactly equal secrets requires transmitting an additional message that involves an error-correcting code, and this transformation makes the system rather more complex as well as necessarily interactive.

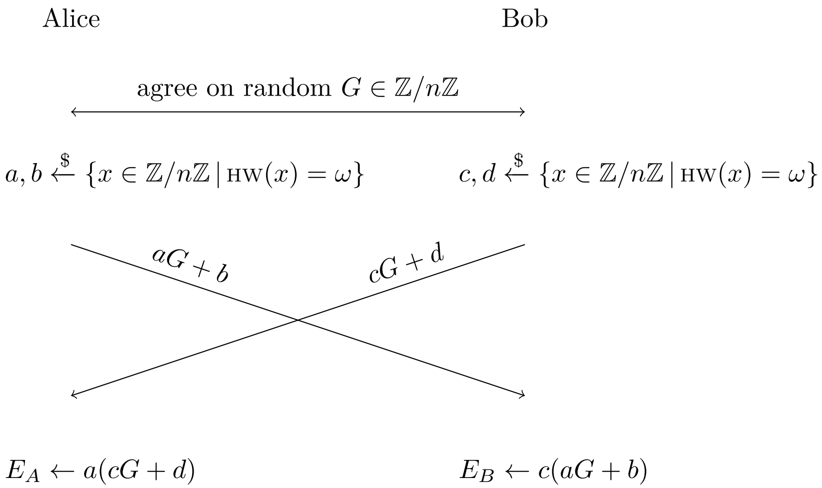

Alice
Bob
agree on random $G \in \mathbb{Z}/n\mathbb{Z}$
$a, b \xleftarrow{\$} \{x \in \mathbb{Z}/n\mathbb{Z} \mid \text{HW}(x) = \omega\}$
$c, d \xleftarrow{\$} \{x \in \mathbb{Z}/n\mathbb{Z} \mid \text{HW}(x) = \omega\}$
$aG + b$
cG + d
$E_A \leftarrow a(cG + d)$
$E_B \leftarrow c(aG + b)$

Figure 4.4: Noisy Diffie-Hellman protocol in a Mersenne ring.

The hard problem for the NTRU-like cryptosystem is to find low Hamming weight integers f and g such that their fraction is equal to a given non-sparse integer H. The noisy Diffie-Hellman protocol requires the hardness of what is essentially an affine version of this problem, called the *Low Hamming Combination (LHC) Problem.* It additionally requires that the analogues of the computational and decisional Diffie-Hellman problems are hard; these problems are called *Low Hamming Diffie-Hellman Search (LHDHS) Problem* and *Low Hamming Diffie-Hellman Decision (LHDHD) Problem*, respectively. These requirements follow from a straightforward depiction of the protocol such as that of Fig. 4.4.

Hard Problem 4.7: LHC Problem

Parameters: Mersenne prime n, weight bound $\omega \ll n$.

Given: two integers $G, H \in \mathbb{Z}/n\mathbb{Z}$

Task: find two integers $a, b \in \mathbb{Z}/n\mathbb{Z}$ such that $\text{HW}(a) \leq \omega$ and $\text{HW}(b) \leq \omega$ and $aG + b = H$.

Hard Problem 4.8: LHDHS Problem

Parameters: Mersenne prime n, weight bound $\omega \ll n$, noise threshold t.

Given: three integers $G, H, F \in \mathbb{Z}/n\mathbb{Z}$ such that there are sparse integers $a, b, c, d \in \mathbb{Z}/n\mathbb{Z}$ of Hamming weight at most ω such that $aG + b = H$ and $cG + d = F$.

Task: find an integer $E \in \mathbb{Z}/n\mathbb{Z}$ such that the Hamming distances $\textsc{hd}(E, aF) \leq t$ and $\textsc{hd}(E, cH) \leq t$.

Hard Problem 4.9: LHDHD Problem

Parameters: Mersenne prime n, weight bound $\omega \ll n$, noise threshold t.

Given: four integers $G, H, F, E \in \mathbb{Z}/n\mathbb{Z}$ such that there are sparse integers $a, b, c, d \in \mathbb{Z}/n\mathbb{Z}$ of Hamming weight at most ω such that $aG + b = H$ and $cG + d = F$.

Task: decide whether the Hamming distances $\textsc{hd}(E, aF) \leq t$ and $\textsc{hd}(E, cH) \leq t$.

To date, the best performing attack against these hard problems is the so-called *slice-and-dice attack* due to Beunardeau *et al.* [27]. The attack targets the LHC problem and attempts to recover a, b from G, H. It starts by choosing a random partition of the binary expansions of a and b. Each partition is identified with a new variable a_i or b_i such that $a = \sum_i 2^{s_{a,i}} a_i$ and $b = \sum_i 2^{s_{b,i}} b_i$, where $s_{a,i}$ and $s_{b,i}$ are the starting positions that define the partition. Then the single equation $aG + b = H$ corresponds to a multivariate equation in terms of the a_i and b_i.

However, half the parts are labeled inactive and the other half active. If it is true that all the 1-bits of the binary expansions of a and b happen to lie in active partitions, then the value of all inactive variables is zero. This means in turn that the equation

$$\left(\sum_{active\, i} 2^{s_{a,i}} a_i \right) G + \left(\sum_{active\, i} 2^{s_{b,i}} b_i \right) = H \tag{4.10}$$

has a solution, which can be found using LLL [93].

The running time of this attack is determined by the probability that a partition and labeling is correct. This event occurs with probability $2^{-2\omega}$ as there are 2ω bits that have to lie in active intervals, which make up half the possible space.

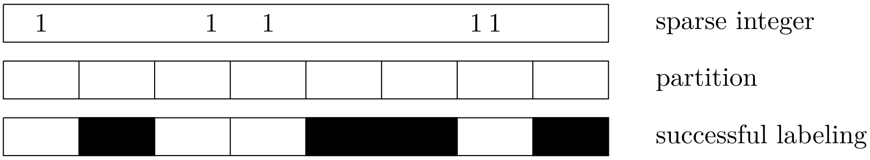


Figure 4.5: Partition and successful labeling in Beunardeau *et al.*'s slice-and-dice attack.

Ignoring the cost of running LLL, this makes for a classical running time of $2^{2\omega}$. Quantumly, one may expect to Groverize the random guess and obtain a running time of 2^{ω}.

Note that while this attack does involve lattice basis reduction, it is not the bottleneck. If a brilliant student has a breakthrough result dramatically decreasing the complexity of the SVP oracle, the security of this cryptosystem will remain unaffected. In a classification of cryptosystems by the hard problem they rely on, the Mersenne-756839 and Ramstake cryptosystems cannot be classified as lattice-based.

Nevertheless, it is possible to identify a lattice with the space of solutions and in this lattice the vector that is identifiable with the solution (a, b) is short. Consider the basis of $(2n+1)$-dimensional row-vectors $(K \cdot 2^i G, 0, \ldots, 0, 1, 0, \ldots, 0)$ where the 1 is in position $i+1$ for $i \in \{0, \ldots, n-1\}$, $(K \cdot 2^i, 0, \ldots, 0, 1, 0, \ldots, 0)$ where the 1 is in position $n+i+1$ for $i \in \{n, \ldots, 2n-1\}$, and $(K \cdot n, 0, \ldots, 0)$, all for a sufficiently large integer K. Then, using the bit expansion of a and b, the vector $(0, a, b)$ is a short vector in this lattice, obtained by applying the weight vector $(a, b, -1)$. This lattice basis is constructible by the adversary from public information. However, this lattice contains parasitical solutions: by subtracting, say, 2 times the second basis vector from the third one we obtain a vector with norm $\sqrt{5}$. Even if the adversary manages to find a sufficiently short reduced basis for this lattice —quite the challenge, given the dimension— the solution $(0, a, b)$ with length $\sqrt{2\omega^2}$ will fail to stand out from the multitude of vectors whose norms are smaller.

Chapter 5

Conclusions

The threat of future (and possibly present) quantum computers poses a unique challenge for designers of public key cryptosystems. When they are built, quantum computers will be able to efficiently solve a class of computational problems that has proved nigh indispensable for the generation of public key cryptosystems. These will be broken as a consequence of this efficient solution. It is therefore fitting and timely to adapt the field of public key cryptography to take this threat into account.

On the one hand, the foundational hard problems from which public key cryptosystems derive their security must be made to resist attacks that run on quantum computers. This means exchanging problems like the integer factorization problem and the discrete logarithm problem for hard problems based on systems of polynomial equations or based on noisy linear algebra, to name just a few popular choices. This is the eponymous mathematical aspect of post-quantum cryptography.

On the other hand, the security proofs that demonstrate the cryptosystems' security must be reconsidered as well. Up until recent years, security proofs have implicitly considered a classical computing model for the adversary. However, when adversaries in the quantum computing model are considered, many of these proofs and proof techniques are invalid. A complete argument for post-quantum security therefore mandates security proofs and proof techniques that hold in the quantum computing model, in addition to the classical one. This is the provable security aspect of post-quantum cryptography.

This introduction, being an introduction, can only touch on so many topics without becoming a comprehensive treatment. Instead, this text opts to convey only the basic principles of quantum computation, provable security, and only two branches of hard mathematical problems. Sophisticated quantum algorithms for specific computational problems and new proof techniques for the quantum computing model are a recurring feature of post-quantum themed conferences. The omission is starker still where the hard problems are concerned: aside from MQ-based and lattice-based problems, three branches of post-quantum problems have only been touched superficially, and cryptography based on non-commutative groups has not been mentioned at all (until now). The focus on the most basic functionalities reflects the urgent demand. However, fancy constructions like homomorphic encryption, multi-party computation, and blind signatures are a major staple of public key cryptography whose lifting to the post-quantum domain has remained largely restricted to lattice-based cryptosystems in the classical random oracle model.

With respect to the topics that are covered, this introduction represents a summary of the state of the art. While some aspects may be less likely than others to see change in coming years, progress is virtually certain. It is worthwhile then, to pause and reflect on some open issues.

Quantum Algorithms. The field of quantum algorithms is very much an active topic, but the intersection between quantum algorithm designers and cryptographers or computer algebra specialists remains rather small. In light of the increasing attention being paid to the quantum adversarial model, the exact quantum hardness of hard problems is a question of prime concern. Breakthroughs of the magnitude of Shor's algorithm are unlikely because low-hanging fruit of this kind has been made rare. However, it remains likely that careful quantum optimizations may improve standard attack strategies in non-fatal ways, and consequently mandate updates to recommended key sizes.

Quantum Random Oracle Model. The chief objective in provable security is to find better proof techniques allowing tighter bounds. In the case of the quantum random oracle model, the bounds are notoriously untight due to the pervasive square root. One question is whether these square roots are indeed a necessary feature of working in the quantum computing model, or whether there is a clever reduction that allows for their elimination. Even if they cannot be eliminated, however, they might be shifted to terms where they have less impact.

There remain classically-valid proof techniques that are invalid in the quantum random oracle model, and that have no obvious translation to the QROM. A major open question is therefore to determine which proof techniques can be saved, and which are inherently anti-quantum. It is conceivable that there be *various* QROM translations of the same principle, each with their pros and cons depending on the context. In the end, the holy grail remains a complete replacement of all quantum random oracles with concrete functions, along with a demonstration that this replacement does not degrade security.

Constructions and Transformations. The set of available proof techniques determines which constructions can be reduced to hard problems, and which functionalities can be constructed out of more basic primitives. For instance, to date there is only one post-quantum interactive-to-non-interactive transformation for zero-knowledge proofs of knowledge, namely that of Unruh [140]. This transformation applies only to commitment-based interactive protocols and even requires first transforming a non-commitment-based protocol into a commitment-based one before making it non-interactive. An alternative to the Unruh transform may come with significant bandwidth improvements for post-quantum signatures.

Another example is the hash-and-sign paradigm, which is presently provably secure only if the underlying trapdoor permutation has random reducibility. However, many post-quantum hash-and-sign signature schemes have been proposed and they seem secure despite the lack of any such proof. On a similar note, an alternative to finding an outright EUF-CMA proof for hash-and-sign protocols is to find a upgrade transform to obtain EUF-CMA secure signature schemes from UUF-CMA ones. However, such an upgrade is unavailable even in the classical random oracle model.

MQ. With respect to multivariate quadratic systems, they key question remains the quantification of their solving difficulty. While the complexity of solving random MQ systems is well understood, most MQ cryptosystems employ the bipolar construction to hide a trapdoor and in this case the resulting public key is far from random. For the specific case of HFE_v^- systems there are upper bounds on the first fall degree, which in turn is upper-bounded by the degree of semi-regularity [47, 48, 50]. However, these bounds are not tight and constrain in the wrong direction from the designer's point of view. For HFE_v^- systems in particular, and for bipolar constructions in general, provable security is a major open question.

Another issue related to provable security is the exact problem definition. The MQ problem is a search problem but there is a decision variant that is **NP**-complete. The hardness estimates apply to the search variant and while it is conceivable that the decision variant is equally hard, this search-decision equivalence remains conjectural. Additionally, the present formulation of the MQ problem is technically speaking not a non-interactive problem but an interactive game: it describes the process that generates the attacker's view rather than describing the instance itself. This is similar to the formulation of the SIS and LWE problems but in those cases at least there is a reduction from SVP_γ, which does have a non-interactive formulation. The average-case hardness of SIS and LWE follows from the *worst-case* hardness of SVP_γ. In the case of MQ, no such worst-to-average-case reduction is known and as a result, the average-case hardness of MQ must be assumed.

With respect to usability, a major downside of MQ systems is the large public key. This public key represents all the coefficients of a system of m quadratic equations in roughly as many variables, leading to a $O(m^3)$ scaling. In practice, public keys tend to be on the order of hundreds of kilobytes or even megabytes, which is far too large for resource constrained devices. A natural question therefore is whether, to what degree, and at what cost this public key can be shrunk.

Lastly, the absence of homomorphic properties makes the generation of public key cryptosystems with fancy properties a challenging task. Recent years have seen some progress for blind and ring signatures using the additivity of public keys [114, 102]. Beyond that, however, properties like homomorphic encryption or threshold signature generation remain unexplored.

Lattices. With respect to lattice-based cryptography, a major question remains determining the concrete quantum hardness of lattice basis reduction. The hardness argument of NewHope should at the very least be complemented with another argument in order to cover the complete parameter space and avoid the abrupt stoppage of Fig. 4.3. Another point of concern is that the quality of the bases output by BKZ is significantly better than the theoretical bounds [37].

Bibliography

[1] On the complexity of the F5 Gröbner basis algorithm. In *J. Symbolic Computation* (2015), pp. 49–70.

[2] Aaronson, S. *Quantum Computing Since Democritus, chapter 9: Quantum.* Cambridge University Press, 2013.

[3] Aggarwal, D., Joux, A., Prakash, A., and Santha, M. Mersenne-756839. Submission to the NIST PQC project.

[4] Aggarwal, D., Joux, A., Prakash, A., and Santha, M. A new public-key cryptosystem via Mersenne numbers. Cryptology ePrint Archive, Report 2017/481, 2017. `https://eprint.iacr.org/2017/481` — version of May 30 2017.

[5] Aggarwal, D., Joux, A., Prakash, A., and Santha, M. A new public-key cryptosystem via Mersenne numbers. Cryptology ePrint Archive, Report 2017/481, 2017. `https://eprint.iacr.org/2017/481` — version of December 6 2017.

[6] Aharonov, D., and Ben-Or, M. Fault-tolerant quantum computation with constant error. In *ACM STOC '97* (1997), F. T. Leighton and P. W. Shor, Eds., ACM, pp. 176–188.

[7] Ajtai, M. Generating hard instances of lattice problems (extended abstract). In *ACM STOC 1996* (1996), G. L. Miller, Ed., ACM, pp. 99–108.

[8] Ajtai, M. The shortest vector problem in L_2 is *NP*-hard for randomized reductions (extended abstract). In *ACM STOC 1998* (1998), J. S. Vitter, Ed., ACM, pp. 10–19.

[9] Albrecht, M. R., Cid, C., Faugère, J., and Perret, L. On the relation between the MXL family of algorithms and Gröbner basis algorithms. *J. Symb. Comput. 47*, 8 (2012), 926–941.

[10] Albrecht, M. R., Curtis, B. R., Deo, A., Davidson, A., Player, R., Postlethwaite, E. W., Virdia, F., and Wunderer, T. Estimate all the {LWE, NTRU} schemes! In *SCN 2018* (2018), D. Catalano and R. D. Prisco, Eds., vol. 11035 of *LNCS*, Springer, pp. 351–367.

[11] Albrecht, M. R., Player, R., and Scott, S. On the concrete hardness of learning with errors. *J. Mathematical Cryptology 9*, 3 (2015), 169–203.

[12] Alkim, E., Ducas, L., Pöppelmann, T., and Schwabe, P. Post-quantum key exchange - A new hope. In *USENIX Security 2016.* (2016), T. Holz and S. Savage, Eds., USENIX Association, pp. 327–343.

[13] Anand, M. V., Targhi, E. E., Tabia, G. N., and Unruh, D. Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In *PQCrypto 2016* (2016), T. Takagi, Ed., vol. 9606 of *LNCS*, Springer, pp. 44–63.

[14] Arora, S., and Ge, R. New algorithms for learning in presence of errors. In *ICALP 2011, Part I* (2011), L. Aceto, M. Henzinger, and J. Sgall, Eds., vol. 6755 of *LNCS*, Springer, pp. 403–415.

[15] Atmanspacher, H. Quantum approaches to consciousness, 2015. https://plato.stanford.edu/entries/qt-consciousness/.

[16] Bardet, M., Faugère, J.-C., and Salvy, B. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *ICPSS 2004* (2004).

[17] Bardet, M., Faugère, J.-C., Salvy, B., and Yang, B.-Y. Asymptotic behaviour of the index of regularity of quadratic semi-regular polynomial systems. In *MEGA 05* (2005), Citeseer, pp. 1–14.

[18] Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. A concrete security treatment of symmetric encryption. In *FOCS '97* (1997), IEEE Computer Society, pp. 394–403.

[19] Bellare, M., and Neven, G. Multi-signatures in the plain public-key model and a general forking lemma. In *ACM CCS 2006* (2006), A. Juels, R. N. Wright, and S. D. C. di Vimercati, Eds., ACM, pp. 390–399.

[20] Bellare, M., and Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS '93* (1993), D. E. Denning, R. Pyle, R. Ganesan, R. S. Sandhu, and V. Ashby, Eds., ACM, pp. 62–73.

[21] Bellare, M., and Rogaway, P. Optimal asymmetric encryption. In *EUROCRYPT '94* (1994), A. D. Santis, Ed., vol. 950 of *LNCS*, Springer, pp. 92–111.

[22] Bellare, M., and Rogaway, P. Code-based game-playing proofs and the security of triple encryption. *IACR Cryptology ePrint Archive 2004* (2004), 331.

[23] Bennett, C. H., and Brassard, G. An update on quantum cryptography. In *CRYPTO '84* (1984), G. R. Blakley and D. Chaum, Eds., vol. 196 of *LNCS*, Springer, pp. 475–480.

[24] Benoit Libert (coordinator). PROMETHEUS. http://prometheuscrypt.gforge.inria.fr/. accessed 2018-05-27.

[25] Bernstein, D. J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., and Wilcox-O'Hearn, Z. SPHINCS: practical stateless hash-based signatures. In *EUROCRYPT 2015 Part I* (2015), E. Oswald and M. Fischlin, Eds., vol. 9056 of *LNCS*, Springer, pp. 368–397.

[26] Beullens, W., Preneel, B., and Szepieniec, A. Public key compression for constrained linear signature schemes. *IACR Cryptology ePrint Archive 2018* (2018), 670. Also available in Part II, §. 6.3.

[27] Beunardeau, M., Connolly, A., Géraud, R., and Naccache, D. On the hardness of the mersenne low hamming ratio assumption. *IACR Cryptology ePrint Archive 2017* (2017), 522.

[28] Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., and Zhandry, M. Random oracles in a quantum world. In *ASIACRYPT 2011* (2011), D. H. Lee and X. Wang, Eds., vol. 7073 of *LNCS*, Springer, pp. 41–69.

[29] Bosma, W., Cannon, J., and Playoust, C. The Magma algebra system. I. The user language. *J. Symbolic Comput. 24*, 3-4 (1997), 235–265. Computational algebra and number theory (London, 1993).

[30] Bouillaguet, C., Fouque, P., and Véber, A. Graph-theoretic algorithms for the "isomorphism of polynomials" problem. In *EUROCRYPT 2013* (2013), T. Johansson and P. Q. Nguyen, Eds., vol. 7881 of *LNCS*, Springer, pp. 211–227.

[31] Buchberger, B. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, 1965.

[32] Calderbank, A. R., and Shor, P. W. Good quantum error-correcting codes exist. *Physical Review A 54*, 2 (1996), 1098.

[33] Castryck, W., Lange, T., Martindale, C., Panny, L., and Renes, J. CSIDH: an efficient post-quantum commutative group action. *IACR Cryptology ePrint Archive 2018* (2018), 383.

[34] Chen, C.-H. O., Yang, B.-Y., and Chen, J.-M. The limit of XL implemented with sparse matrices. In *PQCrypto 2006* (2006), pp. 215–225.

[35] Chen, M., Hülsing, A., Rijneveld, J., Samardjiska, S., and Schwabe, P. From 5-pass *MQ* -based identification to *MQ* -based signatures. In *ASIACRYPT 2016 Part II* (2016), J. H. Cheon and T. Takagi, Eds., vol. 10032 of *LNCS*, pp. 135–165.

[36] Chen, Y. *Reduction de reseau et securité concrète du chiffrement complètement homomorphe.* PhD thesis, 2013.

[37] Chen, Y., and Nguyen, P. Q. BKZ 2.0: Better lattice security estimates. In *ASIACRYPT 2011* (2011), D. H. Lee and X. Wang, Eds., vol. 7073 of *LNCS*, Springer, pp. 1–20.

[38] Chuang, I. L., Gershenfeld, N., and Kubinec, M. Experimental implementation of fast quantum searching. *Phys. Rev. Lett. 80* (Apr 1998), 3408–3411.

[39] Coppersmith, D. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Mathematics of Computation 62*, 205 (1994), 333–350.

[40] Coppersmith, D. Finding a small root of a bivariate integer equation; factoring with high bits known. In *EUROCRYPT '96* (1996), U. M. Maurer, Ed., vol. 1070 of *LNCS*, Springer, pp. 178–189.

[41] Coron, J. On the exact security of full domain hash. In *CRYPTO 2000* (2000), M. Bellare, Ed., vol. 1880 of *LNCS*, Springer, pp. 229–235.

[42] Courtois, N., Klimov, A., Patarin, J., and Shamir, A. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *EUROCRYPT 2000* (2000), B. Preneel, Ed., vol. 1807 of *LNCS*, Springer, pp. 392–407.

[43] Cox, D., Little, J., and O'Shea, D. *Ideals, Varieties, and Algorithms*, 2 ed. Springer.

[44] Deutsch, D. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A 400*, 1818 (1985), 97–117.

[45] Dieks, D. Communication by epr devices. *Physics Letters A 92*, 6 (1982), 271–272.

[46] Ding, J., Cabarcas, D., Schmidt, D., Buchmann, J., and Tohaneanu, S. Mutant gröbner basis algorithm. In *SCC 2008* (2008), pp. 23–32.

[47] Ding, J., and Hodges, T. J. Inverting HFE systems is quasi-polynomial for all fields. In *CRYPTO 2011* (2011), P. Rogaway, Ed., vol. 6841 of *LNCS*, Springer, pp. 724–742.

[48] Ding, J., and Kleinjung, T. Degree of regularity for HFE-. *IACR Cryptology ePrint Archive 2011* (2011), 570.

[49] Ding, J., and Schmidt, D. Rainbow, a new multivariable polynomial signature scheme. In *ACNS 2005* (2005), J. Ioannidis, A. D. Keromytis, and M. Yung, Eds., vol. 3531 of *LNCS*, pp. 164–175.

[50] Ding, J., and Yang, B. Degree of regularity for HFev and HFEv-. In *PQCrypto 2013* (2013), P. Gaborit, Ed., vol. 7932 of *LNCS*, Springer, pp. 52–66.

[51] Ding, J., Yang, B., Cheng, C., Chen, C. O., and Dubois, V. Breaking the symmetry: a way to resist the new differential attack. *IACR Cryptology ePrint Archive 2007* (2007), 366.

[52] Einstein, A., Born, M., Born, H., et al. Born-Einstein letters. M. Born, Ed., Walker, p. 158.

[53] Ekert, A. K. Quantum cryptography based on Bell's theorem. *Physical review letters 67*, 6 (1991), 661.

[54] Faugère, J.-C. A new efficient algorithm for computing Gröbner bases without reduction to zero (f_5). In *ISSAC 2002* (2002), ACM, pp. 75–83.

[55] Faugère, J.-C. A new efficient algorithm for computing Gröbner bases (F4). *Journal of pure and applied algebra 139*, 1-3 (1999), 61–88.

[56] Feo, L. D., Jao, D., and Plût, J. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Mathematical Cryptology 8*, 3 (2014), 209–247.

[57] Feynman, R. P. Simulating physics with computers. *International journal of theoretical physics 21*, 6-7 (1982), 467–488.

[58] Fiat, A., and Shamir, A. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO '86* (1986), A. M. Odlyzko, Ed., vol. 263 of *LNCS*, Springer, pp. 186–194.

[59] Fischlin, M. Communication-efficient non-interactive proofs of knowledge with online extractors. In *CRYPTO 2005* (2005), V. Shoup, Ed., vol. 3621 of *LNCS*, Springer, pp. 152–168.

[60] Fluhrer, S. R. Reassessing Grover's algorithm. *IACR Cryptology ePrint Archive 2017* (2017), 811.

[61] Fujisaki, E., and Okamoto, T. How to enhance the security of public-key encryption at minimum cost. In *PKC '99* (1999), H. Imai and Y. Zheng, Eds., vol. 1560 of *LNCS*, Springer, pp. 53–68.

[62] Gall, F. L. Powers of tensors and fast matrix multiplication. In *ISSAC '14* (2014), K. Nabeshima, K. Nagasaka, F. Winkler, and Á. Szántó, Eds., ACM, pp. 296–303.

[63] Gama, N., and Nguyen, P. Q. Predicting lattice reduction. In *EUROCRYPT 2008* (2008), N. P. Smart, Ed., vol. 4965 of *LNCS*, Springer, pp. 31–51.

[64] Grover, L. K. A fast quantum mechanical algorithm for database search. In *ACM STOC 1996* (1996), G. L. Miller, Ed., ACM, pp. 212–219.

[65] Hardy, L. Quantum theory from five reasonable axioms. *arXiv preprint quant-ph/0101012* (2001).

[66] Hoffstein, J., Pipher, J., and Silverman, J. H. NTRU: A ring-based public key cryptosystem. In *ANTS 1998* (1998), J. Buhler, Ed., vol. 1423 of *LNCS*, Springer, pp. 267–288.

[67] Hosoyamada, A., and Sasaki, Y. Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. *IACR Cryptology ePrint Archive 2017* (2017), 977.

[68] Hosoyamada, A., and Sasaki, Y. Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. In *CT-RSA 2018* (2018), N. P. Smart, Ed., vol. 10808 of *LNCS*, Springer, pp. 198–218.

[69] hugh Everett III. *The theory of the universal wave function.* PhD thesis, 1973.

[70] Hülsing, A., Rijneveld, J., and Song, F. Mitigating multi-target attacks in hash-based signatures. In *PKC 2016, Part I* (2016), C. Cheng, K. Chung, G. Persiano, and B. Yang, Eds., vol. 9614 of *LNCS*, Springer, pp. 387–416.

[71] IBM. IBM builds its most powerful universal quantum computing processors. `https://www-03.ibm.com/press/us/en/pressrelease/52403.wss`. accessed 2018-05-26.

[72] IBM. IBM raises the bar with a 50-qubit quantum computer. https://www.technologyreview.com/s/609451/ibm-raises-the-bar-with-a-50-qubit-quantum-computer/. accessed 2018-05-26.

[73] Ikematsu, Y., Perlner, R. A., Smith-Tone, D., Takagi, T., and Vates, J. HFERP - A new multivariate encryption scheme. In *PQCrypto 2018* (2018), T. Lange and R. Steinwandt, Eds., vol. 10786 of *LNCS*, Springer, pp. 396–416.

[74] Information Technology Laboratory, N. Digital Signature Standard (DSS). https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf. Federal Information Processing Standards (FIPS) 186-4.

[75] Information Technology Laboratory, N. Post-quantum cryptography. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography. accessed 2018-05-27.

[76] Intel. 2018 CES: Intel advances quantum and neuromorphic computing research. https://newsroom.intel.com/news/intel-advances-quantum-neuromorphic-computing-research/. accessed 2018-05-26.

[77] Intel. Intel delivers 17-qubit superconducting chip with advanced packaging to QuTech. https://newsroom.intel.com/news/intel-delivers-17-qubit-superconducting-chip-advanced-packaging-qutech/. accessed 2018-05-26.

[78] Kalai, G. Detrimental decoherence. *arXiv abs/0806.2443* (2008).

[79] Kalai, G. How quantum computers fail: Quantum codes, correlations in physical systems, and noise accumulation. *arXiv abs/1106.0485* (2011).

[80] Kalai, G. The quantum computer puzzle. *Notices of the AMS 63*, 5 (2016), 508–516.

[81] Kaplan, M., Leurent, G., Leverrier, A., and Naya-Plasencia, M. Breaking symmetric cryptosystems using quantum period finding. In *CRYPTO 2016, Part II* (2016), M. Robshaw and J. Katz, Eds., vol. 9815 of *LNCS*, Springer, pp. 207–237.

[82] Kelly, J., and Google. A preview of Bristlecone, Google's new quantum processor. https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html. accessed 2018-05-26.

[83] Kipnis, A., Patarin, J., and Goubin, L. Unbalanced oil and vinegar signature schemes. In *EUROCRYPT '99* (1999), J. Stern, Ed., vol. 1592 of *LNCS*, Springer, pp. 206–222.

[84] Kitaev, A. Y. Quantum error correction with imperfect gates. In *Quantum Communication, Computing, and Measurement*. Springer, 1997, pp. 181–188.

[85] Knill, E., Laflamme, R., and Zurek, W. H. Resilient quantum computation: error models and thresholds. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* (1998), vol. 454-1969, The Royal Society, pp. 365–384.

[86] Kuwakado, H., and Morii, M. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In *IEEE ISIT 2010* (2010), IEEE, pp. 2682–2685.

[87] Kuwakado, H., and Morii, M. Security on the quantum-type even-mansour cipher. In *ISITA 2012* (2012), IEEE, pp. 312–316.

[88] Laboratories, R. PKCS #1 v2. 2: RSA Cryptography Standard. https://www.emc.com/collateral/white-papers/h11300-pkcs-1v2-2-rsa-cryptography-standard-wp.pdf. accessed 2018-05-27.

[89] Lange, T., and Steinwandt, R., Eds. *PQCrypto 2018* (2018), vol. 10786 of *LNCS*, Springer.

[90] Lange, T., and Takagi, T., Eds. *PQCrypto 2017* (2017), vol. 10346 of *LNCS*, Springer.

[91] Langlois, A., and Stehlé, D. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography 75*, 3 (2015), 565–599.

[92] Lenstra, A. K., and Lenstra, H. W., Eds. *The development of the number field sieve.* LNCS. Springer, 1993.

[93] Lenstra, A. K., Lenstra, H. W., and Lovász, L. Factoring polynomials with rational coefficients. *Mathematische Annalen 261*, 4 (1982), 515–534.

[94] Lucero, E., Barends, R., Chen, Y., Kelly, J., Mariantoni, M., Megrant, A., O'Malley, P., Sank, D., Vainsencher, A., Wenner, J., et al. Computing prime factors with a josephson phase qubit quantum processor. *Nature Physics 8*, 10 (2012), 719.

[95] Lyubashevsky, V., and Micciancio, D. Generalized compact knapsacks are collision resistant. In *ICALP 2006, Part II* (2006), M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052 of *LNCS*, Springer, pp. 144–155.

[96] Lyubashevsky, V., Peikert, C., and Regev, O. On ideal lattices and learning with errors over rings. *J. ACM 60*, 6 (2013), 43:1–43:35.

[97] May, A., and Ozerov, I. On computing nearest neighbors with applications to decoding of binary linear codes. In *EUROCRYPT 2015 Part I* (2015), E. Oswald and M. Fischlin, Eds., vol. 9056 of *LNCS*, Springer, pp. 203–228.

[98] McEliece, R. J. A public-key cryptosystem based on algebraic. *Coding Thv 4244* (1978), 114–116.

[99] Mennink, B., and Szepieniec, A. XOR of PRPs in a quantum world. In *PQCrypto 2017* (2017), T. Lange and T. Takagi, Eds., vol. 10346 of *LNCS*, Springer, pp. 367–383.

[100] Mohamed, M. S. E., Cabarcas, D., Ding, J., Buchmann, J. A., and Bulygin, S. MXL3: An efficient algorithm for computing Gröbner bases of zero-dimensional ideals. In *ICISC 2009* (2009), D. H. Lee and S. Hong, Eds., vol. 5984 of *LNCS*, Springer, pp. 87–100.

[101] Mohamed, M. S. E., Mohamed, W. S. A. E., Ding, J., and Buchmann, J. A. MXL2: solving polynomial equations over GF(2) using an improved mutant strategy. In *PQCrypto 2008* (2008), J. A. Buchmann and J. Ding, Eds., vol. 5299 of *LNCS*, Springer, pp. 203–215.

[102] Mohamed, M. S. E., and Petzoldt, A. Ringrainbow - an efficient multivariate ring signature scheme. In *AFRICACRYPT 2017* (2017), M. Joye and A. Nitaj, Eds., vol. 10239 of *LNCS*, Springer, pp. 3–20.

[103] Mohamed, W. S. A., Ding, J., Kleinjung, T., Bulygin, S., and Buchmann, J. PWXL: A parallel Wiedemann-XL algorithm for solving polynomial equations over GF (2). In *Conference on Symbolic Computation and Cryptography* (2010), C. Cid and J. Faugère, Eds., pp. 89–100.

[104] Mosca, M. Cybersecurity in an era with quantum computers: will we be ready? Cryptology ePrint Archive, Report 2015/1075, 2015. `https://eprint.iacr.org/2015/1075`.

[105] Negrevergne, C., Mahesh, T. S., Ryan, C. A., Ditty, M., Cyr-Racine, F., Power, W., Boulant, N., Havel, T., Cory, D. G., and Laflamme, R. Benchmarking quantum control methods on a 12-qubit system. *Phys. Rev. Lett. 96* (May 2006), 170501.

[106] Network Working Group, I. Internet Key Exchange (IKEv2) Protocol. `http://www.ietf.org/rfc/rfc4306.txt`, 2005. IETF RFC 4306.

[107] Nguyen, P. Q., and Shparlinski, I. E. The insecurity of the elliptic curve digital signature algorithm with partially known nonces. *Des. Codes Cryptography 30*, 2 (2003), 201–217.

[108] Nielsen, M. A., and Chuang, I. L. *Quantum computation and quantum information.* Cambridge university press, 2010.

[109] Patarin, J. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In *EUROCRYPT '96* (1996), U. M. Maurer, Ed., vol. 1070 of *LNCS*, Springer, pp. 33–48.

[110] Patarin, J. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In *EUROCRYPT '96* (1996), U. M. Maurer, Ed., vol. 1070 of *LNCS*, Springer, pp. 33–48.

[111] Patarin, J., Goubin, L., and Courtois, N. Improved algorithms for isomorphisms of polynomials. In *EUROCRYPT '98* (1998), K. Nyberg, Ed., vol. 1403 of *LNCS*, Springer, pp. 184–200.

[112] Peikert, C., and Rosen, A. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC 2006* (2006), S. Halevi and T. Rabin, Eds., vol. 3876 of *LNCS*, Springer, pp. 145–166.

[113] Petzoldt, A. *Selecting and reducing key sizes for multivariate cryptography.* PhD thesis, Darmstadt University of Technology, Germany, 2013.

[114] Petzoldt, A., Szepieniec, A., and Mohamed, M. S. E. A practical multivariate blind signature scheme. In *FC 2017* (2017), A. Kiayias, Ed., vol. 10322 of *LNCS*, Springer, pp. 437–454. Also available in Part II, §. 6.2.

[115] Plût, J., Fouque, P., and Macario-Rat, G. Solving the "isomorphism of polynomials with two secrets" problem for all pairs of quadratic forms. *CoRR abs/1406.3163* (2014).

[116] Preskill, J. Sufficient condition on noise correlations for scalable quantum computing. *Quantum Information & Computation 13*, 3-4 (2013), 181–194.

[117] Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In *ACM STOC 2005* (2005), H. N. Gabow and R. Fagin, Eds., ACM, pp. 84–93.

[118] Rivest, R. L., Shamir, A., and Adleman, L. M. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM 21*, 2 (1978), 120–126.

[119] Roetteler, M., Naehrig, M., Svore, K. M., and Lauter, K. E. Quantum resource estimates for computing elliptic curve discrete logarithms. In *ASIACRYPT 2017, Part II* (2017), T. Takagi and T. Peyrin, Eds., vol. 10625 of *LNCS*, Springer, pp. 241–270.

[120] Santoli, T., and Schaffner, C. Using simon's algorithm to attack symmetric-key cryptographic primitives. *Quantum Information & Computation 17*, 1&2 (2017), 65–78.

[121] Schnorr, C. Lattice reduction by random sampling and birthday methods. In *STACS 2003* (2003), H. Alt and M. Habib, Eds., vol. 2607 of *LNCS*, Springer, pp. 145–156.

[122] Schnorr, C., and Euchner, M. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. In *FCT 1991* (1991), L. Budach, Ed., vol. 529 of *LNCS*, Springer, pp. 68–85.

[123] Shor, P. W. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS 1994* (1994), IEEE Computer Society, pp. 124–134.

[124] Shoup, V. A Proposal for an ISO Standard for Public Key Encryption. `http://shoup.net/iso/`.

[125] Shoup, V. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive 2004* (2004), 332.

[126] Simon, D. R. On the power of quantum computation. In *FOCS 1994* (1994), IEEE Computer Society, pp. 116–123.

[127] Statista. Worldwide revenue from the supercomputer market from 2015 to 2021 (in billion U.S. dollars). https://www.statista.com/statistics/568431/hpc-server-revenue-worldwide/. accessed: 2018-05-25.

[128] Steane, A. Multiple-particle interference and quantum error correction. *Proc. R. Soc. Lond. A 452*, 1954 (1996), 2551–2577.

[129] Stehlé, D., Steinfeld, R., Tanaka, K., and Xagawa, K. Efficient public key encryption based on ideal lattices. In *ASIACRYPT 2009* (2009), M. Matsui, Ed., vol. 5912 of *LNCS*, Springer, pp. 617–635.

[130] Szepieniec, A. Ramstake. Submission to the NIST PQC project. Also available in Part II, §. 8.1.

[131] Szepieniec, A., Abidin, A., and Preneel, B. A digital signature scheme from short solutions to nonlinear equations, 2018. Also available in Part II, §. 7.1.

[132] Szepieniec, A., Beullens, W., and Preneel, B. MQ signatures for PKI. In *PQCrypto 2017* (2017), T. Lange and T. Takagi, Eds., vol. 10346 of *LNCS*, Springer, pp. 224–240.

[133] Szepieniec, A., and Preneel, B. Short solutions to nonlinear systems of equations. In *NuTMiC 2017, Revised Selected Papers* (2017), J. Kaczorowski, J. Pieprzyk, and J. Pomykała, Eds., vol. 10737 of *LNCS*, Springer, pp. 71–90. Also available in Part II, §. 6.4.

[134] Szepieniec, A., Reyhanitabar, R., and Preneel, B. Key encapsulation from noisy key agreement in the quantum random oracle model, 2018. Also available in Part II, §. 7.2.

[135] Takagi, T., Ed. *PQCrypto 2016* (2016), vol. 9606 of *LNCS*, Springer.

[136] Tanja Lange (coordinator). PQCRYPTO. https://pqcrypto.eu.org/. accessed 2018-05-27.

[137] Thomae, E., and Wolf, C. Solving underdetermined systems of multivariate quadratic equations revisited. In *PKC 2012* (2012), M. Fischlin, J. A. Buchmann, and M. Manulis, Eds., vol. 7293 of *LNCS*, Springer, pp. 156–171.

[138] Unruh, D. Quantum proofs of knowledge. In *EUROCRYPT 2012* (2012), D. Pointcheval and T. Johansson, Eds., vol. 7237 of *LNCS*, Springer, pp. 135–152.

[139] Unruh, D. Revocable quantum timed-release encryption. In *EUROCRYPT 2014* (2014), P. Q. Nguyen and E. Oswald, Eds., vol. 8441 of *LNCS*, Springer, pp. 129–146.

[140] Unruh, D. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT 2015 Part II* (2015), E. Oswald and M. Fischlin, Eds., vol. 9057 of *LNCS*, Springer, pp. 755–784.

[141] Vandersypen, L. M., Steffen, M., Breyta, G., constantino S Yannoni, Sherwood, M. H., and Chuang, I. L. Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature 414*, 6866 (2001), 883.

[142] Wang, Y., Ikematsu, Y., Duong, D. H., and Takagi, T. Efficient decryption algorithms for extension field cancellation type encryption schemes. In *ACISP* (2018), W. Susilo and G. Yang, Eds., vol. 10946 of *LNCS*, Springer, pp. 487–501.

[143] Wecker, D. Achieving practical quantum computing. https://www.youtube.com/watch?v=msOAS67LrPs&t=0s&list=PLPf_zcX3mNAwVYO_bohkVIIQ8d2mssyL4&index=10, 40:30, 2018. Invited Presentation at PQCrypto 2018.

[144] Wiedemann, D. H. Solving sparse linear equations over finite fields. *IEEE transactions on information theory 32*, 1 (1986), 54–62.

[145] Wikipedia. Hilbert series and Hilbert polynomial. https://en.wikipedia.org/wiki/Hilbert_series_and_Hilbert_polynomial. retrieved 2018-07-18.

[146] WOOA0923 (INTERNET ALIAS). Degree vs index of regularity. `https://crypto.stackexchange.com/questions/60375/degree-vs-index-of-regularity/60459#60459`. retrieved 2018-07-18.

[147] WOOTTERS, W. K., AND ZUREK, W. H. A single quantum cannot be cloned. *Nature 299*, 5886 (1982), 802–803.

[148] YANG, B., AND CHEN, J. All in the XL family: Theory and practice. In *ICISC 2004* (2004), C. Park and S. Chee, Eds., vol. 3506, Springer, pp. 67–86.

[149] YAO, X.-C., WANG, T.-X., CHEN, H.-Z., GAO, W.-B., FOWLER, A. G., RAUSSENDORF, R., CHEN, Z.-B., LIU, N.-L., LU, C.-Y., DENG, Y.-J., ET AL. Experimental demonstration of topological error correction. *Nature 482*, 7386 (2012), 489.

[150] YASUDA, T., AND SAKURAI, K. A multivariate encryption scheme with rainbow. In *ICICS 2015* (2015), S. Qing, E. Okamoto, K. Kim, and D. Liu, Eds., vol. 9543 of *LNCS*, Springer, pp. 236–251.

[151] ZHANDRY, M. How to construct quantum random functions. In *FOCS 2012* (2012), IEEE Computer Society, pp. 679–687.

[152] ZHANDRY, M. Secure identity-based encryption in the quantum random oracle model. In *CRYPTO 2012* (2012), R. Safavi-Naini and R. Canetti, Eds., vol. 7417 of *LNCS*, Springer, pp. 758–775.

[153] ZHAO, Z., CHEN, Y.-A., ZHANG, A.-N., YANG, T., BRIEGEL, H. J., AND PAN, J.-W. Experimental demonstration of five-photon entanglement and open-destination teleportation. *Nature 430*, 6995 (2004), 54.

Chapter 6

Published Papers

6.1 Extension Field Cancellation: A New Central Trapdoor for Multivariate Quadratic Systems

Publication data

Alan Szepieniec and Jintai Ding and Bart Preneel. “Extension Field Cancellation: A New Central Trapdoor for Multivariate Quadratic Systems” *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings*, pp. 182-196, 2016.

Contributions

Principal author.

Notes

This paper is the product of a collaboration with prof. Jintai Ding, whom I visited in August 2015. The key idea behind the construction came from me, but prof. Ding quickly observed that it was similar enough to HFE_v^- form for the HFE_v^- bounds to apply. Consequently, he suggested to omit equations as a countermeasure against attacks.

In hindsight, the present argument for security is rather weak. The HFE_v^- bounds are only upper bounds on the first fall degree, and are known not to be tight. Moreover, the bounds lose more tightness as there are twice as many equations as in the HFE_v^- case. And 80 bits of security is a rather low target.

Ludovic Perret has privately informed me that in his direct algebraic attack using the F_4 algorithm, the step degree never exceeds 4. This is terrible news because that would mean that the 80 bits is off by an order of magnitude. Nevertheless, the apparently bounded step degree might be merely an artifact of the proposed parameters rather than the construction as a whole. Additionally, in combination with the Rainbow-Plus technique [150, 73], it is likely that the number a of dropped polynomials can be decreased, thus allowing a larger field and hence a smaller public key while improving security. Also, it is worth noting that this paper is the subject of follow-up work by other authors [142], who target a higher security level. Moreover, this follow-up work establishes that the degree of regularity does rise with a, thus validating the soundness of the design.

Extension Field Cancellation: a New Central Trapdoor for Multivariate Quadratic Systems

Alan Szepieniec[1,2], Jintai Ding[3] and Bart Preneel[1,2]

[1] Department of Electrical Engineering,
ESAT/COSIC, KU Leuven, Belgium.
[2] iMinds, Belgium.
[3] University of Cincinnati, OH, USA.

Abstract. This paper introduces a new central trapdoor for multivariate quadratic (MQ) public-key cryptosystems that allows for encryption, in contrast to time-tested MQ primitives such as Unbalanced Oil and Vinegar or Hidden Field Equations which only allow for signatures. Our construction is a mixed-field scheme that exploits the commutativity of the extension field to dramatically reduce the complexity of the extension field polynomial implicitly present in the public key. However, this reduction can only be performed by the user who knows concise descriptions of two simple polynomials, which constitute the private key. After applying this transformation, the plaintext can be recovered by solving a linear system. We use the minus and projection modifiers to inoculate our scheme against known attacks. A straightforward C++ implementation confirms the efficient operation of the public key algorithms.

Keywords: MQ, multivariate, quadratic, public-key, post-quantum, encryption, mixed-field, trapdoor

1 Introduction

Since the inception of public-key cryptography, cryptographers have made a huge effort to find new and better computational problems that feature the elusive *trapdoor* — a small piece of information that can turn an otherwise hard to invert function into one that can easily be inverted. This on-going search effort has lead to a tremendous diversification of the computational problems that underpin public-key cryptography. This diversification is a good thing: by keeping all the eggs in separate baskets, a breakthrough in one area is unlikely to spill over to other areas, thus limiting the catastrophic potential of scientific advances.

Of particular interest to this paper is the class of problems known as multivariate quadratic (MQ) systems of equations. Not only do cryptosystems based on this primitive offer performance advantages over well-established ones such as RSA or systems based on elliptic curves, MQ cryptography is also conjectured to be post-quantum — that is to say, it holds promise of resisting attacks on quantum computers. From this point of view, MQ cryptography is certainly a promising line of research.

The key challenge in the design of MQ cryptosystems is to find a suitable central mapping $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ which should be easily invertible in addition to being expressible in terms of multivariate quadratic polynomials. The trapdoor information cannot be recovered efficiently from the public key as it is hidden by two affine transformations. Many central mappings have been proposed, most of which fall in two main categories [31]: single field schemes, such as UOV [16], Rainbow [7] and the triangular variants [30], where the central polynomial system is chosen to have a particular structure that enables efficient inversion; and mixed field schemes, such as C* [18], HFE [21] and Multi-HFE [3], where arithmetic in the base field is mixed with arithmetic in an extension field. However, despite the abundance of proposals, MQ cryptography has an awful track record as most of these proposals have been broken [2, 14, 17, 27, 28, 31].

Consequently, much research in the area of MQ cryptography has been devoted to patchwork — finding small modifications to existing systems that render specific attacks infeasible. A few examples among many that fall into this category are the minus modifier ("−") [24], which inoculates HFE-type systems against Gröbner basis attacks and linearization attacks; vinegar variables ("v") [16], which combines elements from different trapdoors and like "minus" is capable of making a Gröbner basis attack prohibitively expensive; and projection ("p") [9] which appears to successfully thwart the Dubois *et al.* differential attack [10, 11] on SFLASH.

However, the search for modifications to fix broken systems has an equally bad track record. Many of the MQ systems that were supposedly inoculated against some attack by the introduction of a modification, were broken by minor variants of that same attack. For example, both the multivariate generalization and the odd field characteristic variant of HFE were introduced and designed specifically to thwart the algebraic attack on HFE [14]; however, neither variant has managed to withstand cryptanalysis [2]. Another example is given by the fate of SFLASH, one of the three recommended signature schemes of the NESSIE project [1]. The addition of the minus modifier to the basic C* construction did not save the scheme from a new type of differential attack [10, 11]. The rapid spawn of attacks that break the inoculated systems seems to suggest the need for a more prudent design strategy: searching for fundamentally different basic principles for MQ trapdoors, rather than tinkering on the edges of existing ones.

Related work. Encryption schemes have been the bane of multivariate quadratic cryptography. No MQ encryption scheme has withstood the test of time, while several MQ *signature* schemes have. However, some very recent results and proposals in this area pose new and interesting challenges for cryptanalysts.

Porras *et al.* proposed a new central trapdoor which they call ZHFE [23]. Up until this point, the extension field polynomial in HFE-based cryptosystem required the number of nonzero coefficients to be small and its degree to be relatively low, so as to allow efficient root calculation. The idea of Porras *et al.* exchanges this single low-degree polynomial for a pair of high-degree polynomials that make up the central map. Additionally, these polynomials are chosen such that there exists a third polynomial, $\Psi(\mathcal{X})$, which is a function of the first two and yet has low degree. In order to invert a given image, it suffices to factorize this third polynomial. As the degree of the polynomials increases, so does the degree of regularity of the system. This increase in the degree of regularity, in turn, renders a direct algebraic attack infeasible, even though the very same attack broke the regular HFE cryptosystem.

Tao *et al.* proposed a multivariate quadratic encryption scheme called Simple Matrix Encryption, or simply ABC Encryption [26]. Their construction is based on a fundamentally new idea: embedding polynomial matrix arithmetic inside the central trapdoor function. The trapdoor can be inverted with high probability because the matrix, albeit evaluated in a single point, can be reconstructed from the output. With high probability this matrix can be inverted, giving rise to a system of linear equations which describe the input.

Our contributions. We introduce a new central trapdoor for multivariate quadratic encryption schemes. Our proposal is a mixed-field scheme — similar to the C* and HFE string of proposals because we use an embedding function to pretend as though a vector of variables in the base field were actually a single variable in the extension field. However, our proposal is notably different from its predecessors, where the restriction on the degree of this embedded polynomial was key both to their efficiency and to their demise; our proposal allows for a high-degree embedded polynomial and undoes this complexity by exploiting the commutative property of the extension field. Our proposal allows for encryption, in stark contrast to most other members of the HFE family.

Like the ABC Encryption Scheme, decryption of a ciphertext consists of essentially solving linear systems. This linear system is parameterized by the particular ciphertext or message: every possible ciphertext or message implicitly defines a unique linear system. Knowledge of the private key allows the user to obtain the linear system efficiently, while the adversary who attacks the system without this crucial information has no advantage to solve the quadratic system.

Like ZHFE, the central map consists of two high-degree extension field polynomials that satisfy a special relation which is obviously hidden from the adversary. The decryption algorithm exploits this relation to turn the otherwise hard inversion problem into an easy one.

Another important similarity between our map and both ABC and ZHFE is that all three are expanding maps, *i.e.*, $\mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ where $m = 2n$. This commonality is no accident, because in order allow unique decryption, the map must be injective. However, if $m \approx n$, the differential of this nearly-bijective map is readily differentiable from that of a random one — not a desirable property for multivariate quadratic maps to have.

Despite these similarities, the main advantage of our scheme is that its construction is notably *different* from ABC and ZHFE. Consequently, as-yet undiscovered weaknesses or even attacks that affect ABC or ZHFE may leave our scheme intact. Furthermore, this diversification opens the door for a combination of strategies whose end result reaps the benefits of both worlds. Certainly the case of HFEv proves that such a combination may indeed increase both security and performance.

In line with a common theme throughout MQ cryptography, we are unable to prove the security of our scheme or even to reduce it to a plausible computational assumption. An exhaustive list of all known attacks on MQ systems and why they fail against our system is beyond the scope of this paper. Nevertheless, we identify several pertinent attacks that may be launched against a naïve implementation of our scheme, and we propose strategies to thwart them. Patarin's linearization attack [20] is foiled by the minus modifier and repeated applications of the same modifier make the extended MinRank attack [4, 17] as well as the direct algebraic attack [14] prohibitively inefficient. The scheme seems naturally resistant to Dubois *et al.*'s differential attack [10,11], but we nevertheless recommend to use the projection modifier, which is the proper countermeasure against this attack.

Outline. We introduce notation and recall basic properties of MQ systems as well as of extension field embeddings in Section 2. Next, Section 3 defines the trapdoor proposed in this paper as well as several necessary modifiers. We recommend parameters for 80 bits of security in the first part of Section 4 and afterwards discuss the efficiency of our scheme, both from a theoretical point of view and by referencing timing results from a software implementation. Section 5 concludes the text.

2 Preliminaries

2.1 Notation and Definitions

We use small case letters (s) to denote scalars in the base field; extension field elements are denoted by calligraphic capital letters ($\mathcal{C}$); small case bold letters ($\mathbf{v}$) denote column vectors; and regular capital letters are used for matrices (M).

Let $\mathbb{F}_q$ denote the finite field with q elements, which we call the *base field.* With any combination of a finite field $\mathbb{F}_q$ with a polynomial $f(x) \in \mathbb{F}_q[x]$ one can associate a finite ring $\mathbb{E} = \mathbb{F}_q[x]/\langle f(x)\rangle$ of residue classes after division by $f(x)$. If f is irreducible over $\mathbb{F}_q$ and has degree n, then $\mathbb{E} = \mathbb{F}_{q^n}$ is a finite field we call the *extension field.* There exists a natural homomorphism $\varphi : (\mathbb{F}_q)^n \to \mathbb{F}_{q^n}$ that maps a vector $\mathbf{v} = (v_1, \dots, v_n)^\mathsf{T} \in \mathbb{F}_q^n$ onto an element $\mathcal{V} \in \mathbb{F}_{q^n}$ of the extension field. We can apply this embedding function to the vector of indeterminates $\mathbf{x}$ in order to get the extension field indeterminate $\mathcal{X} = \varphi(\mathbf{x})$.

2.2 Multivariate Quadratic Systems

The public key of an MQ cryptosystem is a system of quadratic polynomials mapping n input variables to m output variables: $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$; the public operation consists of evaluating this system of polynomials in a point. The secret key consists of a pair of invertible affine mappings on the input and output variables, S and T, and an alternate quadratic system of polynomials, $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, such that $\mathcal{P} = T \circ \mathcal{F} \circ S$. The affine transformations are trivially inverted; the central system $\mathcal{F}$ is constructed in such a way that it is also easy to invert. However, the attacker cannot efficiently recover $\mathcal{F}$ from $\mathcal{P}$ and calculate the inverse as $\mathcal{F}$ is hidden by the affine transformations. A schematic overview is given in Fig. 1.

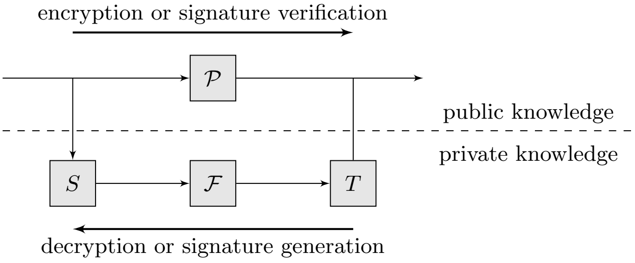

encryption or signature verification
$\mathcal{P}$
public knowledge
private knowledge
S
$\mathcal{F}$
T
decryption or signature generation

Fig. 1: Schematic representation of multivariate quadratic cryptosystems.

Given a central trapdoor $\mathcal{F}$ it is easy to construct a multivariate quadratic cryptosystem by composing it with two affine transformations. This process is out of the scope of the present paper. Rather, we restrict our attention to the construction of the central trapdoors.

3 Central Map

3.1 The Basic Construction

Let $A \in \mathbb{F}_q^{n\times n}$ be a random matrix over the base field. Then $A\mathbf{x} \in (\mathbb{F}_q[\mathbf{x}])^n$ represents a vector where each element is a linear polynomial in $\mathbf{x}$. And then $\alpha(\mathbf{x}) = \varphi(A\mathbf{x})$ is an extension field element. The square matrix that represents multiplication by $\alpha(\mathbf{x})$ is denoted by $\alpha_m(\mathbf{x}) \in \mathbb{F}_q^{n\times n}$. We use $\alpha(\mathcal{X})$ to stress the fact that α may also be considered as a univariate polynomial in $\mathcal{X}$ over the extension field, regardless of its representation, although the degree of this polynomial is larger than one.

Similarly, let $\beta(\mathbf{x}) = \varphi(B\mathbf{x})$ for a random $n\times n$ matrix $B \in \mathbb{F}_q^{n\times n}$. With these polynomials α and β, we define the central trapdoor as follows:

$$\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^{2n} : \mathbf{x} \mapsto \begin{pmatrix} \alpha_m(\mathbf{x})\mathbf{x} \\ \beta_m(\mathbf{x})\mathbf{x} \end{pmatrix} \ . \tag{1}$$

To see how we are able to invert $\mathcal{F}(\mathbf{x}) = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix}$, consider first the equality $\alpha(\mathbf{x})\beta(\mathbf{x}) = \beta(\mathbf{x})\alpha(\mathbf{x})$ which holds due to the commutativity of the extension field. We can proceed to construct a system of linear equations in $\mathbf{x}$:

$$\beta_m(\mathbf{x})\mathbf{d}_1 - \alpha_m(\mathbf{x})\mathbf{d}_2 = 0 \ . \tag{2}$$

While Gaussian elimination is in this case guaranteed to find a solution, this solution need not be unique. Nevertheless, this set of solutions is expected to be small, in accordance with the number of solutions to random linear systems. Moreover, this set can be pruned by iteratively plugging the potential solution into the function $\mathcal{F}$ and verifying that the correct output image $(\mathbf{d}_1; \mathbf{d}_2)$ is produced.

3.2 Modifiers

The trapdoor as described above is insecure. In particular, it is broken by the bilinear attack, the MinRank attack, as well as an algebraic attack using fast Gröbner basis algorithms. We apply the "minus" to inoculate basic EFC against these attacks. While not strictly necessary, "projection" may guard against new differential attacks at very little cost whereas "Frobenius tail" drastically drops the cost of decryption.

Minus.

Although Patarin's linearization attack [20] was originally conceived to attack C*, it also applies to unprotected EFC. Indeed, Equation 2 describes a bilinear polynomial in the plaintext and ciphertext, whose coefficients can be calculated using linear algebra after obtaining enough plaintext-ciphertext pairs. Once these coefficients are known, obtaining a plaintext that matches a given ciphertext is easy. However, dropping just one polynomial from the public key is enough to foil this attack. In this case, the attacker must guess the missing information for every plaintext-ciphertext pair, making them useless for exact linear algebra.

This "minus" modifier, which consists of removing one or more polynomials from the public key [22], is more than just a countermeasure against Patarin's attack. A pair of important results by Ding *et al.* [6, 8] indicates that this modifier is much better thought of as a fundamental building block of multivariate quadratic cryptosystems rather than a mere patch. Indeed, not only does the first application of this modifier block Patarin's linearization attack; every repeated application increments by one the rank of the quadratic form associated with the extension field polynomial, rendering the MinRank attack due to Kipnis and Shamir [17] as well as its subsequent improvement by Courtois [4] that much more infeasible. Furthermore, this rank increase in turn increases the degree of regularity of the system, resulting in a similarly infeasible algebraic attack.

The use of this modifier does come at the cost of a performance penalty. In particular, the decryption algorithm must first guess the values of the missing polynomials before undoing the output transformation T. Under this guess, it can proceed to the linear system in Equation 2 and compute the potential matching plaintext $\mathbf{x}$. If indeed $\mathcal{F}(\mathbf{x}) = (\mathbf{d}_1; \mathbf{d}_2)$, then the correct plaintext was found. If not, then the guess was wrong and the algorithm must start all over again with a new one.

Fortunately, as long as the number of dropped polynomials a is small enough, the correct plaintext will still be found with overwhelming probability. In order for the decryption algorithm to produce the wrong plaintext $\mathbf{x}$ upon decrypting the ciphertext $\mathbf{y}$, there must exist at least two guesses $\mathbf{g}_1 \in \mathbb{F}_q^a$ and $\mathbf{g}_2 \in \mathbb{F}_q^a$ such that both $(\mathbf{y}; \mathbf{g}_1)$ and $(\mathbf{y}; \mathbf{g}_2)$ are in the range of $\mathcal{P}$. If $\mathcal{P}$ is to be modeled as a random function $\mathbb{F}_q^n \to \mathbb{F}_q^{2n-a}$, then its range is a uniform subset of $\mathbb{F}_q^{2n-a}$ of size q^n, and then the probability of this event is approximately $q^n \times q^{-2n+a} = q^{-n+a}$. Consequently, as long as $a \ll n$, the probability of decryption error remains astronomically small.

Fig. 2 offers empirical validation of this argument. It shows the probability of decryption error for various even values for a as a function of n. Only when a and n are on the same order of magnitude, is this probability noticeable; when n rises to practical values, this probability does indeed drop to zero.

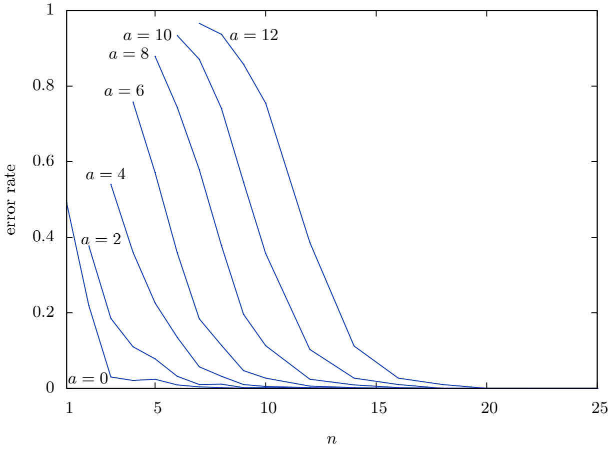


Fig. 2: Observed decryption error rate.

In similar fashion to C^{*-} and HFE^{-}, this modifier will be denoted by the superscript "$-$", *i.e.*, EFC^{-}. The number of dropped polynomials will be denoted by a.

Projection.

The differential symmetry attacks by Dubois *et al.* [10, 11] on SFLASH, a C* variant, show that the minus operator is not enough to secure it. Dubois *et al.* identify a symmetry in the differential of the C* map $\mathcal{F}$:

$$D\mathcal{F}(L\mathbf{x}, \mathbf{y}) + D\mathcal{F}(\mathbf{x}, L\mathbf{y}) = \Lambda\mathcal{F}(\mathbf{x}, \mathbf{y})$$

for some matrices L and Λ. The presence of this symmetry proved fatal.

Fortunately, Ding *et al.* [9] show experimentally that a small tweak by the name of "projection" completely foils this line of attack. In particular, pSFLASH projects the input vector $\mathbf{x}$ onto a lower-dimensional space before passing it through the central map. Smith-Tone [25] has since offered a theoretical basis for the efficacy of this modifier. At the core of Smith-Tone's argument is the following theorem:

Theorem 1 (Smith-Tone, [25]). *A polynomial $f : \mathbb{F}_{q^n} \rightarrow \mathbb{F}_{q^n}$ with a bilinear differential has the multiplicative symmetry if and only if it has one quadratic monomial summand.*

While the components of EFC do have bilinear differentials, they do not consist of a single quadratic monomial but of a sum of them. For example, the first component is described by $\alpha(\mathcal{X})\mathcal{X} = \sum_{i=0}^{n-1} \mathcal{A}_i \mathcal{X}^{q^i+1}$ where the coefficients $\mathcal{A}_i$ are with overwhelming probability not all but one equal to zero. Therefore, by Smith-Tone's theorem, the differential multiplicative symmetry is absent with overwhelming probability.

Nevertheless, in anticipation of more general attacks using a similar differential invariant, we follow a perspective offered at the conclusion Smith-Tone's paper: *projection does not destroy the differential symmetry, but pushes it down to a subfield.* Since this modifier is cheap in terms of performance and cannot degrade security, we choose to err on the side of safety and ensure that no such subfield can exist. In particular, we guarantee that the matrices A and B have rank $n-1$, and that n is a prime number. Moreover, the kernels of A and B do not intersect except at the origin. This modifier will be denoted by the subscript p, *e.g.* EFC$_p$.

Frobenius Tail in Characteristic Two (or Three).

The trapdoor as described so far can be implemented over any base field and unless the minus operator is applied, the rank of the quadratic forms associated with the extension field is two. However, if we restrict to characteristic two, we can naturally increase this rank by adding an extra "tail" term to both expressions. In turn, we must drop fewer equations to ensure the same level of security, and this results in a significant speedup of the decryption algorithm. We will use the subscript t^2 to denote the use of this technique, *e.g.* EFC$_{t^2}$.

This trick exploits the following property of fields of characteristic two. Let $f(\mathcal{X})$ be a linear function, then $f(\mathcal{X})^3$ is a quadratic function and multiplication by $f(\mathcal{X})$ gives $f(\mathcal{X})^4$ which is once again a linear function.

Let α and β be defined as earlier. Then this enhancement adds the quadratic terms $\alpha(\mathcal{X})^3$ and $\beta(\mathcal{X})^3$ as follows:

$$\mathcal{F} : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}^2 : \mathcal{X} \mapsto \begin{pmatrix} \alpha(\mathcal{X})\mathcal{X} + \beta(\mathcal{X})^3 \\ \beta(\mathcal{X})\mathcal{X} + \alpha(\mathcal{X})^3 \end{pmatrix} \ . \tag{3}$$

In order to decrypt $\mathcal{F}(\mathcal{X}) = (\mathcal{D}_1; \mathcal{D}_2)$, the user solves the linear system

$$\alpha(\mathcal{X})\mathcal{D}_2 - \beta(\mathcal{X})\mathcal{D}_1 = \alpha(\mathcal{X})^4 - \beta(\mathcal{X})^4 \ . \tag{4}$$

Afterwards, the set of solutions is pruned based on $\mathcal{F}(\mathcal{X}) = (\mathcal{D}_1; \mathcal{D}_2)$.

A similar trick is possible in fields of characteristic three. For linear functions $f(\mathcal{X})$ the term $f(\mathcal{X})^2$ is quadratic and multiplication by $f(\mathcal{X})$ gives $f(\mathcal{X})^3$ which is once again a linear function. Although this particular Frobenius tail does destroy the common factor in the two polynomials, it merely increases the rank of the quadratic form to three. The use of this trick will be denoted by the subscript t^3.

4 Efficiency

4.1 Recommended Parameters

We predict that the most efficient attack on our system is the algebraic attack using efficient Gröbner basis algorithms such as Faugère's F_4 or F_5 [12,13]. Taking this attack into account, we propose parameters to ensure at least 80 bits of security.

We follow the argument due to Ding *et al.* [5,8], who develop an upper bound for the degree of regularity of HFE$^-$ systems. In this line of reasoning, the degree of regularity D_{reg} is intricately linked to the rank r of the quadratic form associated with the extension field polynomial. Moreover, a applications of the minus modifier effectively increases this rank by a. Especially for small base fields, the degree of regularity is expected to lie near its upper bound:

$$D_{\text{reg}} \leq \frac{(q-1)(r+a)}{2} + 2 \ . \tag{5}$$

This argument applies to a single quadratic form. However, the central map of EFC consists of *two* quadratic forms. Nevertheless, we argue that the effect of minus is replicated across both quadratic forms. The polynomials are dropped *after* the output transformation T is applied, meaning that the effect of the missing information passes through T^{-1} and is not isolated to one quadratic form but spread across both. Although this reasoning underscores the following parameter recommendations, we note it is not perfectly rigorous and warrants further study.

Considering the two components of our central map separately, we see that their rank is $r = 2$. If the Frobenius tail modifiers are applied, this is increased to $r = 4$ and $r = 3$ for characteristics 2 and 3, respectively. For a security level of 80 bits, we recommend to ensure this adjusted rank is at least 12 for $\mathbb{F}_2$ and 8 for $\mathbb{F}_3$.

$$a = \begin{cases} 10 & q = 2,\, n = 83,\, \text{EFC}_p^- \\ 8 & q = 2,\, n = 83,\, \text{EFC}_{pt^2}^- \\ 6 & q = 3,\, n = 59,\, \text{EFC}_p^- \end{cases} . \tag{6}$$

Then we can estimate the degrees of regularity for these base fields:

$$D_{\text{reg}} \leq \frac{(q-1)(r+a)}{2} + 2 = \begin{cases} 8 & q = 2 \\ 10 & q = 3 \end{cases} . \tag{7}$$

The running time of efficient Gröbner basis algorithms is dominated by Gaussian elimination in the matrix of coefficients associated with the monomials of degree D_{reg}. We can use this bottleneck to estimate the algorithm's total complexity. In particular, the number of monomials of this degree is given by $T = \binom{n}{D_{\text{reg}}} \approx 2^{35}$ both for $n = 83, q = 2$ as well as $n = 59, q = 3$. Moreover, the number of nonzero monomials is on the order of $\tau = \binom{n}{2} \geq 2^{10}$. Assuming a Wiedemann-type algorithm [29] for sparse Gaussian elimination, this amounts to $\tau T^2 \geq 2^{80}$ in both cases.

Fig. 3 offers some experimental evidence in support of this argument. It plots the running time of MAGMA's F_4 algorithm to recover the plaintext from the ciphertext and the public key. The graph on the left starts out with $q = 2, n = 35$ and $a = 1$; from there on out, the parameter a increases. The graph on the right lets n vary from 15 to 38 with $q = 2$, and keeps a constant at 10 for the basic trapdoor EFC_p^- (blue circles) and at 8 for the Frobenius tail equivalent $\mathrm{EFC}_{pt^2}^-$ (red crosses).

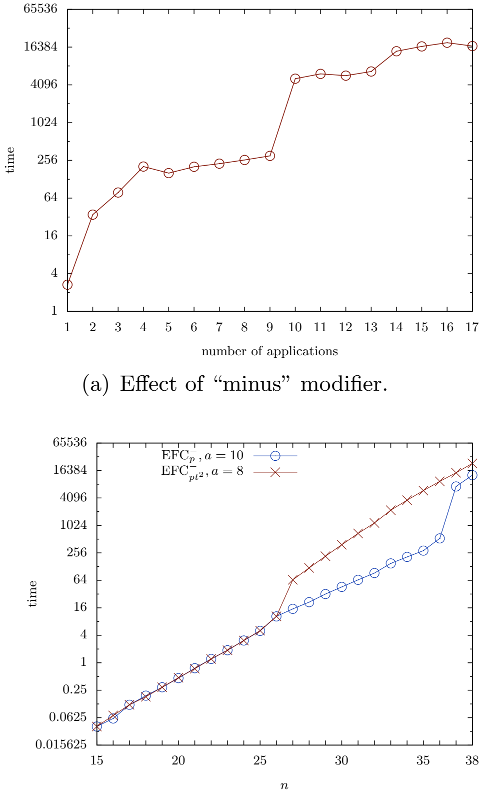
The graphs indicate two things. First, the minus modifier enhances security with (nearly) every application, occasionally lifting the system into the next degree of regularity. Second, the Frobenius tail modifier enhances security, even compensating for the rank drop associated with going from $a = 10$ to $a = 8$.

4.2 Complexity

The basic trapdoor, as well as all the modified variants, feature only quadratic terms. Therefore, the transformations T and S should be linear and not affine, and consequently also the public key will consist of only quadratic terms.

The public key consists of $2n - a$ polynomials of degree 2 in n variables. Thus the number of coefficients from $\mathbb{F}_q$ in the public key is $(2n - a) \times \frac{n(n-1)}{2} = n^3 - (a + 1)n^2 + an = O(n^3)$ because $a \ll n$. However, we note that there is a considerable amount of redundancy in the public key which we expect can be exploited to produce smaller keys.

The private key consists of two linear transformations S and T, along with a degree-n irreducible polynomial $\psi(z)$, and matrices A



(a) Effect of “minus” modifier.

(b) Effect of parameter n.

Fig. 3: Running time of algebraic attack for various parameters.

and B. This amounts to $n^2 + (2n)^2 + 2(n^2) + n = 7n^2 + n = O(n^2)$ coefficients in $\mathbb{F}_q$.

The most computationally intensive part of the key generation algorithm is the symbolic matrix-vector multiplication — once in $\varphi(A\mathbf{x})\mathbf{x}$ and once in $\varphi(B\mathbf{x})\mathbf{x}$. Both procedures require n^2 polynomial-multiplications, each of which consists of n multiplications in $\mathbb{F}_q$. Since the other steps in the key generation algorithm are less complex, the asymptotic time complexity of this entire algorithm is $O(n^3)$. For the Frobenius tail modifier, this complexity is worse because the additional extension field products $\varphi(A\mathbf{x})(QA\mathbf{x})$ and

$\varphi(B\mathbf{x})(QB\mathbf{x})$ (where Q is the matrix associated with the Frobenius map $x \mapsto x^2$) have dense right-side multiplicands. Consequently, the cost of polynomial multiplication rises to n^2 multiplications and the total time complexity of the key generation to $O(n^4)$.

Encryption consists of evaluating $2n - a$ quadratic polynomials in n variables. This comes down to two time steps with unlimited parallelism. Without parallelism, however, each of the $(2n - a) \times (n(n-1) + 2n)$ base field operations must be executed sequentially and the time complexity is therefore $O(n^3)$.

Decryption consists of the following steps for q^a different guesses, which may be executed in parallel if the resources are available: (1) inversion of T, which requires $(2n)^2$ operations; (2) computation of $\varphi(\mathbf{d}_1)$ and $\varphi(\mathbf{d}_2)$, which requires n vectorized additions for a total of n^2 operations; (3) two matrix multiplications of n^3 operations each, followed by a matrix subtraction; (4) a Gaussian elimination of some $2n^3/3$ operations; (5) inversion of S requiring some n^2 operations; and finally (6) pruning, which has an almost constant expected running time. Thus, decryption has an expected running time of $O(q^a n^3)$. While this expression does involve an exponential factor, the exponent is rather small — on the order of $a \approx \log n$, so that decryption is still practically speaking a polynomial-time algorithm.

Fig. 4 emphasizes this exponential behavior by logarithmically plotting the decryption time as a function of a. Even a moderate increase in the number of dropped parameters can make decryption impractically slow.

4.3 Speed

Table 1 shows some timing results obtained from a straightforward C++ implementation on a 64-bit 3.3 GHz Intel CPU. Despite the scheme's obvious capacity for parallelism, it is not exploited beyond bit packing and vectorized addition (byte-wise xor) for $\mathbb{F}_2$. The only other optimization that was used was the compiler's optimization flag. For $q = 3$, the sizes are computed by representing elements of $\mathbb{F}_3$ by two bits.

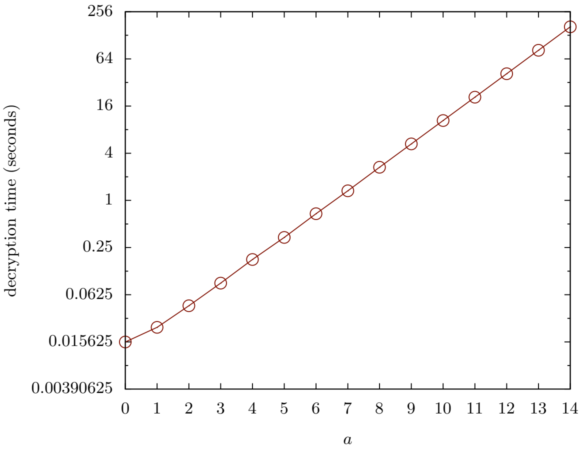


Fig. 4: Decryption time as a function of a for $n = 83$ and $q = 2$.

Table 1: Implementation results — timings of key generation, encryption and decryption algorithms along with public key, secret key and ciphertext size.

construction	sec. key	pub. key	ctxt.	key gen.	enc.	dec.
$\mathrm{EFC}_p^-, q = 2, n = 83, a = 10$	48.3 KB	509 KB	20 B	2.45 s	0.004 s	9.074 s
$\mathrm{EFC}_{pt^2}^-, q = 2, n = 83, a = 8$	48.3 KB	523 KB	20 B	3.982 s	0.004 s	2.481 s
$\mathrm{EFC}_p^-, q = 3, n = 59, a = 6$	48.8 KB	375 KB	28 B	2.938 s	0.004 s	12.359 s

5 Conclusion

Extension Field Cancellation (EFC) is a new construction for central trapdoors in MQ cryptosystems which exploits the commutativity of the extension field in order to cancel the complexity of the extension field polynomials. After cancellation, the plaintext can be obtained by solving a linear system. We anticipate several known attacks and use the projection and minus modifiers to inoculate EFC against these attacks.

We estimate parameters associated with 80 bits of security from the running time of an algebraic attack and offer some experimental validation of its complexity. Our implementation confirms the correctness of our schemes as well as their practical efficiency. Encryption can be done in only a few milliseconds, on par with other post-quantum cryptosystems such as NTRU [15] and McEliece [19]. However, due to the missing information from the minus modifier, decryption takes several seconds.

This minus modifier is an obvious candidate for improvement. While it is necessary for security, any significant number of dropped polynomials constitutes an onerous cost on the decryption function because its running time is exponential in this number. In fact, the minus modifier is ideally suited for MQ *signature* schemes, but ill-suited for MQ *encryption* schemes. The reason is that for signatures, any assignment to the missing variables will do; in contrast, the decryption algorithm must iterate over all possible assignments in order to find the correct plaintext. Any alternative modifier that has the same effect on security but obviates the need for exhaustive search can drastically accelerate decryption.

Another question is to determine to which extent the public keys can be shrunk. While it is difficult to shrink the secret keys without throwing away entropy, the public keys contain a large amount of redundancy. Even a relatively moderate reduction in the public key size can make the cryptosystem a feasible option for applications where the public key size is critical and currently too large.

Acknowledgments. The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work was supported by the Flemish Government, FWO WET G.0213.11N and by the European Commission through the ICT programme under contract FP7-ICT-2011-284833 PUFFIN, FP7-ICT-2013-10-SEP-210076296 PRACTICE, through the Horizon 2020research and innovation programme under grant agreement No H2020-ICT-2014-644371 WITDOM and H2020-ICT-2014-645622 PQCRYPTO; as well as by grant USDC (NIST) 60NAN15D059 from the Nation Institute of Standards of Technology. Alan Szepieniec is funded by a research grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

1. NESSIE, New European Schemes for Signatures, Integrity, and Encryption. Online: https://www.cosic.esat.kuleuven.be/nessie/ (2003), [accessed 2014-11-05]
2. Bettale, L., Faugère, J., Perret, L.: Cryptanalysis of hfe, multi-hfe and variants for odd and even characteristic. Des. Codes Cryptography 69(1), 1–52 (2013)
3. Billet, O., Patarin, J., Seurin, Y.: Analysis of intermediate field systems. IACR Cryptology ePrint Archive 2009, 542 (2009), http://eprint.iacr.org/2009/542
4. Courtois, N.: The security of hidden field equations (HFE). In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 266–281. Springer (2001), http://dx.doi.org/10.1007/3-540-45353-9_20
5. Ding, J., Hodges, T.J.: Inverting HFE systems is quasi-polynomial for all fields. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 724–742. Springer (2011)
6. Ding, J., Kleinjung, T.: Degree of regularity for HFE-. IACR Cryptology ePrint Archive 2011, 570 (2011), http://eprint.iacr.org/2011/570
7. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 164–175 (2005)
8. Ding, J., Yang, B.: Degree of regularity for hfev and hfev-. In: Gaborit, P. (ed.) PQCrypto 2013. Lecture Notes in Computer Science, vol. 7932, pp. 52–66. Springer (2013)
9. Ding, J., Yang, B., Cheng, C., Chen, C.O., Dubois, V.: Breaking the symmetry: a way to resist the new differential attack. IACR Cryptology ePrint Archive 2007, 366 (2007), http://eprint.iacr.org/2007/366
10. Dubois, V., Fouque, P., Shamir, A., Stern, J.: Practical cryptanalysis of SFLASH. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 1–12. Springer (2007)
11. Dubois, V., Fouque, P., Stern, J.: Cryptanalysis of SFLASH with slightly modified parameters. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 264–275. Springer (2007)
12. Faugère, J.C.: A new efficient algorithm for computing grÖbner bases without reduction to zero (f5). In: ISSAC 2002
13. Faugere, J.C.: A new efficient algorithm for computing gröbner bases (f 4). Journal of pure and applied algebra 139(1), 61–88 (1999)
14. Faugère, J., Joux, A.: Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using gröbner bases. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 44–60. Springer (2003)
15. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J. (ed.) ANTS-III. LNCS, vol. 1423, pp. 267–288. Springer (1998)
16. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) EUROCRYPT '99. LNCS, vol. 1592, pp. 206–222. Springer (1999)
17. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Wiener, M.J. (ed.) CRYPTO '99. LNCS, vol. 1666, pp. 19–30. Springer (1999)
18. Matsumoto, T., Imai, H.: Public quadratic polynominal-tuples for efficient signature-verification and message-encryption. In: Günther, C.G. (ed.) EUROCRYPT '88. LNCS, vol. 330, pp. 419–453. Springer (1988)
19. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. DSN progress report 42(44), 114–116 (1978)

20. Patarin, J.: Cryptoanalysis of the matsumoto and imai public key scheme of eurocrypt'88. In: Coppersmith, D. (ed.) CRYPTO '95. LNCS, vol. 963, pp. 248–261. Springer (1995)
21. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In: Maurer, U.M. (ed.) EUROCRYPT '96. LNCS, vol. 1070, pp. 33–48. Springer (1996)
22. Patarin, J., Goubin, L., Courtois, N.: $C^{*}{}_{-+}$ and HM: variations around two schemes of t. matsumoto and h. imai. In: Ohta, K., Pei, D. (eds.) ASIACRYPT '98. LNCS, vol. 1514, pp. 35–49. Springer (1998)
23. Porras, J., Baena, J., Ding, J.: Zhfe, a new multivariate public key encryption scheme. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 229–245. Springer (2014)
24. Shamir, A.: Efficient signature schemes based on birational permutations. In: Stinson, D.R. (ed.) CRYPTO '93. LNCS, vol. 773, pp. 1–12. Springer (1993)
25. Smith-Tone, D.: Properties of the discrete differential with cryptographic applications. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 1–12. Springer (2010)
26. Tao, C., Diene, A., Tang, S., Ding, J.: Simple matrix scheme for encryption. In: Gaborit, P. (ed.) PQCrypto 2013. LNCS, vol. 7932, pp. 231–242. Springer (2013)
27. Thomae, E.: About the Security of Multivariate Quadratic Public Key Schemes. Ph.D. thesis, Ruhr-Universität Bochum (2013)
28. Thomae, E., Wolf, C.: Cryptanalysis of Enhanced TTS, STS and all its variants, or: Why cross-terms are important. In: Mitrokotsa, A., Vaudenay, S. (eds.) AFRICACRYPT 2012. LNCS, vol. 7374, pp. 188–202. Springer (2012)
29. Wiedemann, D.H.: Solving sparse linear equations over finite fields. Information Theory, IEEE Transactions on 32(1), 54–62 (1986)
30. Wolf, C., Braeken, A., Preneel, B.: On the security of stepwise triangular systems. Des. Codes Cryptography 40(3), 285–302 (2006)
31. Wolf, C., Preneel, B.: Taxonomy of public key schemes based on the problem of multivariate quadratic equations. IACR Cryptology ePrint Archive 2005, 77 (2005)

6.2 A Practical Multivariate Blind Signature Scheme

Publication data

Albrecht Petzoldt and Alan Szepieniec and Mohamed Saied Emam Mohamed. "A Practical Multivariate Blind Signature Scheme" *Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers*, pp. 437–454, 2017.

Contributions

Contributing author

Notes

This paper is the product of a collaboration with Albrecht Petzoldt and Mohamed Mohamed. They had already been working on MQ signature schemes with special properties and were trying various constructions using the additivity of public keys. At some event I was in a conversation with Mohamed and the topic turned to blind signatures. After thinking for a few minutes I came to the conclusion that had to involve zero-knowledge proofs somehow, although I could not make the full picture work. However, it turns out that our ideas were perfectly complementary.

A Practical Multivariate Blind Signature Scheme

Albrecht Petzoldt[1], Alan Szepieniec[2], Mohamed Saied Emam Mohamed[3]
albrecht.petzoldt@nist.gov, alan.szepieniec@esat.kuleuven.be,
mohamed@cdc.informatik.tu-darmstadt.de

[1] Kyushu University, Fukuoka, Japan & NIST, USA
[2] KU Leuven, ESAT/COSIC & imec, Belgium
[3] Technische Universität Darmstadt, Germany

Abstract. Multivariate Cryptography is one of the main candidates for creating post-quantum cryptosystems. Especially in the area of digital signatures, there exist many practical and secure multivariate schemes. However, there is a lack of multivariate signature schemes with special properties such as blind, ring and group signatures. In this paper, we propose a technique to transform the Rainbow multivariate signature schemes into a blind signature scheme. The resulting scheme satisfies the usual blindness criterion and a one-more-unforgeability criterion adapted to MQ signatures, produces short blind signatures and is very efficient.

Keywords: Multivariate Cryptography, Blind Signatures, Rainbow Signature Scheme

1 Introduction

Cryptographic techniques are an essential tool to guarantee the security of communication in modern society. Today, the security of nearly all of the cryptographic schemes used in practice is based on number theoretic problems such as factoring large integers and solving discrete logarithms. The best known schemes in this area are RSA [25], DSA [14] and ECC. However, schemes like these will become insecure as soon as large enough quantum computers are built. The reason for this is Shor's algorithm [29], which solves number theoretic problems like integer factorization and discrete logarithms in polynomial time on a quantum computer. Therefore, one needs alternatives to those classical public key schemes which are based on hard mathematical problems not affected by quantum computer attacks (so called post-quantum cryptosystems).
The increasing importance of research in this area has recently been emphasized by a number of authorities. For example, the american National Security Agency has recommended governmental organizations to change their security infrastructures from schemes like RSA to post-quantum schemes [17] and the National Institute of Standards and Technologies (NIST) is preparing to standardize these schemes [18]. According to NIST, multivariate cryptography is one of the main candidates for this standardization process. Multivariate schemes

are in general very fast and require only modest computational resources, which makes them attractive for the use on low cost devices like smart cards and RFID chips [5,6]. However, while there exist many practical multivariate standard signature schemes such as UOV [15], Rainbow [9] and Gui [24], there is a lack of multivariate signature schemes with special properties such as blind, ring, and group signatures.

Blind signature schemes allow a user, who is not in charge of the private signing key, to obtain a signature for a message d by interacting with the signer. The important point is that this signer, who holds the secret key, receives no information about the message d that is signed nor about the signature s that is created through the interaction. Nevertheless, anyone with access to the public verification key is capable of verifying that signature. Because of these unlinkability and public verifiability properties, blind signature schemes are an indispensable primitive in a host of privacy-preserving applications ranging from electronic cash to anonymous database access, e-voting, and anonymous reputation systems.

In this paper, we present a technique to transform Rainbow, a multivariate quadratic (MQ) signature scheme, into a blind signature scheme. This transformation is accomplished by joining the MQ signature scheme with the zero-knowledge MQ-based identification scheme of Sakumoto *et al.* [28]. The user queries the signer on a blinded version of the message to be signed; the signer's response is then combined with the blinding information in order to produce a non-interactive zero-knowledge proof of knowledge of a pre-image under the public verification key, which is a set of quadratic polynomials that contains the signer's public key in addition to a large random term. The only way the user can produce such a proof is by querying the signer at some point for a partial pre-image; however, because it is zero-knowledge, this proof contains no information on the message that was seen and signed by the signer, thus preventing linkage and ensuring the user's privacy.

We obtain one of the first multivariate signature schemes with special properties and more generally one of the very few candidates for establishing practical and secure post-quantum blind signatures. In terms of security requirements, our scheme satisfies the usual blindness notion, but an adapted one-more-unforgeability one which we call *universal*-one-more-unforgeability. This change is justified by the observation that the usual one-more-unforgeability notion generalizes *existential* unforgeability for regular signatures; however, MQ signatures can only be shown to offer *universal* unforgeability and hence require a universal one-more-unforgeability generalization. While our technique applies to some other MQ signature schemes also, we instantiate our scheme with the Rainbow signature scheme and propose parameters targeting various levels of security.

The rest of this paper is organized as follows. Section 2 recalls the basic concepts of blind signatures and discusses the basic security notions. In Section 3 we recall the basic concepts of multivariate cryptography and review the Rainbow signature scheme, Sakumoto's multivariate identification scheme [28], and its transformation into a digital signature scheme due to Hülsing [12]. Section 4 presents our technique to extend multivariate signature schemes such as Rainbow to blind signature schemes, while Section 5 discusses the security of our construction. In Section 6 we give concrete parameter sets and analyze the efficiency of our scheme. Furthermore, in this section, we describe a proof of concept implementation of our scheme and compare it with other existing (classical and post-quantum) blind signature schemes. Finally, Section 7 concludes the paper.

2 Blind Signatures

Blind signature schemes as proposed by David Chaum in [3] allow a user, who is not in charge of the private signing key, to obtain a signature for a message d on behalf of the owner of the private key (called the signer). The key point hereby is that the signer gets no information about the content of the message d.

The signature generation process of a blind signature scheme is an interactive process between the user and the signer. In the first step, the user computes from the message d a blinded message $d^\star$ and sends it to the signer. The signer uses his private key to generate a signature $\sigma^\star$ for the message $d^\star$ and sends it back to the signer. Due to certain homomorphic properties in the inner structure of the blind signature scheme, the user is able to compute from $\sigma^\star$ a valid signature σ for the original message d. The receiver of a signed message can check the authenticity of the signature σ in the same way as in the case of a standard signature scheme. Figure 1 shows a graphical illustration of the signature generation process of a blind signature scheme.

Formally, a blind signature scheme $\mathcal{BS}$ is a three-tuple, consisting of two polynomial time algorithms `KeyGen` and `Verify` and an interactive signing protocol `Sign` [13].

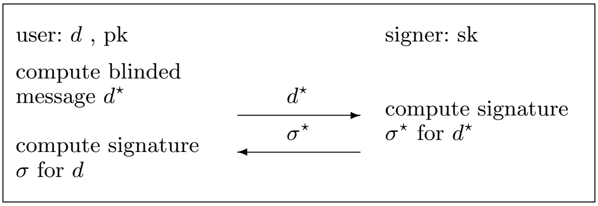

user: d , pk
signer: sk
compute blinded message $d^\star$
$d^\star$
compute signature $\sigma^\star$ for $d^\star$
$\sigma^\star$
compute signature σ for d

Fig. 1. Signature Generation Process of a Blind Signature Scheme

- `KeyGen`(1^κ): The probabilistic algorithm `KeyGen` takes as input a security parameter κ and outputs a key pair (sk, pk) of the blind signature scheme.

- `Sign`: The signature generation step is an interactive protocol between the `User`, who gets as input a message d and a public key pk and the `Signer` who is given the pair (pk, sk) generated by algorithm `KeyGen`. At the end of the protocol, the `Signer` outputs either "completed" or "non-completed", while the user outputs either "failed" or a signature σ.
- `Verify`$((d, \sigma), pk)$: The deterministic algorithm `Verify` takes as input a message/signature pair (d, σ) and the public key pk. It outputs **TRUE**, if σ is a valid signature for the message d and **FALSE** otherwise.

In the following, we assume the *correctness* of the blind signature scheme $\mathcal{BS}$: If both the `User` and the `Signer` follow the protocol, the `Signer` outputs always "completed", independently of the message d and the output (sk, pk) of the algorithm `KeyGen`. Similarly, the `User` always outputs a signature σ and we have

$$\Pr[\texttt{Verify}((d, \sigma), pk) = \textbf{TRUE}] = 1.$$

The basic security criteria of a blind signature scheme are Blindness and One-More-Unforgeability.

- **Blindness**: By signing the blinded message $d^{\star}$, the signer of a message gets no information about the content of the message to be signed nor about the final blind signature σ. More formally, blindness can be defined using the following security game.

  Game[Blindness]:
  1. The adversary $\mathcal{A}$ uses the algorithm `KeyGen` to generate a key pair (sk, pk) of the blind signature scheme. The public key pk is made public, while $\mathcal{A}$ keeps sk as his private key.
  2. The adversary $\mathcal{A}$ outputs two messages d_0 and d_1, which might depend on sk and pk.
  3. Let u_0 and u_1 be users with access to the public key pk but not to the secret key sk. For a random bit b that is unknown to $\mathcal{A}$, user u_0 is given the message d_b, while the message d_{1-b} is sent to user u_1. Both users engage in the interactive signing protocol (with $\mathcal{A}$ as signer), obtaining blind signatures σ_0 and σ_1 for the messages d_0 and d_1. The message/signature pairs (d_0, σ_0) and (d_1, σ_1) are given to the adversary $\mathcal{A}$.
  4. $\mathcal{A}$ outputs a bit $\bar{b}$. He wins the game, if and only if $\bar{b} = b$ holds.

  The blind signature scheme $\mathcal{BS}$ is said to fulfill the blindness property, if the advantage

  $$\mathrm{Adv}_{\mathcal{BS}}^{\mathsf{blindness}}(\mathcal{A}) = |2 \cdot \Pr[b' = b] - 1|$$

  for every PPT adversary $\mathcal{A}$ is negligible in the security parameter.

- **One-More-Unforgeability**: Even after having successfully completed L rounds of the interactive signing protocol, an adversary $\mathcal{A}$ not in charge of the private key sk cannot forge another valid blind signatures for a given message. More formally, we can define One-More-Unforgeability using the following game.

  Game [Universal-One-More-Unforgeability]
  1. The algorithm `KeyGen` is used to generate a key pair (sk, pk). The public key pk is given to the adversary $\mathcal{A}$, while sk is kept secret by the challenger.
  2. The adversary $\mathcal{A}$ engages himself in polynomially many interactive signing protocols with different instances of `Signer`. Let L be the number of cases in which the `Signer` outputs *completed.*
  3. $\mathcal{A}$ outputs a list $\mathcal{L}$ of L message / signature pairs. The challenger checks if all the message / signature pairs are valid and pairwise distinct.
  4. The challenger outputs a message $d^\star$ not contained in the list $\mathcal{L}$. The adversary wins the game, if he is able to generate a valid blind signature σ for the message $d^\star$, i.e. if `Verify`$((d^\star, \sigma), pk) =$ **TRUE** holds.

The blind signature scheme $\mathcal{BS}$ is said to provide the One-More-Unforgeability property, if the success probability

$$\Pr[\mathcal{A} \text{ wins}]$$

is, for any PPT adversary $\mathcal{A}$, negligible in the security parameter.
We note that this formalism is different from the standard security game for blindness, where the adversary is allowed to choose his own message but is required to forge at least $L + 1$ valid and distinct signatures. We choose to restrict the adversary's choice to accurately reflect the similar lack of choice in the standard security model for MQ signatures: *universal* unforgeability as opposed to *existential* unforgeability.
In the existential unforgeability game, the adversary wins whenever he is capable of producing any forgery, regardless of which message is signed. In contrast, in the universal unforgeability game the adversary obtains a message from the challenger and the adversary only wins if he can forge a signature for that specific message. Nevertheless, the universal adversary is allowed to query signatures after obtaining the target message; just not signatures on the same message. The reason why our formalism of universal-one-more-unforgeability does not allow blind-signature queries after delivering the target message to the adversary is precisely because the signature-queries are blind: the challenger should not be able to tell if it is the target message that is being blind-signed or something else.

3 Multivariate Cryptography

The basic objects of multivariate cryptography are systems of multivariate quadratic polynomials. Their security is based on the **MQ Problem**: Given m multivariate quadratic polynomials $p^{(1)}(\mathbf{x}), \ldots, p^{(m)}(\mathbf{x})$ in n variables $x_1, \ldots, x_n$, find a vector $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_n)$ such that $p^{(1)}(\bar{\mathbf{x}}) = \ldots = p^{(m)}(\bar{\mathbf{x}}) = 0$.

The MQ problem is proven to be NP-hard even for quadratic polynomials over the field GF(2) [11]. Moreover, it is widely assumed as well as experimentally validated that solving *random* instances of the MQ problem (with $m \approx n$) is a hard task, see for example [31].
To build a public key cryptosystem on the basis of the MQ problem, one starts with an easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ (central map). To hide the structure of $\mathcal{F}$ in the public key, one composes it with two invertible affine (or linear) maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. The *public key* of the scheme is therefore given by $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$. The *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$ and therefore allows to invert the public key.

Note: Due to the above construction, the security of multivariate schemes is not only based on the MQ-Problem, but also on the EIP-Problem ("Extended Isomorphism of Polynomials") of finding the decomposition of $\mathcal{P}$.
In this paper we concentrate on multivariate signature schemes. The standard signature generation and verification process of a multivariate signature scheme works as shown in Figure 2.

Signature Generation

$$\begin{tikzcd}
\mathbf{w} \in \mathbb{F}^m \arrow[r, "\mathcal{S}^{-1}"] & \mathbf{x} \in \mathbb{F}^m \arrow[r, "\mathcal{F}^{-1}"] & \mathbf{y} \in \mathbb{F}^n \arrow[r, "\mathcal{T}^{-1}"] & \mathbf{z} \in \mathbb{F}^n \arrow[lll, bend left=40, "\mathcal{P}"']
\end{tikzcd}$$

Signature Verification

Fig. 2. Standard workflow of multivariate signature schemes

Signature generation: To sign a message $\mathbf{w} \in \mathbb{F}^m$, one computes recursively $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x}) \in \mathbb{F}^n$ and $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$. The signature of the message $\mathbf{w}$ is $\mathbf{z} \in \mathbb{F}^n$. Here, $\mathcal{F}^{-1}(\mathbf{x})$ means finding one (of possibly many) pre-image of $\mathbf{x}$ under the central map $\mathcal{F}$.

Verification: To check the authenticity of a signature $\mathbf{z} \in \mathbb{F}^n$, one simply computes $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^m$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

3.1 The Rainbow Signature Scheme

The Rainbow signature scheme [9] is one of the most promising and best studied multivariate signature schemes. The scheme can be described as follows:

Let $\mathbb{F} = \mathbb{F}_q$ be a finite field with q elements, $n \in \mathbb{N}$ and $v_1 < v_2 < \ldots < v_\ell < v_{\ell+1} = n$ be a sequence of integers. We set $m = n - v_1$, $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ and $V_i = \{1, \ldots, v_i\}$ $(i = 1, \ldots, \ell)$.

Key Generation: The *private key* of the scheme consists of two invertible affine maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$ and a quadratic map $\mathcal{F}(\mathbf{x}) = (f^{(v_1+1)}(\mathbf{x}), \ldots, f^{(n)}(\mathbf{x})) : \mathbb{F}^n \to \mathbb{F}^m$. The polynomials $f^{(i)}$ $(i = v_1 + 1, \ldots, n\}$ are of the form

$$f^{(i)} = \sum_{k,l \in V_j} \alpha_{k,l}^{(i)} \cdot x_k \cdot x_l + \sum_{k \in V_j, l \in O_j} \beta_{k,l}^{(i)} \cdot x_k \cdot x_l + \sum_{k \in V_j \cup O_j} \gamma_k^{(i)} \cdot x_k + \eta^{(i)} \quad (1)$$

with coefficients randomly chosen from $\mathbb{F}$. Here, j is the only integer such that $i \in O_j$. The *public key* is the composed map $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$.

Signature Generation: To generate a signature for a document $\mathbf{w} \in \mathbb{F}^m$, we compute recursively $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x}) \in \mathbb{F}^n$ and $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$. Here, $\mathcal{F}^{-1}(\mathbf{x})$ means finding one (of approximately q^{v_1}) pre-image of $\mathbf{x}$ under the central map $\mathcal{F}$. This is done as shown in Algorithm 1.

Algorithm 1 Inversion of the Rainbow central map

```
Input: Rainbow central map $\mathcal{F}$, vector $\mathbf{x} \in \mathbb{F}^m$.
Output: vector $\mathbf{y} \in \mathbb{F}^n$ such that $\mathcal{F}(\mathbf{y}) = \mathbf{x}$.
1: Choose random values for the variables $y_1, \ldots, y_{v_1}$ and substitute these values into the polynomials $f^{(i)}$ $(i = v_1 + 1, \ldots, n)$.
2: for $k = 1$ to $\ell$ do
3:     Perform Gaussian Elimination on the polynomials $f^{(i)}$ $(i \in O_k)$ to get the values of the variables $y_i$ $(i \in O_k)$.
4:     Substitute the values of $y_i$ $(i \in O_k)$ into the polynomials $f^{(i)}$, $i \in \{v_{k+1} + 1, \ldots, n\}$.
5: end for
```

It might happen that one of the linear systems in step 3 of the algorithm does not have a solution. In this case one has to choose other values for $y_1, \ldots, y_{v_1}$ and start again. The signature of the document $\mathbf{w}$ is $\mathbf{z} \in \mathbb{F}^n$.

Signature Verification: To verify the authenticity of a signature $\mathbf{z} \in \mathbb{F}^n$, one simply computes $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^m$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

3.2 The MQ-based Identification Scheme

In [28] Sakumoto *et al.* proposed an identification scheme based on multivariate polynomials. There exist two versions of the scheme: a 3-pass and a 5-pass variant. In this section we introduce the 5-pass variant.
The scheme uses a system $\mathcal{P}$ of m multivariate quadratic polynomials in n variables as a public parameter. The prover chooses a random vector $\mathbf{s} \in \mathbb{F}^n$ as his secret key and computes the public key $\mathbf{v} \in \mathbb{F}^m$ by $\mathbf{v} = \mathcal{P}(\mathbf{s})$.
To prove his identity to a verifier, the prover performs several rounds of the interactive protocol shown in Figure 3.
Here,

$$\mathcal{G}(\mathbf{x}, \mathbf{y}) = \mathcal{P}(\mathbf{x} + \mathbf{y}) - \mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{y}) + \mathcal{P}(\mathbf{0}) \tag{2}$$

is the polar form of the system $\mathcal{P}$.
The scheme is a zero-knowledge argument of knowledge for a solution of the system $\mathcal{P}(\mathbf{x}) = \mathbf{v}$.
The knowledge error per round is $\frac{1}{2} + \frac{1}{2q}$. To decrease the impersonation probability below $2^{-\eta}$, one therefore needs to perform $r = \lceil \frac{-\eta}{\log_2(1/2+1/2q)} \rceil$ rounds of the protocol. For identification purposes, $\eta \approx 30$ may be sufficient, but for signatures we require η to be at least as large as the security level.

Prover: $\mathcal{P}, \mathbf{v}, \mathbf{s}$		**Verifier**: $\mathcal{P}, \mathbf{v}$
$\mathbf{r_0}, \mathbf{t_0} \in_R \mathbb{F}^n,\ \mathbf{e_0} \in_R \mathbb{F}^m$		
$\mathbf{r_1} = \mathbf{s} - \mathbf{r_0}$		
$c_0 = Com(\mathbf{r_0}, \mathbf{t_0}, \mathbf{e_0})$		
$c_1 = Com(\mathbf{r_1}, \mathcal{G}(\mathbf{t_0}, \mathbf{r_1}) + \mathbf{e_0})$	$\xrightarrow{(c_0, c_1)}$	
	$\xleftarrow{\alpha}$	$\alpha \in_R \mathbb{F}$
$\mathbf{t_1} = \alpha \mathbf{r_0} - \mathbf{t_0}$		
$\mathbf{e_1} = \alpha \mathcal{P}(\mathbf{r_0}) - \mathbf{e_0}$	$\xrightarrow{(\mathbf{t_1}, \mathbf{e_1})}$	$ch \in_R \{0, 1\}$
	$\xleftarrow{ch}$	
If $ch = 0$, resp $= \mathbf{r_0}$		
Else, resp $= \mathbf{r_1}$	$\xrightarrow{\text{resp}}$	
		If $ch = 0$, check
		$c_0 \overset{?}{=} Com(\mathbf{r_0}, \alpha \mathbf{r_0} - \mathbf{t_1},$ $\alpha \mathcal{P}(\mathbf{r_0}) - \mathbf{e_1})$
		If $ch = 1$, check
		$c_1 \overset{?}{=} Com(\mathbf{r_1}, \alpha(\mathbf{v} - \mathcal{P}(\mathbf{r_1}))$ $- \mathcal{G}(\mathbf{t_1}, \mathbf{r_1}) - \mathbf{e_1})$

Fig. 3. The 5-pass MQ identification scheme of Sakumoto *et al.* [28].

3.3 The MQDSS signature scheme

In [12], Hülsing et al. developed a technique to transform (2n+1) pass identification schemes into signature schemes. The technique can be used to transform the above described 5-pass multivariate identification scheme into an EU-CMA secure signature scheme.

To generate an MQDSS signature for a message d, the signer produces a transcript of the above identification protocol over r rounds. The challenges $\alpha_1, \dots, \alpha_r$ and $ch_1, \dots, ch_r$ are hereby computed from the message d and the commitments (using a publicly known hash function $\mathcal{H}$). Therefore, the signature has the form

$$\sigma = (c_{0,1}, c_{1,1}, \dots, c_{0,r}, c_{1,r}, t_{1,1}, e_{1,1}, \dots, t_{1,r}, e_{1,r}, \mathrm{resp}_1, \dots, \mathrm{resp}_r).$$

To check the authenticity of a signature σ, the verifier parses σ into its components, uses the commitments to compute the challenges α_i and ch_i $(i = 1, \dots, r)$ and checks the correctness of the responses resp_i as shown in Figure 3 (for $i = 1, \dots, r$).

4 Our Blind Signature Scheme

In this section we present MBSS, construction for blind signatures based on Rainbow. We chose to restrict our attention to Rainbow due to its short signatures and good performance. Moreover, the key sizes of Rainbow are acceptable and can be further reduced by the technique of Petzoldt *et al.* [22].

Nevertheless, our technique applies to any MQ signature scheme relying on the construction of Fig. 2, *i.e.*, relying on the hiding of a trapdoor to a quadratic map behind linear or affine transforms. As the other MQ signature schemes rely on the same construction, our technique applies to those cryptosystems as well. We do not use any property of Rainbow that is not shared by, *e.g.*, HFEv^- [24], pC^* [7], or UOV [15]. The exceptions are the MQ signature schemes that do not have the construction of Fig. 2, such as Quartz [19] and MQDSS [12].

4.1 The Basic Idea

The public key of our scheme consists of two multivariate quadratic systems $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^m$ and $\mathcal{R} : \mathbb{F}^m \to \mathbb{F}^m$. Hereby, $\mathcal{P}$ is the Rainbow public key, while $\mathcal{R}$ is a random system. The signer's private key allows him to invert the system $\mathcal{P}$ using the algorithm from Section 3.1.

In order to obtain a blind signature for a message (hash value) $\mathbf{w} \in \mathbb{F}^m$, the user chooses randomly a vector $\mathbf{z}^\star \in \mathbb{F}^m$, computes $\tilde{\mathbf{w}} = \mathbf{w} - \mathcal{R}(\mathbf{z}^\star)$ and sends $\tilde{\mathbf{w}}$ to the signer. The signer uses his private key to compute a signature $\mathbf{z}$ for the message $\tilde{\mathbf{w}}$ and sends it to the user. Therefore, the user obtains a solution $(\mathbf{z}, \mathbf{z}^\star)$ of the system $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{w}$. However, the user can not publish $(\mathbf{z}, \mathbf{z}^\star)$ as his signature for the document $\mathbf{w}$ since this would destroy the blindness of the scheme. Instead, the user has to prove knowledge of a solution to the system $\mathcal{P}(\mathbf{x}_1)+\mathcal{R}(\mathbf{x}_2) = \mathbf{w}$ using a zero knowledge protocol. We use the MQDSS technique (see Section 3.3) for this proof.

4.2 Description of the Scheme

In this section we give a detailed description of our blind signature scheme. As every blind signature scheme, MBSS consists of three algorithms *KeyGen*, *Sign* and *Verify*, where *Sign* is an interactive protocol between user and signer.

Parameters: Finite field $\mathbb{F}$, integers m, n and r (depending on a security parameter κ). r hereby determines, how many rounds of the identification scheme are performed during the generation of a signature.

Key Generation: The signer chooses randomly a Rainbow private key (consisting of two affine maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$ and a secret central map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$). He computes the public key $\mathcal{P}$ as $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$ (see Section 3.1) and uses a CSPRNG to generate the system $\mathcal{R} = \texttt{CSPRNG}(\mathcal{P}) : \mathbb{F}^m \to \mathbb{F}^m$. The *public key* of our blind signature scheme is the pair $(\mathcal{P}, \mathcal{R})$, the signer's *private key* consists of $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$. However, since $\mathcal{R}$ can be computed from the system $\mathcal{P}$, it is not necessary to publish $\mathcal{R}$ (if the CSPRNG in use is publicly accessible).

Signature Generation: The interactive signature generation process of our blind signature scheme can be described as follows: To get a signature for the message d with hash value $\mathcal{H}(d) = \mathbf{w} \in \mathbb{F}^m$, the user chooses randomly a vector $\mathbf{z}^\star \in \mathbb{F}^m$. He computes $\mathbf{w}^\star = \mathcal{R}(\mathbf{z}^\star) \in \mathbb{F}^m$ and sends $\tilde{\mathbf{w}} = \mathbf{w} - \mathbf{w}^\star \in \mathbb{F}^m$ to the signer. The signer uses his private key $(\mathcal{S}, \mathcal{F}, \mathcal{T})$ to compute a signature $\mathbf{z} \in \mathbb{F}^n$ such that $\mathcal{P}(\mathbf{z}) = \tilde{\mathbf{w}}$ and sends $\mathbf{z}$ back to the user, who therefore obtains a solution $(\mathbf{z}, \mathbf{z}^\star)$ of the system $\bar{\mathcal{P}}(\mathbf{x}) = \mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{w}$.
To prove this knowledge to the verifier in a zero knowledge way, the user generates an MQDSS signature for the message $\mathbf{w}$. As the public parameter of the scheme he hereby uses the system $\bar{\mathcal{P}}(\mathbf{x}) = \mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2)$, which is a system of m quadratic equations in $n+m$ variables. Furthermore, $\mathcal{G}(\mathbf{x}, \mathbf{y})$ is the polar form of the system $\bar{\mathcal{P}}$, i.e. $\mathcal{G}(\mathbf{x}, \mathbf{y}) = \bar{\mathcal{P}}(\mathbf{x} + \mathbf{y}) - \bar{\mathcal{P}}(\mathbf{x}) - \bar{\mathcal{P}}(\mathbf{y}) + \bar{\mathcal{P}}(\mathbf{0})$. In particular, the user performs the following steps.

1. Use a publicly known hash function $\mathcal{H}$ to compute $\mathcal{C} = \mathcal{H}(\mathcal{P}||\mathbf{w})$ and $\mathcal{D} = \mathcal{H}(\mathcal{C}||\mathbf{w})$.
2. Choose random values for $\mathbf{r}_{0,1}, \ldots, \mathbf{r}_{0,r}, \mathbf{t}_{0,1}, \ldots, \mathbf{t}_{0,r} \in \mathbb{F}^{m+n}$, $\mathbf{e}_{0,1}, \ldots, \mathbf{e}_{0,r} \in \mathbb{F}^m$, set $\mathbf{r}_{1,i} = (\mathbf{z}||\mathbf{z}^\star) - \mathbf{r}_{0,i}$ $(i = 1, \ldots, r)$ and compute the commitments
$$\begin{aligned} c_{0,i} &= Com(\mathbf{r}_{0,i}, \mathbf{t}_{0,i}, \mathbf{e}_{0,i}) \text{ and} \\ c_{1,i} &= Com(\mathbf{r}_{1,i}, \mathcal{G}(\mathbf{t}_{0,i}, \mathbf{r}_{1,i}) - \mathbf{e}_{0,i}) \ \ (i = 1, \ldots, r). \end{aligned}$$
Set COM $= (c_{0,1}, c_{1,1}, c_{0,2}, c_{1,2}, \ldots, c_{0,r}, c_{1,r})$.
3. Derive the challenges $\alpha_1, \ldots, \alpha_r \in \mathbb{F}$ from $(\mathcal{D}, COM)$.
4. Compute $\mathbf{t}_{1,i} = \alpha_i \cdot \mathbf{r}_{0,i} - \mathbf{t}_{0,i} \in \mathbb{F}^{m+n}$ and $\mathbf{e}_{1,i} = \alpha_i \cdot \bar{\mathcal{P}}(\mathbf{r}_{0,i}) - \mathbf{e}_{0,i}$ $(i = 1, \ldots, r)$. Set $Rsp_1 = (\mathbf{t}_{1,1}, \mathbf{e}_{1,1}, \ldots, \mathbf{t}_{1,r}, \mathbf{e}_{1,r})$.
5. Derive the challenges $(ch_1, \ldots, ch_r)$ from $(\mathcal{D}, COM, Rsp_1)$.
6. Set $Rsp_2 = (\mathbf{r}_{ch_1,1}, \ldots, \mathbf{r}_{ch_r,r})$.

7. The blind signature σ for the message $\mathbf{w} \in \mathbb{F}^m$ is given by

$$\sigma = (\mathcal{C}, COM, Rsp_1, Rsp_2).$$

The length of the blind signature σ is given by

$$|\sigma| = 1 \cdot |\text{hash value}| + 2r \cdot |\text{Commitment}| + r \cdot (2n + 3m) \ \ \mathbb{F}-\text{elements}.$$

Figure 4 shows the full protocol for obtaining a blind signature.
Signature Verification: To check the authenticity of a blind signature σ for a message d with hash value $\mathbf{w} \in \mathbb{F}^m$, the verifier parses σ into its components and computes $\mathcal{D} = \mathcal{H}(\mathcal{C}||\mathbf{w})$. He derives the challenges $\alpha_i \in \mathbb{F}$ from $(\mathcal{D}, COM)$ and ch_i from $(\mathcal{D}, COM, Rsp_1)$ $(i = 1, \ldots, r)$.
Finally, he parses COM into $(c_{0,1}, c_{1,1}, c_{0,2}, c_{1,2}, \ldots, c_{0,r}, c_{1,r}), Rsp_1$ into $\mathbf{t}_1, \mathbf{e}_1, \ldots, \mathbf{t}_r, \mathbf{e}_r$ and Rsp_2 into $\mathbf{r}_1, \ldots, \mathbf{r}_r$ and checks if, for all $i = 1, \ldots, r$, $\mathbf{r}_i$ is a correct response to ch_i with respect to COM, $\mathbf{t}_i$ and $\mathbf{e}_i$, i.e.

$$\begin{aligned} c_{0,i} &\stackrel{?}{=} Com(\mathbf{r_i}, \alpha_i \cdot \mathbf{r_i} - \mathbf{t_i}, \alpha_i \cdot \mathcal{P}(\mathbf{r_i}) - \mathbf{e_i}) \ \ (\text{for } ch_i = 0) \\ c_{1,i} &\stackrel{?}{=} Com(\mathbf{r_i}, \alpha_i \cdot (\mathbf{w} - \mathcal{P}(\mathbf{r_i})) - \mathcal{G}(\mathbf{t_i}, \mathbf{r_i}) - \mathbf{e_i}) \ \ (\text{for } ch_i = 1). \end{aligned} \tag{3}$$

If all of these tests are fulfilled, the blind signature σ is accepted, otherwise rejected.

Note: As the resulting blind signature depends on the randomness sampled for generating the zero-knowledge proof, there may be many signatures associated to one tuple $(\mathbf{z}, \mathbf{z}^\star)$. To prevent a malicious user from reusing the same preimage to $\mathcal{P}(\bar{\mathbf{x}}_1) + \mathcal{R}(\bar{\mathbf{x}}_2)$, two signatures to messages d_1, d_2 are considered *essentially* different whenever $\mathbf{w}_1 = \mathcal{H}(d_1) \neq \mathbf{w}_2 = \mathcal{H}(d_2)$. In other words, the zero-knowledge proof is taken into account for validity but not for distinctness.

4.3 Reducing the Signature Length

In this section we present a technique to reduce the length of the blind signature σ, which was already mentioned in [28] and [12].
Instead of including all of the commitments $c_{0,1}, c_{1,1}, \ldots, c_{0,r}, c_{1,r}$ into the signature, we just transmit $COM = \mathcal{H}(c_{0,1}||c_{1,1} \ldots c_{0,r}||c_{1,r})$. However, in this scenario, we have to add $(c_{1-ch_1,1}, \ldots, c_{1-ch_r,r})$ to Rsp_2. In the verification process, the verifier recovers $(c_{ch_1,1}, \ldots, c_{ch_r,r})$ by equation (3) and checks if

$$COM \stackrel{?}{=} \mathcal{H}(c_{0,1}, c_{1,1}, \ldots, c_{0,r}, c_{1,r})$$

is fulfilled. By doing so, we can reduce the length of the blind signature σ to

$$|\sigma| = 2 \cdot |\text{hash value}| + r \cdot (2n + 3m) \ \mathbb{F} \text{ elements} + r \cdot |\text{Commitment}| \ \ .$$

User: $\mathcal{P}, \mathcal{R}, \mathcal{H}, d$ **Signer**: $\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{P}, \mathcal{R}$

$\boxed{1}$ $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$,

$\mathbf{z}^\star \in_R \mathbb{F}^m$,

$\mathbf{w}^\star = \mathcal{R}(\mathbf{z}^\star) \in \mathbb{F}^m$,

$\tilde{\mathbf{w}} = \mathbf{w} - \mathbf{w}^\star \in \mathbb{F}^m \quad \longrightarrow \quad \tilde{\mathbf{w}} \in \mathbb{F}^m$

$\boxed{2}$ $\mathbf{z} \in \mathbb{F}^{\mathbf{n}} \quad \longleftarrow \quad \mathbf{z} = \mathcal{T}^{-1} \circ \mathcal{F}^{-1} \circ \mathcal{S}^{-1}(\tilde{\mathbf{w}})$

$\bar{\mathcal{P}}(\mathbf{z}, \mathbf{z}^\star) = \mathcal{P}(\mathbf{z}) + \mathcal{R}(\mathbf{z}^\star) \overset{?}{=} \mathbf{w}$, abort if not true

$\boxed{3}$ $\mathcal{G}(\mathbf{x}, \mathbf{y}) = \bar{\mathcal{P}}(\mathbf{x} + \mathbf{y}) - \bar{\mathcal{P}}(\mathbf{x}) - \bar{\mathcal{P}}(\mathbf{y}) + \bar{\mathcal{P}}(\mathbf{0})$,

$\mathcal{C} = \mathcal{H}(\mathcal{P}||\mathbf{w})$ and $\mathcal{D} = \mathcal{H}(\mathcal{C}||\mathbf{w})$,

$\mathbf{r}_{0,1}, \ldots, \mathbf{r}_{0,r}, \mathbf{t}_{0,1}, \ldots, \mathbf{t}_{0,r} \in_R \mathbb{F}^{m+n}$, $\mathbf{e}_{0,1}, \ldots, \mathbf{e}_{0,r} \in_R \mathbb{F}^m$,

$\mathbf{r}_{1,i} = (\mathbf{z}||\mathbf{z}^\star) - \mathbf{r}_{0,i},\ i \in \{1, \ldots, r\}$,

$c_{0,i} = Com(\mathbf{r}_{0,i}, \mathbf{t}_{0,i}, \mathbf{e}_{0,i})$,

$c_{1,i} = Com(\mathbf{r}_{1,i}, \mathcal{G}(\mathbf{t}_{0,i}, \mathbf{r}_{1,i}) - \mathbf{e}_{0,i}),\ i \in \{1, \ldots, r\}$,

$\mathrm{COM} = (c_{0,1}, c_{1,1}, c_{0,2}, c_{1,2}, \ldots, c_{0,r}, c_{1,r})$,

$(\mathcal{D}, COM) \Rightarrow \alpha_1, \ldots, \alpha_r \in \mathbb{F}$,

$\mathbf{t}_{1,i} = \alpha_i \cdot \mathbf{r}_{0,i} - \mathbf{t}_{0,i} \in \mathbb{F}^{m+n}$,

$\mathbf{e}_{1,i} = \alpha_i \cdot \bar{\mathcal{P}}(\mathbf{r}_{0,i}) - \mathbf{e}_{0,i} \quad (i = 1, \ldots, r)$,

$Rsp_1 = (\mathbf{t}_{1,1}, \mathbf{e}_{1,1}, \ldots, \mathbf{t}_{1,r}, \mathbf{e}_{1,r})$,

$(\mathcal{D}, COM, Rsp_1) \Rightarrow (ch_1, \ldots, ch_r)$,

$Rsp_2 = (\mathbf{r}_{ch_1,1}, \ldots, \mathbf{r}_{ch_r,r})$,

$\sigma = (\mathcal{C}, COM, Rsp_1, Rsp_2)$.

Fig. 4. Our blind signing protocol.

4.4 Correctness

Theorem 1. *Blind signatures generated by honest participants in the protocols of our multivariate blind signature scheme will be accepted with probability 1.*

Proof. The proof consists out of two steps. In the first step we show that, at the end of the interactive process, the user obtains a solution $(\mathbf{z}, \mathbf{z}^\star)$ of the system $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{w}$. This can be seen as follows. In the course of the interactive protocol, the (honest) user chooses randomly a vector $\mathbf{z}^\star$, computes $\mathbf{w}^\star = \mathcal{R}(\mathbf{z}^\star)$ and $\tilde{\mathbf{w}} = \mathbf{w} - \mathbf{w}^\star$ and sends $\tilde{\mathbf{w}}$ to the signer. The (honest) signer uses his private key to compute a vector $\mathbf{z}$ such that $\mathcal{P}(\mathbf{z}) = \tilde{\mathbf{w}}$. Altogether, we get $\mathcal{P}(\mathbf{z}) + \mathcal{R}(\mathbf{z}^\star) = \tilde{\mathbf{w}} + \mathbf{w}^\star = \mathbf{w} - \mathbf{w}^\star + \mathbf{w}^\star = \mathbf{w}$, which means that $(\mathbf{z}, \mathbf{z}^\star)$ is indeed a solution of the public system $\bar{\mathcal{P}}(\mathbf{x}) = \mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2)$.
In the second step we simply use the correctness proof of the MQDSS [12] to show that an MQDSS signature produced by an honest signer knowing a solution to the public system $\bar{\mathcal{P}}$ is, by an honest verifier, accepted with probability 1.

5 Security

In this section, we analyze the security of our construction, assuming abstractly that Rainbow is secure. (For a concrete security analysis of the underlying Rainbow scheme we refer to [21].) For this, we have to show the blindness and one-more-unforgeability of the derived scheme.

5.1 Blindness

Theorem 2. *Assume that the distribution of $\mathcal{R}(\mathbf{x})$ for uniform $\mathbf{x} \in \mathbb{F}_q^m$ is computationally indistinguishable from uniform, and assume that a perfectly hiding commitment scheme is used. Then our multivariate blind signature scheme provides blindness against any computationally bounded adversary. In particular, for all PPT adversaries $\mathcal{A}$, their advantage in the blindness game (of Section 2) for our scheme is at most negligible:*

$$\forall \mathcal{A} \,.\, \mathrm{Adv}^{\mathsf{blindness}}_{\mathcal{MBSS}}(\mathcal{A}) \leq \mathsf{negl} \ .$$

Proof. The adversary has to link $\tilde{\mathbf{w}}$ from one interaction, to the pair (d, σ) from another interaction. Due to the perfect zero-knowledge property of the perfectly hiding commitment scheme, σ contains no information about the solution $(\mathbf{z}, \mathbf{z}^\star)$ and hence no information about $\mathcal{R}(\mathbf{z}^\star)$ or $\mathcal{P}(\mathbf{z})$. Therefore the adversary's task is equivalent linking $\tilde{\mathbf{w}}$ to d, since knowledge of σ gives him no advantage. However, $\mathbf{z}^\star$ is chosen uniformly at random and so $\mathcal{R}(\mathbf{z}^\star)$ is computationally indistinguishable from uniform. As a result, the blinded message $\tilde{\mathbf{w}} = \mathbf{w} - \mathcal{R}(\mathbf{z}^\star)$ is computationally indistinguishable from uniform and no polynomial-time adversary can compute any predicate of $\mathbf{w}$ from $\tilde{\mathbf{w}}$ with more than a negligible success probability. This includes the predicate $\mathcal{H}(d) \overset{?}{=} \mathbf{w}$ or any similar predicate that would allow the adversary to link $\tilde{\mathbf{w}}$ to d.

5.2 Universal One-More-Unforgeability

Theorem 3. *If Rainbow is secure and if finding a solution* $(\mathbf{x}_1, \mathbf{x}_2)$ *to* $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{0}$ *for a randomly chosen quadratic map* $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ *and a Rainbow public key* $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ *is a hard problem, then our multivariate blind signature scheme satisfies universal-one-more-unforgeability against computationally bounded adversaries. That is to say, for all PPT adversaries* $\mathcal{A}$*, their advantage in winning the universal-one-more-unforgeability game (of Section 2) is at most negligible:*

$$\forall \mathcal{A} \,.\, \mathsf{Adv}_{\mathcal{MBSS}}^{\mathsf{universal-one-more-unforgeability}}(\mathcal{A}) \leq \mathsf{negl} \ .$$

Proof. We present a sequence of games argument showing that any adversary winning the Universal-One-More-Unforgeability game logically implies that the mentioned hard problem is efficiently solvable.

Let **Game 0** be the universal-one-more-unforgeability game as defined in Section 2. By assumption, we have an adversary $\mathcal{A}$ who wins with noticeable probability in polynomial time.

Let **Game 1** be the universal-one-more-unforgeability game but for the modified blind signature scheme where for each signature knowledge of $(\mathbf{z}, \mathbf{z}^\star)$ satisfying $\mathcal{P}(\mathbf{z}) + \mathcal{R}(\mathbf{z}^\star) = \mathcal{H}(d)$ is proven interactively using the protocol of Section 3.2, instead of producing a non-interactive proof σ. The simulator can win this game by simulating an instance of **Game 0** and presenting the **Game 0**-adversary with a random oracle that is programmed to respond with the same challenge-message that the simulator receives from the challenger.

Let **Game 2** be the universal-one-more-unforgeability game for the modified scheme that drops blindness altogether. Instead of proving knowledge of $(\mathbf{z}, \mathbf{z}^\star)$ in zero-knowledge, knowledge is proven straightforwardly by simply sending this pair to the challenger. The simulator can win this game by simulating **Game 1** and using the extractor machine associated with the zero-knowledge proof to obtain $(\mathbf{z}, \mathbf{z}^\star)$.

Let **Game 3** be the universal unforgeability under chosen message attack game for the signature scheme whose public key is $(\mathcal{P}, \mathcal{R})$, with the additional option for the adversary to query inverses under $\mathcal{P}$ as long as the message $d^\star$, the message for which a signature is to be forged, was not yet sent. The simulator wins this game by simulating **Game 2**. The blind-signature requests are answered by querying for an inverse under $\mathcal{P}$. After the adversary outputs his list $\mathcal{L}$ of message / signature pairs, the simulator requests the message $d^\star$ from the challenger for which a signature is to be forged. This message is relayed to the simulated adversary.

Let **Game 4** be the proper universal unforgeability under chosen message attack game for the signature scheme whose public key is $(\mathcal{P}, \mathcal{R})$, *i.e.*, without the ability to query for inverses under $\mathcal{P}$. Heuristically, the same adversary that wins **Game 3** should win **Game 4**. The reason is that the ability to query inverses under $\mathcal{P}$ before $d^\star$ is known does not help the adversary at all. Since $\mathcal{P}$ is a Rainbow public key and Rainbow is secure in its own right, the ability to query inverses should not help the adversary to either recover the secret key or find

his own inverses. Otherwise it would be possible to mount an attack exploiting this fact.
Let **Game 5** be the following non-interactive game, or problem: given $(\mathcal{P}, \mathcal{R})$, find $(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{F}_q^n \times \mathbb{F}_q^m$ such that $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = 0$. The simulator can solve this problem by picking a random $\mathbf{s} \in_R \mathbb{F}_q^m$. He then simulates **Game 4** and presents its adversary with $(\mathcal{P}, \mathcal{R} + \mathbf{s})$ and with access to the backdoored random oracle $\mathcal{H}'(x) = \mathcal{P}(\mathcal{H}_1(x)) + \mathcal{R}(\mathcal{H}_2(x)) + \mathbf{s}$, where $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{F}_q^n$ and $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{F}_q^m$ are true random oracles. Under the (very reasonable) assumption that the distribution of $\mathcal{H}'$ is computationally indistinguishable from that of a true random oracle, the adversary's winning probability is still significant. The simulator answers a signature query $d \in \{0,1\}^*$ with $(\mathbf{x}_1, \mathbf{x}_2)$ where $\mathbf{x}_1 = \mathcal{H}_1(d)$ and $\mathbf{x}_2 = \mathcal{H}_2(d)$, which is necessarily a valid signature from the point of view of the adversary who can verify that $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) + \mathbf{s} = \mathcal{H}'(d)$. When the adversary indicates he is done with querying signatures, the simulator chooses a new message $d^\star$, programs $\mathcal{H}'(d^\star) = \mathbf{s}$, and sends $d^\star$ to the adversary. A winning adversary therefore solves $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) + \mathbf{s} = \mathbf{s}$, which is hard because it is equivalent to solving $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{0}$. This concludes the proof of Thm. 3.

One of the premises of Thm. 3 remains to be shown: that finding a solution to the system $\bar{\mathcal{P}}(\mathbf{x}) = \mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{0}$, which is a system of m quadratic equations in $n+m$ variables, is a difficult task. We have no rigorous proof for this (such a proof would imply $\mathbf{P} \neq \mathbf{NP}$) but we justify making this assumption based on common hardness arguments from MQ cryptography. In particular, there are two attack strategies known against multivariate systems:

Direct Attacks: In a direct attack, one tries to solve the system $\bar{\mathcal{P}}(\mathbf{x}) = \mathbf{0}$ as an instance of the MQ Problem. Since the system $\bar{\mathcal{P}}$ is underdetermined, there are two possibilities to do this. One can use a special algorithm against underdetermined multivariate systems [30] or, after fixing n of the variables, a Gröbner Basis algorithm such as Faugéres F_4 [10]. For suitably chosen parameters, both approaches are infeasible.

The second possibility to solve a multivariate system such as $\mathcal{P}'$ are the so called **Structural Attacks**. In this type of attack one uses the known structure of the system $\bar{\mathcal{P}}$ in order to find a decomposition $\bar{\mathcal{P}}$ into easily invertible maps. Note that, in our case we can write

$$\begin{aligned}\bar{\mathcal{P}}(\mathbf{x}) &= \mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2)\\ &= \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}(\mathbf{x}_1) + \mathcal{S} \circ \underbrace{\mathcal{S}^{-1} \circ \mathcal{R}}_{\mathcal{R}'}(\mathbf{x}_2)\\ &= \mathcal{S} \circ \underbrace{(\mathcal{F} + \mathcal{R}')}_{\mathcal{F}'} \circ \mathcal{T}'(\mathbf{x}),\end{aligned}$$

where the matrix T' representing the linear transformation $\mathcal{T}'$ is given by

$$T' = \begin{pmatrix} T & 0 \\ 0 & 1_m \end{pmatrix} \in \mathbb{F}^{(n+m)\times(n+m)}.$$

In order to solve the system $\bar{\mathcal{P}}$ using a structural attack, we have to use the known structure of the map $\mathcal{F}' = \mathcal{F} + \mathcal{S}^{-1} \circ \mathcal{R}$ to recover the linear maps $\mathcal{S}$ and $\mathcal{T}'$ (or, since the structure of $\mathcal{T}'$ is mostly known, the matrix T). However, since the coefficients of both $\mathcal{S}$ and $\mathcal{R}$ are chosen uniformly at random, the map $\mathcal{R}' = \mathcal{S}^{-1} \circ \mathcal{R}$ is a random quadratic map over $\mathbb{F}^m$. The only structure we can use for a structural attack therefore comes from the map $\mathcal{F}$, which is the central map of the underlying multivariate signature scheme. Therefore, we are in exactly the same situation as if attacking the underlying multivariate scheme using a structural attack. This means that a structural attack against our blind signature scheme is at least as hard as a structural attack against the underlying multivariate signature scheme. By choosing the parameters of the underlying scheme in an appropriate way, we therefore can prevent this type of attack against our blind signature scheme.

5.3 Quantum Security

The technique proposed in [12] is capable of transforming $(2n + 1)$-pass zero-knowledge proofs into non-interactive zero-knowledge proofs that are secure against classical adversaries in the random oracle model. However, the behaviour of this transform against quantum adversaries is not well understood because the random oracle should be accessible to the quantum adversary and answer queries *in quantum superposition*, and many standard proof techniques do not carry over to this setting. See Boneh et al. [2] for an excellent treatment of proofs that fail in the quantum random oracle model.

Formally proving soundness against quantum adversaries seems to be a rather involved task beyond the scope of this paper. Instead, we are content to conjecture that there exists a commitment scheme such that the technique of [12] results in a non-interactive zero-knowledge proof that is secure against quantum adversaries as well as classical ones. This conjecture is implicit in the works of Sakumoto et al. [28], and Hülsing et al. [12].

6 Discussion

6.1 Parameters

In this section we propose concrete parameter sets for our blind signature scheme. As observed in the previous section, we have to choose the parameters in a way that

a) solving a random system of m quadratic equations in m variables is infeasible,
b) inverting an MQ public key with the given parameters is infeasible, and
c) a direct attack against a system of m quadratic equations in $n+m$ variables is infeasible.

Since condition (a) is implied by (c), we only have to consider (b) and (c). In order to defend our scheme against attacks of type (b), we follow the recommendations of [21]. Regarding (c), we have to consider that the system $\mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2) = \mathbf{w}$ is highly underdetermined (in the case of $\mathcal{P}$ being a Rainbow public key, the number of variables in this system exceeds the number of equations by a factor of about 3). As a result of Thomae et al. shows, such systems can be solved significantly faster than determined systems.

Proposition 1. *[30] Solving an MQ system of m equations in $n = \omega \cdot m$ variables is only as hard as solving a determined MQ system of $m - \lfloor \omega \rfloor + 1$ equations.*

According to this result, we have to increase the number of equations in our system by 2 (compared to the parameters of a standard Rainbow instance). Table 1 shows the parameters we propose for our scheme for various targeted security levels.

security level (bit)	parameters $(\mathbb{F}, (v_1, o_1, o_2))$	# rounds	public key size (kB)	private key size (kB)	blind sig. size (kB)
80	(GF(31),(16,18,17))	84	29.4	20.1	11.5
100	(GF(31),(20,22,21))	105	54.6	36.6	17.6
128	(GF(31),(25,27,27))	135	106.8	70.2	28.5
192	(GF(31),(37,35,35))	202	342.8	219.0	63.2
256	(GF(31),(50,53,53))	269	802.4	507.1	111.9

Table 1. Proposed parameters for our blind signature scheme (GF(31)).

6.2 Efficiency

During the interactive part of the signature generation process, the signer has to generate one Rainbow signature for the message $\tilde{\mathbf{w}} = \mathbf{w} - \mathbf{w}^\star$.

For the user, the most costly part of the signature generation is the repeated evaluation of the system $\bar{\mathcal{P}}(\mathbf{x}) = \mathcal{P}(\mathbf{x}_1) + \mathcal{R}(\mathbf{x}_2)$. During the computation of the commitments $\mathbf{c}_{0,i}$ and $\mathbf{c}_{1,i}$ $(i = 1, \ldots, r)$ (step 2 of the signature generation process) this has to be done $3 \cdot r$ times (one evaluation of $\mathcal{G}$ corresponds to 3 evaluations of $\bar{\mathcal{P}}$). In step 4 of the process (computation of $\mathbf{e}_{1,i}$) we need r evaluations of $\bar{\mathcal{P}}$. Altogether, the user has to evaluate the system $4r$ times.

During verification, the verifier has to compute the commitments $c_{ch_i,i}$ $(i = 1, \ldots, r)$. If $ch_i = 0$, he needs for this 1 evaluation of $\bar{\mathcal{P}}$, in the case of $ch_2 = 1$ he needs 4 evaluations. On average, the verifier needs therefore $\frac{r}{2} \cdot (1 + 4) = 2.5 \cdot r$ evaluations of the system $\bar{\mathcal{P}}$.

While the system $\bar{P}$ consists of m quadratic equations in $m + n$ variables, the inner structure of the system can be used to speed up the evaluation. In fact, the system $\bar{\mathcal{P}}$ is the sum of two smaller systems $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^m$ and $\mathcal{R} : \mathbb{F}^m \to \mathbb{F}^m$. Therefore, we can evaluate $\bar{\mathcal{P}}$ by evaluating $\mathcal{P}$ and $\mathcal{R}$ separately and adding the results.

6.3 Implementation

We implemented all functionalities in Sage [27] to prove concept validity. Table 2 contains the timing results for the matching parameter sets of Table 1, demonstrating that our scheme is somewhat efficient and practicable even for very poorly-optimized Sage code. These results were obtained on a 3.3 GHz Intel Quadcore with 6,144 kB of cache.

Despite of these relatively large numbers, we are very optimistic about the speed of our blind signatures when implemented in a less abstract and more memory-conscious programming language. For instance, Hülsing et al.'s optimized MQDSS manages to generate (classically) 256-bit-secure signatures in 6.79 ms and verify them in even less time [12]. As the MQDSS represents the bottleneck of our scheme, a similarly optimized implementation could potentially drop signature generation and verification time by several orders of magnitude.

sec. lvl.	Key Gen.	Sign (`Signer`)	Sig. Gen. (`User`)	Sig. Verification
80	4,007	7	2,018	1,424
100	9,392	13	3,649	2,656
128	25,517	19	7,760	5,505
192	87,073	41	23,692	16,040
256	613,968	103	86,540	59,669

Table 2. Timing results of a Sage implementation of our blind signature scheme. All units are milliseconds, except for the security level.

6.4 Comparison

Table 3 shows a comparison of our scheme to the standard RSA blind signature scheme and the lattice-based blind signature scheme of Rückert [26]. The RSA blind signature scheme does not offer any security against quantum computers. The public keys of Rückert's scheme are smaller than those of our scheme, although ours are still competitive. Like the standard RSA blind signature scheme, our scheme requires 2 steps of communication between the user and the signer in order to produce the blind signature. This is in contrast to Rückert's scheme where this number is 4. More importantly, our scheme outperforms that of Rückert in terms of signature size.

At this point, an apples-to-apples comparison of operational speed is not possible. Nevertheless, regardless of speed, the main selling point of our scheme is its reliance on different computational problems from those used in other branches of cryptography, including lattice-based cryptography.

Security lvl. (bit)	Scheme	comm.	Pub. key size (kB)	Sig. size (kB)	Post-quantum?
76	RSA-1229	2	1.2	1.2	×
	Lattice-1024	4	10.2	66.9	✓
	Our scheme(GF(31),16,18,17)	2	29.4	11.5	✓
102	RSA-3313	2	3.3	3.3	×
	Lattice-2048	4	23.6	89.4	✓
	Our scheme(GF(31),20,22,21)	2	54.6	17.6	✓

Table 3. Comparison of different blind signature schemes. The secrutiy levels are adopted from Rückert [26].

7 Conclusion

In this paper we proposed the first multivariate based blind signature scheme. Our scheme is very efficient and produces much shorter blind signatures than the lattice based scheme of Rückert [26], making our scheme the most promising candidate for establishing a post-quantum blind signature scheme.

Our construction is notably generic. While we only show that it applies to Rainbow and MQDSS, we use their properties abstractly and it is perfectly conceivable that another combination of trapdoor-based MQ signature scheme with a non-interactive proof of knowledge of the solution to an MQ system will give the same result. Indeed, our design demonstrates that the combination of a dedicated signature scheme with an identification scheme relying on the same hard problem, is a powerful construction — and may apply in other branches of cryptography as well.

Lastly, one major use case of blind signatures is anonymous identification. In this scenario, one may reasonably dispense with the transformed signature scheme and instead directly use the underlying interactive identification scheme, thus sacrificing non-interactivity for less computation and bandwidth. Likewise, other use cases such as anonymous database access require *reusable* anonymous credentials. Our scheme can be adapted to fit this scenario as well, simply by specifying that all users obtain a blind signature on the same public parameter.

Acknowledgements

The authors would like to thank the reviewers and the shepherd in particular for their helpful comments. This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work was supported by the European Commission through the Horizon 2020research and innovation programme under grant agreement No H2020-ICT-2014-644371 WITDOM, H2020-ICT-2014-645622 PQCRYPTO and H2020-DS-2014-653497 PANORAMIX, and through the SECURITY programme under FP7-SEC-2013-1-607049 EKSISTENZ. Alan Szepieniec is being supported by a doctoral grant of the Flemish Agency for Innovation and Entrepreneurship (VLAIO, formerly IWT).

References

1. M. Bellare, C. Namprempre, D. Pointcheval, M. Semanko: The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. Journal of Cryptology. Volume 16, Issue 3, pp. 185 - 215. Springer, Jun. 2003.
2. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, M. Zhandry: Random oracles in a quantum world. In International Conference on the Theory and Application of Cryptology and Information Security, pp. 41-69. Springer Berlin Heidelberg, 2011.
3. D. Chaum: Blind Signatures for untraceable payment. Proceedings of CRYPTO 1982, pp. 199 - 203. Plenum Press, 1983.
4. D.J. Bernstein, J. Buchmann, E. Dahmen (eds.): post-quantum Cryptography. Springer, 2009.
5. A. Bogdanov, T. Eisenbarth, A. Rupp, C. Wolf: Time-area optimized public-key engines: MQ-cryptosystems as replacement for elliptic curves? CHES 2008, LNCS vol. 5154, pp. 45-61. Springer, 2008.
6. A.I.T. Chen, M.-S. Chen, T.-R. Chen, C.-M. Cheng, J. Ding, E. L.-H. Kuo, F. Y.-S. Lee, B.-Y. Yang: SSE implementation of multivariate PKCs on modern x86 cpus. CHES 2009, LNCS vol. 5747, pp. 33 - 48. Springer, 2009.
7. J. Ding, V. Dubois, B.-Y. Yang, O. C.-H. Chen, C.-M. Cheng: Could SFLASH be repaired? International Colloquium on Automata, Languages, and Programming, 2008. pp. 691-701.
8. J. Ding, J. E. Gower, D. S. Schmidt: Multivariate Public Key Cryptosystems. Springer, 2006.
9. J. Ding, D. S. Schmidt: Rainbow, a new multivariate polynomial signature scheme. ACNS 2005, LNCS vol. 3531, pp. 164-175. Springer, 2005.
10. J.C. Faugère: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra 139, pp. 61-88 (1999).
11. M. R. Garey and D. S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company 1979.
12. A. Hülsing, J. Rijneveld, S. Samardjiska, P. Schwabe: From 5-pass MQ-based identification to MQ-based signatures. Cryptology ePrint Archive: Report 2016/708
13. A. Juels, M. Luby, R. Ostrovsky: Security of Blind Digital Signatures. CRYPTO 1997, LNCS vol. 1294, pp. 150 - 164. Springer 1997.
14. D. Kravitz: Digital Signature Algorithm. US patent 5231668 (July 1991).
15. A. Kipnis, L. Patarin, L. Goubin: Unbalanced Oil and Vinegar Schemes. EUROCRYPT 1999, LNCS vol. 1592, pp. 206–222. Springer, 1999.
16. T. Matsumoto, H. Imai: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. EUROCRYPT 1988. LNCS vol. 330, pp. 419-453. Springer, 1988.
17. D. Goodin: NSA preps quantum-resistant algorithms to head off crypto-apocalypse. http://arstechnica.com/security/2015/08/nsa-preps-quantum-resistant-algorithms-to-head-off-crypto-apocolypse/.
18. National Institute of Standards and Technology: Report on post-quantum Cryptography. NISTIR draft 8105, http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf.
19. J. Patarin, N. Courtois, and L. Goubin. "Quartz, 128-bit long digital signatures." Cryptographers' Track at the RSA Conference. Springer Berlin Heidelberg, 2001.
20. A. Petzoldt, S. Bulygin, J. Buchmann: A Multivariate based Threshold Ring Signature Scheme. Appl. Algebra Eng. Commun. Comput. 24(3-4); 255-275 (2012).

21. A. Petzoldt, S. Bulygin, J. Buchmann: Selecting Parameters for the Rainbow Signature Scheme. PQCrypto 2010, LNCS vol. 6061, pp. 218-240. Springer, 2010.
22. A. Petzoldt, S. Bulygin, J. Buchmann: CyclicRainbow - A Multivariate Signature Scheme with a Partially Cyclic Public Key. INDOCRYPT 2010, LNCS vol. 6498, pp. 33-48. Springer, 2010.
23. A. Petzoldt, S. Bulygin, J. Buchmann: Fast Verification for Improved Versions of the UOV and Rainbow Signature Schemes. PQCrypto, LNCS vol. 7932, pp. 188-202. Springer, 2013.
24. A. Petzoldt, M.S. Chen, B.Y. Yang, C. Tao, J. Ding: Design Principles for HFEv-based Signature Schemes. ASIACRYPT 2015 - Part 1, LNCS vol. 9452, pp. 311-334.Springer, 2015.
25. R. L. Rivest, A. Shamir, L. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM 21 (2), pp. 120-126 (1978).
26. M. Rückert: Lattice-Based Blind Signatures. ASIACRYPT 2010 , LNCS vol. 6477, pp. 413-430. Springer, 2010.
27. SageMath, the Sage Mathematics Software System (Version 7.1), The Sage Developers, 2016, `http://www.sagemath.org`.
28. K. Sakumoto, T. Shirai, H. Hiwatari: Public-Key Identification Schemes Based on Multivariate Quadratic Polynomials. CRYPTO 2011, LNCS vol. 6841, pp. 706 - 723. Springer, 2011.
29. P. Shor: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM J. Comput. 26 (5), pp. 1484 - 1509 (1997).
30. E. Thomae, C. Wolf: Solving Underdetermined Systems of Multivariate Quadratic Equations Revisited. PQCrypto 2012, LNCS vol. 7293, pp. 156–171. Springer, 2012.
31. T. Yasuda, X. Dahan, Y-J Huang, T. Takagi, K. Sakurai: MQ Challenge: Hardness Evaluation of Solving Multivariate Quadratic Problems. IACR Cryptology ePrint Archive 2015 (2015): 275.
32. B.Y. Yang, J.M. Chen: Building secure tame-like multivariate public-key cryptosystems.: The new TTS. CHES 2004, LNCS vol. 3156, pp. 371- 385. Springer, 2004.

6.3 Public Key Compression for Constrained Linear Signature Schemes

Publication data

Ward Beullens and Bart Preneel and Alan Szepieniec, “Public Key Compression for Constrained Linear Signature Schemes” *Selected Areas in Cryptography - SAC 2018 - 28th International Conference, University of Calgary, Alberta, August 15-17, 2018, Revised Selected Papers.*, 2018. *(Page numbers not known yet.)*

Contributions

Contributing author

Notes

This paper supersedes a previous collaboration with the same co-authors, “MQ Signatures for PKI” [132] which is essentially the same idea but specifically for MQ signature schemes. Incidentally, that paper became the basis for a submission to the NIST competition under the name “DualModeMS” [75] by researchers from Paris who generously coined the term “SBP transform”. The novelty with respect to the predecessor paper consists of the constrained-linear signature scheme formalism and the quantum random oracle model proof. Ward and I judged each other to have roughly equal contributions so we opted for alphabetical author listing here.

Public Key Compression for Constrained Linear Signature Schemes

Ward Beullens and Bart Preneel and Alan Szepieniec

imec-COSIC KU Leuven, Belgium
ward.beullens@esat.kuleuven.be, bart.preneel@esat.kuleuven.be,
alan.szepieniec@esat.kuleuven.be

Abstract. We formalize the notion of a *constrained linear trapdoor* as an abstract strategy for the generation of signature schemes, concrete instantiations of which can be found in MQ-based, code-based, and lattice-based cryptography. Moreover, we revisit and expand on a transformation by Szepieniec *et al.* [39] to shrink the public key at the cost of a larger signature while reducing their combined size. This transformation can be used in a way that is provably secure in the random oracle model, and in a more aggressive variant whose security remained unproven. In this paper we show that this transformation applies to any constrained linear trapdoor signature scheme, and prove the security of the first mode in the quantum random oracle model. Moreover, we identify a property of constrained linear trapdoors that is sufficient (and necessary) for the more aggressive variant to be secure in the quantum random oracle model. We apply the transformation to an MQ-based scheme, a code-based scheme and a lattice-based scheme targeting 128-bits of post quantum security, and we show that in some cases the combined size of a signature and a public key can be reduced by more than a factor 300.

Keywords: digital signatures, post-quantum, quantum random oracle model, key size reduction

1 Introduction

Trapdoor functions are an important tool in public key cryptography due to the computational asymmetry they bring about. On the one hand, the function is a proper cryptographic one-way function to anyone who is ignorant of the secret trapdoor information; but on the other hand, anyone who does know this trapdoor information can use it to find inverse images quickly.

The case of *surjective* trapdoor functions is especially interesting for generating *digital signature schemes.* A cryptographic hash function maps a message of any size to a random point in the trapdoor function's output space. An inverse of this point under the trapdoor function, or *signature*, testifies to the involvement of the trapdoor information, or *secret key*, in its generation. This testimony ensures the target property of *non-repudiation of origin*: the secret key holder cannot deny generating the signature at a later date.

Since their inception in the seminal paper by Diffie and Hellman [10], various digital signature schemes have been deployed whose security is based on the hardness of integer factorization [35] and the discrete logarithm problem [36,30]. However, the advent of quantum computers threatens the security of these signature schemes because both hard problems are solved efficiently by Shor's quantum algorithm [37]. This ultimatum drives the need to design, develop and deploy so-called *post-quantum* cryptosystems, *i.e.*, cryptography that can be run on classical hardware but promises to resist attacks by quantum computers.

Even though the RSA trapdoor is broken by quantum computers, the hash-and-sign construction that RSA signatures are based on seems to survive the transition to post-quantum cryptography. To achieve post-quantum secure signature schemes it suffices to exchange the underlying trapdoor for one that has the desired security against quantum adversaries. There is no shortage of trapdoor-based signature schemes based on the MQ problem [21,11,34], coding theory [8,9], or lattices [15,3,27].

Unfortunately, the public keys in these schemes are prohibitively large, measurable in hundreds of kilobytes if not megabytes. In contrast, post-quantum signature schemes derived from zero-knowledge proofs require only a one-way function whose selection can be random or might as well be determined by a short seed and an implicit pseudorandom generator. Signature schemes based on zero-knowledge proofs tend to exchange tiny public keys for prohibitively large signatures [38,7,23,18], and moreover require complicated and expansive non-interactivity transforms to retain security against quantum attackers [40]. Although provable security in the case of hash-based signature schemes is much more straightforward, this family of constructions follows the same pattern: tiny public keys but huge signatures [5,4].

Szepieniec, Beullens and Preneel offer an alternative to the dilemma between large public keys or large signatures [39], motivated by the desire to minimize the combined size of public key and signature. This minimization is particularly important in the context of public key infrastructure (PKI) where a chain of signatures and public keys is transmitted in order to authenticate a message with respect to a pre-shared root public key. The construction of Szepieniec *et al.* applies specifically to MQ trapdoors and relies on the observation that verifying a couple of random linear combinations of the public key's polynomial equations can be as good as verifying all of them. The coefficients of this linear combination are determined as a function of the produced signature, and the combination itself is transmitted along with this signature in addition to information authenticating its link to the public key. This transformation reduces the size of public key plus that of the signature by roughly a factor three whilst provably retaining security in the random oracle model; and by a much larger factor at the expense of a heuristic security argument.

This article expands on the paper of Szepieniec *et al.* in several ways. We observe that this transformation also applies to other post-quantum trapdoor signature schemes, most notably code-based and lattice-based trapdoors. From a general perspective, these three hard problems are variations on a common theme, which we call *constrained linear signature schemes.* This commonality allows a generic presentation of the transformation. The security proofs of Szepieniec *et al.* only work in the classical random oracle model. However, security proofs that purport to defend against quantum adversaries should additionally hold in the *quantum random oracle model*, which our proof does. Moreover, we identify a necessary and sufficient security property, called (σ, r)-hash-and-sign-security ((σ, r)-HSS), that a constrained linear signature scheme must have in order for the more aggressive parameter choices of Szepieniec *et al.* to be provably secure. This leads to an improved understanding of the security of instantiations of this construction, which includes the DualModeMS submission of Faugère *et al.* [12] to the NIST PQC standardization project [29]. To showcase the key size improvements that can be achieved with the transformation, we apply the transformation to a lattice-based, code-based and multivariate constrained linear signature scheme with parameters targeting 128 bits of security against quantum computers.

2 Preliminaries

Random oracle model. We use a hash function in our construction. For the purpose of proving security we model it by a *random oracle*, which is a random function $\mathsf{H} : \{0,1\}^* \rightarrow \{0,1\}^\kappa$ with a fixed output length, typically equal to the security parameter. If necessary, the random oracle's output space can be lifted to any finite set X. We use subscripts to differentiate the random oracles associated with different output spaces. A security proof relying on the modelling of hash function as random oracles is said to hold in the *random oracle model.* When quantum adversaries are considered, the security proofs should allow for superposition queries to the random oracle [6]; a security proof with this property is said to hold in the *quantum random oracle model.*

Trapdoor functions. A trapdoor function is a function that can be efficiently computed in one direction, but for which it is hard to compute preimages *unless* by someone who knows a secret piece of information called the *trapdoor.* We associate three algorithms to a trapdoor function family:

- $\mathsf{GenTrapdoor}$ takes a security parameter as input and outputs a trapdoor function f and a trapdoor t.
- $\mathsf{Evaluate}$ takes a description of the trapdoor function f and an argument x as input, and returns the evaluation of f at x. In the rest of the paper, we simply write this as $f(x)$.
- Invert takes the function f, the trapdoor t and an image y as input, and outputs a value x such that $f(x) = y$.

Signature scheme. A public key signature scheme is defined as a triple of polynomial-time algorithms ($\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}$). The probabilistic key generation algorithm takes the security level κ (in unary notation) and produces a secret and public key: $\mathsf{KeyGen}(1^\kappa) = (sk, pk)$; the signature generation algorithm produces a signature: $s = \mathsf{Sign}(\mathsf{sk}, m) \in \{0,1\}^*$. The verification algorithm takes the public key, the message and the signature and decides if the signature is valid: $\mathsf{Verify}(pk, m, s) \in \{0,1\}$; we refer to these outputs as "reject" and "accept", respectively. The signature scheme is *correct* if signing a message with the secret key produces a valid signature under the matching public key:

$$(\mathsf{KeyGen}(1^\kappa) \Rightarrow (sk, pk)) \quad \Longrightarrow \quad \forall m \in \{0,1\}^* \,.\, \mathsf{Verify}\left(pk, m, \mathsf{Sign}(sk, m)\right) = 1 \ .$$

Here and elsewhere we use $\Rightarrow$ to denote the event of the probabilistic algorithm on the left hand producing the output on the right hand, and $\Longrightarrow$ to denote logical implication.

Security is defined with respect to the Existential Unforgeability under Chosen Message Attack (EUF-CMA) game of Goldwasser *et al.* [17]. The adversary A is allowed to make a polynomial number of queries $m_i, i \in \{1, \ldots, q\}, q \leq \kappa^c$ for some c, which the challenger signs using the secret key and sends back: $s_i \leftarrow \text{Sign}(\mathsf{sk}, m_i)$. At the end of the game, the adversary must produce a pair of values (m', s') where m' was not queried before: $m' \notin \{m_i\}_{i=1}^q$. The adversary wins if $\mathsf{Verify}(pk, m', s') = 1$. In the game below, the Iverson brackets $[\![\cdot]\!]$ return 0 if the expression is False or 1 if it is True.

```
Game EUF-CMA
$sk, pk \leftarrow \mathsf{KeyGen}(1^\kappa)$
$\mathcal{M} \leftarrow \varnothing$
define $\mathsf{S}(m)$ as
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
    return $\mathsf{Sign}(sk, m)$
end definition
$(m, s) \leftarrow \mathsf{A}^{\mathsf{S}}(pk)$
return $[\![\mathsf{Verify}(pk, m, s) = \mathsf{True} \wedge m \notin \mathcal{M}]\!]$
```

We define the insecurity function $\mathsf{InSec}^{\text{EUF-CMA}}_{\mathsf{scheme}}(Q_\mathsf{S}; t)$ as the maximum winning probability across all quantum adversaries that run in time t and that make at most Q_S signature queries.

Hash-and-sign signature schemes. Given a trapdoor function family and a hash function H that hashes arbitrary messages to elements in the range of the trapdoor functions we can use the hash-and-sign construction to build a (not necessarily secure) signature scheme. The key generation algorithm simply calls the $\mathsf{GenTrapdoor}$ function to get (f, t). The public key is then the description of f, and the trapdoor t is the private key. To sign a message m, the signer uses his trapdoor t to produce a preimage s for $\mathsf{H}(m)$. This preimage is the signature for m. Lastly, to verify the validity of a signature the verifier computes $\mathsf{H}(m)$, uses the public key to evaluate f at s and checks if $f(s) = \mathsf{H}(m)$.

Merkle tree. A Merkle tree [26] is a balanced binary tree whose root authenticates a list of data items which are contained in the leaves. Every non-leaf node, including the root, has a value equal to the hash of the concatenation of its two children. A leaf can be proven to be a member of the tree by tracing a path from the leaf to the root and listing all siblings of nodes on that path: every step can be verified by computing one hash. We associate three algorithms with a Merkle tree:

- CalculateMerkleRoot takes a list of leaf items, computes the entire Merkle tree, and returns its root.
- OpenMerklePath takes a list of leaf nodes and an index, and outputs its authentication path: the list of all siblings of nodes on the path from the indicated leaf node to the root.
- VerifyMerklePath takes an index, a leaf node, a Merkle path, and a root, and decides whether the leaf node is a member of the tree with the given root.

3 Trapdoor-Based Signature Schemes

3.1 MQ Trapdoors

Multivariate quadratic (MQ) trapdoor functions date back to the C* scheme of Matsumoto and Imai [25], which has since given rise to a number of viable candidates including HFE$_v^-$ [32], UOV [21] and Rainbow [11]. The idea is to compose a special quadratic map $\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ with two linear transforms, $T \in \mathsf{GL}_m(\mathbb{F}_q)$ and $S \in \mathsf{GL}_n(\mathbb{F}_q)$ to obtain the public key $\mathcal{P} = T \circ \mathcal{F} \circ S$. A vector $\mathbf{s} \in \mathbb{F}_q^n$ that represents an assignment to the variables, is a valid signature for the document $d \in \{0,1\}^*$ whenever

$$\mathcal{P}(\mathbf{s}) = \mathsf{H}(d) \ . \tag{1}$$

In order to find $\mathbf{s}$, the signer computes $\mathbf{z} = \mathsf{H}(d)$, $\mathbf{y} = T^{-1}\mathbf{z}$, uses the special structure of $\mathcal{F}$ to sample an inverse $\mathbf{x}$ such that $\mathcal{F}(\mathbf{x}) = \mathbf{y}$, and then computes $\mathbf{s} = S^{-1}\mathbf{x}$.

We focus on the Rainbow submission to the NIST PQC project [29], where the parameter set ($q = 256, v = 68, o_1 = 36, o_2 = 36$) is proposed. In this case, $n = v + o_1 + o_2 = 140$ and $m = o_1 + o_2 = 72$. While the proposal does not employ Petzoldt's compression trick [33] we note that it is possible in principle, in which case $v(v+1)/2 + vo_1$ columns of the public Macaulay matrix are set as the output of a PRG expanding a seed of 32 bytes.[1] Allocating five bits per field element, we obtain signatures of 140 bytes and public keys of 356.9 kB. Without Petzoldt's compression trick the public key is 694.0 kB.

3.2 Code-Based Trapdoors

The first code-based signature scheme was proposed by Courtois, Finiasz and Sendrier (CFS) [8]; it relies on the difficulty of finding a low Hamming weight

[1] In fact, Petzoldt manages to fix more elements of the public key's Macaulay matrix, but as these elements are not arranged into columns they are incompatible with our compression technique.

word associated with a given syndrome. The public key in such a signature scheme is a parity check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$. A signature $(\mathbf{s}, i) \in \mathbb{F}_2^{1\times n} \times \mathbb{Z}$ on a document $d \in \{0,1\}^*$ consists of an error vector and an index; it is valid when the error vector has Hamming weight at most t and syndrome equal to the hash of the document concatenated with the index i. The index i can be thought of as selecting a different hash function. Formulaically:

$$H\mathbf{s}^{\mathsf{T}} = \mathsf{H}(d\|i) \quad \text{and} \quad \mathsf{HW}(\mathbf{s}) \leq t \ . \tag{2}$$

By our calculations, a 128-bit post-quantum security level is achieved with the parameter set $m = 26$, $t = 15$ and thus $n = 2^m = 2^{26}$ and $n - k = tm = 390$. At this point the public key is 3.05 GB but the signatures are 390 bits. We refer to Appendix A for a derivation of these parameters. We choose not to consider the question whether the cryptosystem is practically usable with these parameters and instead focus on the obtained compression factor. The CFS scheme is used as a generic stand-in for code-based signature schemes using the hash-and-sign paradigm and relying on the hardness of syndrome decoding.

3.3 Lattice-Based Trapdoors

A first trapdoor-based signature schemes from lattices was proposed by Goldreich, Goldwasser and Halevi (GGH) at Crypto '97 [16]. The signatures of this scheme leak information about the private key, and the scheme was broken by Nguyen and Regev [31]. Gentry, Peikert and Vaikuntanathan [15] showed how to sample signatures that do not leak information and constructed a provably secure signature scheme. Later improvements by Alwen and Peikert [3] and by Micciancio and Peikert [27] make the scheme more efficient. The main idea is the same in all schemes: the public key is a matrix $A \in \mathbb{F}_q^{n\times m}$ with large coefficients but such that there exists another matrix $S \in \mathbb{Z}^{m\times m}$ with small coefficients with $AS = 0 \bmod q$. In order to generate a signature for a document $d \in \{0,1\}^*$, the signer uses the secret key S to obtain a small-coefficient vector $\mathbf{z} \in \mathbb{Z}^m$. It is a valid signature whenever

$$A\mathbf{z} = \mathsf{H}(d) \bmod q \quad \text{and} \quad \|\mathbf{z}\|_2 \leq \beta \ , \tag{3}$$

for some length bound $\beta \in \mathbb{R}_{>0}$.

Using the methodology of [28], and the estimator for the concrete hardness of the SIS problem of Albrecht et al. [1], we choose parameters for the scheme of [27] that achieves 128 bits of security. This results in the parameters $n = 321, q = 2^{26} - 5, m = 16692$ and $\beta = 112296$, a public key of $n \times m \times 26$ bits $= 16.6$ MB, and signatures of $\lceil \log_2(\beta) \rceil \times m$ bits $= 34.6$ KB. We chose q to be prime as this is required for our security proof to work. The first half of the matrix A can be chosen randomly, so we can fix this part with a PRG to cut the size of the public key in half.

3.4 A Unifying View

The above three signature schemes can be thought of as variations on a common theme. These schemes are all hash-and-sign signature schemes with a linear trapdoor function $f : \mathbb{F}_q^\ell \to \mathbb{F}_q^k$, but with f restricted to a domain defined by a nonlinear constraint function $\mathsf{nc} : \mathbb{F}_q^\ell \to \{\mathsf{True}, \mathsf{False}\}$. We call these trapdoor functions **constrained linear trapdoor functions**, and if they are used in a hash-and-sign construction, we call the resulting signature scheme a **constrained linear signature scheme**.

For all the constrained linear signature schemes the public key is a matrix $M \in \mathbb{F}_q^{k\times\ell}$ with $k < \ell$ which represents the trapdoor function f and a signature is represented by a vector $\mathbf{s} \in \mathbb{F}_q^\ell$. A signature is valid if $M\mathbf{s}$ is equal to a target $\mathbf{t} \in \mathbb{F}_q^k$, which is the evaluation of a hash function at a document, and if the vector $\mathbf{s}$ also satisfies the constraint nc. Symbolically:

$$\mathsf{Verify}(sk, m, \mathbf{s}) = 1 \iff M\mathbf{s} = \mathbf{t} = \mathsf{H}(m) \wedge \mathsf{nc}(\mathbf{s}) = \mathsf{True} \ .$$

In the case of lattice-based trapdoors, the signature is valid only if $\mathbf{s}$ is a short vector. In the case of code-based trapdoors, it is valid only if the Hamming weight of $\mathbf{s}$ is low. And in the case of MQ trapdoors, the matrix M is the coefficient matrix (or Macaulay matrix) of the quadratic polynomial map $\mathcal{P}$ and the signature $\mathbf{s}$ must be factorizable as a vector of products of n variables: $\mathbf{s}^\mathsf{T} = (x_1^2, x_1x_2, \ldots, x_n^2)$. Formally, we capture this difference between MQ, code-based, and lattice-based trapdoors with the nonlinear constraint nc, namely by defining for

- code-based trapdoors: $\mathsf{nc}(\mathbf{s}) = \mathsf{True} \Leftrightarrow \textsc{hw}(\mathbf{s}) \le t$;
- lattice-based trapdoors: $\mathsf{nc}(\mathbf{s}) = \mathsf{True} \Leftrightarrow \|\mathbf{s}\|_2 \le \beta$;
- MQ trapdoors: $\mathsf{nc}(\mathbf{s}) = \mathsf{True} \Leftrightarrow \exists\, x_1, \ldots, x_n \in \mathbb{F}_q \,.\, \mathbf{s}^\mathsf{T} = (x_1^2, x_1x_2, \ldots, x_n^2)$.

3.5 Additional security properties

We say that a surjective trapdoor function f is one-way (OW) if it is hard to find a preimage for a randomly chosen output, and we say that f is hash-and-sign secure (HSS) if using the trapdoor function f in the hash-and-sign construction leads to a signature scheme that is EUF-CMA secure. If f is a constrained linear trapdoor function we can define stronger versions of the OW and HSS security properties that will be useful for the security analysis of the transformation.

(σ, r)-one-wayness. For any two non-negative integers $\sigma > r$ we define (σ, r)-one-wayness and (σ, r)-hash-and-sign security. To break (σ, r)-one-wayness, an adversary has to find σ preimages $\mathbf{x}_1, \ldots, \mathbf{x}_\sigma \in \mathbb{F}_q^\ell$ for σ vectors $\mathbf{y}_1, \ldots, \mathbf{y}_\sigma \in \mathbb{F}_q^k$. However, the adversary is allowed to make mistakes in each of the σ preimages it produces, as long as the errors $f(\mathbf{x}_i) - \mathbf{y}_\mathbf{i}$ are contained in a vector space of dimension r. The $(1, 0)$-one-wayness property is identical to the one-wayness property, because the adversary only needs to find a preimage for one target and it is not allowed to make any mistakes.

The (σ, r)-OW property is a generalization of the AMQ problem introduced in [39]; an MQ trapdoor $\mathcal{P}$ is (σ, r)-one-way precisely if the Approximate MQ problem with σ targets and rank r is hard for the map $\mathcal{P}$.

(σ, r)-hash-and-sign security. We also define a (σ, r)-variant of the HSS property. The security game behind this property is similar to the EUF-CMA game of the hash-and-sign signature scheme induced by f. To break this property, an adversary has to come up with a message m and σ 'signatures' $\mathbf{s}_1, \cdots, \mathbf{s}_\sigma$ such that the errors $f(\mathbf{s}_i) - \mathsf{H}(m||i)$ are contained in a a subspace of dimension r. The adversary can query a signing oracle S any (polynomially bounded) number of times. When given a message m', this signing oracle uses the trapdoor to produce preimages for $\mathsf{H}(m'||1), \cdots, \mathsf{H}(m'||\sigma)$ and returns these σ preimages. The adversary loses the game if it returns a message m for which it has queried the signing oracle, as is the case for the familiar EUF-CMA game.

We define the insecurity function $\mathsf{InSec}_f^{(\sigma,r)-\mathsf{HSS}}(Q_\mathsf{S}, Q_\mathsf{H}; t)$ as the maximal winning probability of an adversary that plays the (σ, r)-HSS game of f, that makes Q_S queries to the signing oracle, Q_H queries to the random oracle and that runs in time t. The $(1, 0)$-HSS property is equivalent to the HSS property.

Remark 1. If f is a *collision-resistant preimage-sampleable trapdoor function* (as is the case for some lattice-based trapdoor functions), the one-wayness of f can be reduced tightly to its hash-and-sign security and so OW and HSS are equivalent [15, Prop. 6.1]. Under the same assumption on f, the security proof of [15] can be modified to prove that (σ, r)-OW and (σ, r)-HSS are equivalent for all $\sigma > r \geq 0$.

4 Construction

4.1 Description

This section describes the transform of Szepieniec *et al.* but adapted to apply generically to constrained linear signature schemes. The parameters for the transformation are:

- $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$, the constrained linear signature scheme to start from. We denote the hash function used in the verification algorithm by H_1 and the nonlinear constraint by nc.
- τ, the number of leaves in the Merkle tree.
- e, the extension degree of $\mathbb{F}_{q^e}$, which is the field over which the error-correcting code is defined. This value is constrained by $q^e \geq \tau$.
- ϑ, the number of Merkle paths that are opened with each new signature.
- σ, the number of signatures of the original signature scheme that is included in each signature of the new scheme.

```
Game $(\sigma, r)$-OW
$(f, t) \leftarrow \mathsf{GenTrapdoor}(1^\kappa)$
$\mathbf{y}_1, \dots, \mathbf{y}_\sigma \xleftarrow{\$} \mathbb{F}_q^k$
$\mathbf{x}_1, \dots, \mathbf{x}_\sigma \leftarrow \mathsf{A}(f, \mathbf{y_1}, \dots, \mathbf{y}_\sigma)$
return $[\![\dim(\langle f(\mathsf{x}_i) - \mathsf{y}_i \rangle_i) \leq r]\!]$
```

```
Game $(\sigma, r)$-HSS
$(f, t) \leftarrow \mathsf{GenTrapdoor}(1^\kappa)$
$\mathcal{M} \leftarrow \varnothing$
define $\mathsf{S}(m)$ as
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
    for $i$ from 1 to $\sigma$ do
        $\mathbf{s}_i \leftarrow \mathsf{Invert}(f, t, \mathsf{H}(m||i))$
    end for
    return $\mathbf{s}_1, \dots, \mathbf{s}_\sigma$
end definition
$m, \mathbf{s}_1, \dots, \mathbf{s}_\sigma \leftarrow \mathsf{A}^{\mathsf{H},\mathsf{S}(\cdot)}(f)$
$d = \dim(\langle f(\mathbf{s}_i) - \mathsf{H}(m||i) \rangle_i)$
return $[\![(d \leq r) \wedge (m \notin \mathcal{M})]\!]$
```

Fig. 1: The security game of the $(\sigma, r)-\mathsf{OW}$ property (left) and of the $(\sigma, r)-\mathsf{HSS}$ property (right).

$$\begin{tikzcd}
\text{OW} & (\sigma, r)\text{-OW} \arrow[l, Rightarrow] \\
\text{HSS} \arrow[u, Rightarrow] & (\sigma, r)\text{-HSS} \arrow[l, Rightarrow] \arrow[u, Rightarrow]
\end{tikzcd}$$

Fig. 2: Security properties of constrained linear trapdoor functions, and implications between them.

- H_2, a hash function that outputs a α-by-k matrix over $\mathbb{F}_q$.
- H_3, a hash function that outputs a set of ϑ numbers between 1 and τ.
- H_4, a hash function used for building a Merkle tree.

The transformation outputs a new signature scheme ($\mathsf{NEW.KeyGen}$, $\mathsf{NEW.Sign}$, $\mathsf{NEW.Verify}$) with a smaller public key but larger signatures.

Random Linear Combinations. A signature of the new signature scheme consists of σ signatures of the original signature scheme, along with some information to verify them. The ith signature is obtained by using the signature generation algorithm of the original contrained-linear signature scheme to sign $d\|i$. It is not necessary to communicate the entire public key $M \in \mathbb{F}_q^{k \times \ell}$. Rather, it suffices to transmit a few random linear combinations of its rows. Therefore, part of the new signature consists of a matrix T that is equal to RM, where R is drawn uniformly at random from the space of $\alpha \times k$ matrices. Instead of checking whether $M\mathbf{s}_i = \mathsf{H}_1(d\|i)$, the verifier can now check wheter $T\mathbf{s}_i = R\mathsf{H}_1(d\|i)$.

Obviously, if all signatures are valid, then the latter equations will also be satisfied for any matrix R. Conversely, if at least one signature is invalid, *i.e.*, $M\mathbf{s}_i \neq \mathsf{H}_1(d\|i)$ for some i, then the probability that $RM\mathbf{s} = R\mathsf{H}_1(d\|i)$ is at most $q^{-\alpha}$. By choosing α large enough, the probability of accepting an invalid signature can be made arbitrarily small.

Determining *R*. In order for the above argument to work, R must be chosen independently from $\mathbf{s} = \mathbf{s}_1\| \cdots \|\mathbf{s}_\sigma$. Therefore, we determine R with a hash function as $R = \mathsf{H}_2(d\|\mathbf{s}_1\| \cdots \|\mathbf{s}_\sigma)$ to ensure that a forger cannot use knowledge about R in his choice of the $\mathbf{s}_i$.

Verifying *T*. An attacker can present the verifier with a signature containing a matrix T which is totally unrelated to the matrix M. How can the verifier be sure that the matrix T that is included in the signature, is really equal to RM with $R = \mathsf{H}_2(d\|\mathbf{s}_1\| \cdots \|\mathbf{s}_\sigma)$? We solve this problem with a probabilistic test based on an $\mathbb{F}_q$-linear error correcting code. This is a code whose alphabet consists of the elements of a finite field $\mathbb{F}_q$, with the property that any $\mathbb{F}_q$-linear combination of codewords is again a codeword. We work with Reed-Solomon Codes[2] over $\mathbb{F}_{q^e}$ with message length $L = \lceil \ell/e \rceil$ (we pack e elements of $\mathbb{F}_q$ into each symbol), codeword length τ and minimal codeword distance $D = \tau - L$. We use $\mathsf{Enc} : \mathbb{F}_{q^e}^{a \times L} \to \mathbb{F}_{q^e}^{a \times \tau}$ to denote the operation of encoding the rows of a matrix.

In the key generation phase, we compute $E = \mathsf{Enc}(M)$. Then we commit to this matrix E by building a Merkle tree whose leaves contain the columns of E, which are denoted by e_i for $i \in \{1, \ldots, \tau\}$. The new public key is the root of this tree. If $T = RM$, then by $\mathbb{F}_q$-linearity of the error correcting code, we have that $\mathsf{Enc}(T)$ is equal to $R\mathsf{Enc}(M) = RE$. Conversely, if $T \neq RM$, then $\mathsf{Enc}(T)$ and RE differ in at least one row. These rows are different codewords, so they differ in at least D of the τ symbols. To verify that $T = RM$, we now select ϑ columns $e_{b_1}, \cdots, e_{b_\vartheta}$ of E with the hash function H_3 and we check whether the b_i-th column of T agrees with Re_{b_i} for all i in $1, \cdots, \vartheta$. If T is not equal to RM, this will go undetected with a probability of at most $(\frac{L}{\tau})^\vartheta$.

Pseudocode. Algorithms 1, 2 and 3 present pseudocode for the new signature scheme (NEW.KeyGen, NEW.Sign, NEW.Verify) obtained from transforming the old constrained-linear signature scheme (KeyGen, Sign, Verify).

Key and signature sizes. For a post-quantum security level of κ bits, the new public key is 2κ bits in size, as it represents the Merkle root. The new signature consists of σ old signatures, α linear combinations of the rows of M (each one

[2] While the original description of the transformation used MAC-polynomials, we think it is better to describe the same transformation it in the language of Reed-Solomon error correcting codes.

```
Algorithm NEW.KeyGen

input: $1^\kappa$ — security level (in unary)
       random coins
output: $root$ — A public key
        $(sk, M)$ — A corresponding secret key

$(sk, M) \leftarrow \mathsf{KeyGen}(1^\kappa)$
$E \leftarrow \mathsf{Enc}(M)$                                   ▷ Encode $M$ row by row.
$root \leftarrow \mathsf{CalculateMerkleRoot}(e_1, \cdots, e_\tau)$      ▷ Build tree on columns of $E$
return $(root\ , (sk, M))$
```

Alg. 1: The key generation algorithm

of which consists of ℓ field elements of size $\lceil \log_2 q \rceil$ bits), ϑ columns of $\mathsf{Enc}(M)$ (each one of which consists of k field elements of $e \times \lceil \log_2 q \rceil$ bits), and ϑ Merkle paths of consisting of $\log_2 \tau$ hash images of 2κ bits each. Put all together, we have

$$|\mathsf{NEW}.\mathit{signature}| = \sigma|\mathsf{OLD}.\mathit{signature}| + (\alpha\ell + \vartheta k e) \times \lceil \log_2 q \rceil + 2\vartheta\kappa \times \log_2 \tau \ . \quad (4)$$

The old signatures can be represented as ℓ field elements but in some cases a more concise encoding is possible. For instance, CFS signatures require only the positions of the 1-bits, and MQ signatures require only an assignment to the variables from which the vector of quadratic monomials can be derived.

4.2 Security

Before we present the security statement and its proof, we need to introduce a pair of security games that will be important for our security analysis. In particular, we need hash functions that are one-way and second-preimage resistant, in both cases with respect to multiple targets. Both games are formalized with respect to a hash function H that is randomly selected from a hash function family $\mathcal{H}$. We follow the formalisms of Hülsing *et al.* [20].

- In the *single-function, multiple-target one-wayness* (SM-OW) game, the adversary is given a list of target outputs and it wins if it can produce a single input that maps to any one of the outputs. We write $\mathsf{InSec}^{\text{SM-OW}}_{\mathsf{H},P}(Q)$ to denote the maximum success probability across all adversaries that make at most Q queries and with respect to the hash function family $\mathcal{H}$ and where P is the number of target outputs.
- In the *single-function, multiple-target second-preimage resistance* (SM-SPR) game, the adversary is given a list of inputs and it wins if it can produce a second preimage that maps to the same output as any one of the input preimages. We write $\mathsf{InSec}^{\text{SM-SPR}}_{\mathsf{H},P}(Q)$ to denote the maximum success probability across all adversaries that make at most Q queries and with respect to the hash function family $\mathcal{H}$ and where P is the number of input preimages.

```
Algorithm NEW.Sign

input: $d$ — A document to sign
       $(sk, M)$ — A private key
output: $(\mathbf{s}_1, \cdots, \mathbf{s}_\sigma, T, v_{b_1}, \cdots, v_{b_\vartheta}, paths)$ — A signature for $d$

for $i$ from 1 to $\sigma$ do
    $\mathbf{s}_i \leftarrow \mathsf{Sign}(d \| i, sk)$
end for
$R \leftarrow \mathsf{H}_2(d \| \mathbf{s}_1 \| \cdots \| \mathbf{s}_\sigma)$
$T \leftarrow RM$
$E \leftarrow \mathsf{Enc}(M)$                                  ▷ Encode $M$ row by row.
$b_1, \cdots, b_\vartheta \leftarrow \mathsf{H}_3(d \| \mathbf{s}_1 \| \cdots \| \mathbf{s}_\sigma \| T)$
$paths \leftarrow$ empty list
for $i$ from 1 to $\vartheta$ do
    $paths.\mathsf{append}(\mathsf{OpenMerklePath}(e_1, \cdots, e_\tau, b_i))$
end for
return $(\mathbf{s}_1, \cdots, \mathbf{s}_\vartheta, T, e_{b_1}, \cdots, e_{b_\vartheta}, paths)$
```

Alg. 2: The signature generation algorithm.

```
Game SM-OW

$\mathsf{H} \xleftarrow{\$} \mathcal{H}$
for $i$ from 1 to $P$ do
    $M_i \xleftarrow{\$} \{0,1\}^m$
    $Y_i \leftarrow \mathsf{H}(M_i)$
end for
$M' \leftarrow \mathsf{A}^{\mathsf{H}}(Y_1, \ldots, Y_P)$
return $[\![\exists i \,.\, \mathsf{H}(M') = Y_i]\!]$
```

```
Game SM-SPR

$\mathsf{H} \xleftarrow{\$} \mathcal{H}$
for $i$ from 1 to $P$ do
    $M_i \xleftarrow{\$} \{0,1\}^m$
end for
$M' \leftarrow \mathsf{A}^{\mathsf{H}}(M_1, \ldots, M_P)$
return $[\![\exists i \,.\, \mathsf{H}(M') = Y_i \wedge M' \neq M_i]\!]$
```

Hülsing *et al.* obtain values for these insecurity functions in the random oracle model, *i.e.* where H is drawn uniformly at random from the set of all functions from the given input space to the given output space. In the classical random oracle model we have

$$\mathsf{InSec}^{\text{SM-OW}}_{\mathsf{H},P}(Q) = \mathsf{InSec}^{\text{SM-SPR}}_{\mathsf{H},P}(Q) = \frac{(Q+1)P}{|\mathsf{range}(\mathsf{H})|} \quad . \tag{5}$$

In the quantum random oracle model, where the adversary is allowed $\hat{Q}$ quantum queries, we have

$$\mathsf{InSec}^{\text{SM-OW}}_{\mathsf{H},P}(\hat{Q}) = \mathsf{InSec}^{\text{SM-SPR}}_{\mathsf{H},P}(\hat{Q}) = \Theta\left(\frac{(\hat{Q}+1)^2 P}{|\mathsf{range}(\mathsf{H})|}\right) \quad . \tag{6}$$

Algorithm NEW.Verify

```
input: $d$ — document
       $(\mathbf{s}_1, \cdots, \mathbf{s}_\vartheta, T, v_{b_1}, \cdots, v_{b_\vartheta}, paths)$ — signature
       $root$ — public key
output: 1 if the signature is valid, 0 otherwise

$R \leftarrow \mathsf{H}_2(d \| \mathbf{s}_1 \| \cdots \| \mathbf{s}_\sigma)$
for $i$ from 1 to $\sigma$ do
    if $T\mathbf{s}_i \neq R\mathsf{H}_1(d \| i)$ or $\mathsf{nc}(\mathbf{s}_i) = \mathsf{False}$ then
        return 0
    end if
end for
$b_1, \cdots, b_\vartheta \leftarrow \mathsf{H}_3(d \| \mathbf{s}_1 \| \cdots \| \mathbf{s}_\sigma \| T)$
for $i$ from 1 to $\vartheta$ do
    if $\mathsf{Enc}(T)_{*,b_i} \neq Re_{b_i}$ then
        return 0
    end if
    if $\mathsf{VerifyMerklePath}(b_i, e_{b_i}, paths[i], root) = \mathsf{Fail}$ then
        return 0
    end if
end for
return 1
```

Alg. 3: The signature verification algorithm.

The SM-OW game does not quite capture one of the transitions in our security proof. The reason for this is that the adversary cannot be given a definite list of target output images because whether an output of the hash function is suitable for the adversary depends on the input of the hash function. We model this task by a new game, *marked element search* (MES), in which the adversary does not have a list of target outputs but a marking function $\mathsf{mark} : \mathsf{domain}(\mathsf{H}) \times \mathsf{range}(\mathsf{H}) \to \{0, 1\}$ that determines whether the pair (*input*, *output*) is suitable. We write $\mathsf{InSec}^{\mathrm{MES}}_{\mathsf{H},\mathsf{mark}}(Q)$ to denote the maximum success probability across all adversaries that make at most Q queries to the hash oracle in the MES game. In the quantum random oracle model this notion is reducible to SM-OW.

Game MES

```
$\mathsf{H} \xleftarrow{\$} \mathcal{H}$
$M \leftarrow \mathsf{A}^{\mathsf{H}}()$
return $\mathsf{mark}(M, \mathsf{H}(M))$
```

Proposition 1 ($\mathsf{SM\text{-}OW} \leq \mathsf{MES}$)**.** *In the (quantum) random oracle model, we have that for any marking function* mark *with* $P = \mathsf{max}_X \, |\{Y \,|\, \mathsf{mark}(X,Y) = 1\}|$,

$$\mathsf{InSec}^{\mathrm{MES}}_{\mathsf{H},\mathsf{mark}}(Q) \leq \mathsf{InSec}^{\mathrm{SM\text{-}OW}}_{\mathsf{H},P}(Q) \ . \tag{7}$$

Proof. We show an algorithm, $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ in the SM-OW game, that simulates a given algorithm $\mathsf{A}_{\mathsf{MES}}$ for the MES game with marking function mark, and wins with at least the same probability. The input of $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ is a list of P images $\{Y_1, \ldots, Y_P\}$ and access to a random oracle H. The algorithm $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ programs a random oracle H' that on input X returns $\sigma_X^{-1}(\mathsf{H}(X))$, where σ_X is a permutation (chosen deterministically) with the property that the elements Y that satisfy $\mathsf{mark}(X,Y) = 1$ are mapped into the set $\{Y_1, \ldots, Y_P\}$. By assumption, $|\{Y \,|\, \mathsf{mark}(X,Y) = 1\}| \leq P$, so such a permutation always exists. Note that $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ is bounded in the number of queries it can make to H, but not bounded in time or memory. Therefore it will be able to choose such a permutation σ_X. Then, $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ invokes $\mathsf{A}_{\mathsf{MES}}$ with the programmed random oracle H'. Since H' only applies a permutation to the ouput of H, the ouputs of H' will be independent and uniformly distributed. Hence, H' is itself a perfect random oracle. Pseudocode for $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ is given below.

```
Algorithm $\mathsf{B}_{\mathsf{SM\text{-}OW}}$
define $\mathsf{H}'(X)$ as
    pick $\sigma_X$ s.t. $\sigma_X(\{Y \,|\, \mathsf{mark}(X,Y) = 1\}) \subset \{Y_1, \cdots, Y_P\}$
    return $\sigma_X^{-1} \circ \mathsf{H}(X)$
end definition
return $\mathsf{A}_{\mathsf{MES}}^{\mathsf{H}'(\cdot)}()$
```

Clearly, the number of queries that $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ makes to H is identical to the number of queries made by the simulated algorithm $\mathsf{A}_{\mathsf{MES}}$. Eventually, $\mathsf{A}_{\mathsf{MES}}$ returns a preimage X. $\mathsf{A}_{\mathsf{MES}}$ wins the MES game if $\mathsf{mark}(X, \sigma_X^{-1}(\mathsf{H}(X))) = \mathsf{True}$. By our choice of σ_X this implies that $\sigma_X(\sigma_X^{-1}(\mathsf{H}(X))) = \mathsf{H}(X) \in \{Y_1, \cdots, Y_P\}$, which shows that $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ wins his SM-OW game in this case. So $\mathsf{InSec}^{\mathsf{MES}}_{\mathsf{H},\mathsf{mark}}(Q) \leq \mathsf{InSec}^{\mathsf{SM\text{-}OW}}_{\mathsf{H},P}(Q)$. □

We are now in a position to state and prove our security claim.

Theorem 1. *Let* NEW *be the signature scheme derived from applying the transformation to a constrained linear scheme* OLD*. The maximum winning probability across all time-t adversaries in the EUF-CMA game against* NEW *that make* Q_s *signature queries and* Q_1, Q_2, Q_3, Q_4 *queries to the random oracles* $\mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3, \mathsf{H}_4$ *respectively is bounded by*

$$\begin{aligned}\mathsf{InSec}^{\mathrm{EUF\text{-}CMA}}_{\mathsf{NEW}}(Q_s, Q_1, Q_2, Q_3, Q_4; t) \leq \mathsf{InSec}^{(\sigma,\mathrm{r})\text{-}\mathrm{HSS}}_{f}(Q_s, Q_1; O(t)) + \mathsf{InSec}^{\mathrm{SM\text{-}SPR}}_{\mathsf{H}_4, 2\tau-1}(Q_4) \\ + \mathsf{InSec}^{\mathrm{SM-OW}}_{\mathsf{H}_3, L^{\vartheta}}(Q_3) + \mathsf{InSec}^{\mathrm{SM-OW}}_{\mathsf{H}_2, q^{\alpha \times (k-r+1)}}(Q_2) \ . \end{aligned} \tag{8}$$

Proof. We show through a sequence of four games how an adversary for the EUF-CMA game against NEW can be transformed into an adversary for the (σ, r)-HSS property of the underlying constrained linear trapdoor function f that wins with the same probability conditional on each of the transitions being successful. By bounding the failure probability of each transition and summing the terms we obtain a bound on the winning probability of the adversary against NEW. The sequence of games is as follows:

- The first game G_1 is the EUF-CMA game against NEW.
- The second game G_2 drops the Merkle tree. Instead, the public key consists of all the τ columns of E, and the verifier checks directly if the columns that are included in the signature are correct.
- The game G_3 drops the codeword identity testing. Instead, the public key is now the original public key ($i.e.$, M), and the verifier tests directly if the matrix T, which is included in the signature is equal to RM.
- The last game G_4 drops the random linear combinations for signature validity testing, instead G_4 is won if the errors $f(\mathbf{s_i}) - \mathsf{H}_1(m||i)$ are contained in a subspace of dimension r. G_4 is thus the (σ, r)-HSS game for the constrained linear trapdoor function f.

In games G_2, G_3 and G_4, the adversary B simulates the previous game's adversary A in order to win his own game. In particular, this means that B must answer the signing queries that A makes. This is not a problem, because in all cases B can just forward the queries that A makes to its own signing oracle, remove some information that is not required for the game that A is playing from the signature and pass the response back to A. In each case, we define the transition's failure probability as the probability that A wins but B does not. In all cases the adversary A has unbridled access (perhaps even quantum access) to the hash functions H_1, H_2, H_3 and H_4.

The event that A wins G_1 but B does not win G_2 occurs only if the signature outputted by A passes the Merkle root test, but the columns included in this signature do not agree with the columns in $E = \mathsf{Enc}(M)$. This event requires finding a second preimage for one of the $2\tau - 1$ nodes of the Merkle tree, so the failure probability is bounded by

$$\mathsf{InSec}^{\text{SM-SPR}}_{\mathsf{H}_4, 2\tau-1}(Q_4) \quad .$$

Likewise, the event that A wins the G_2 game, but B does not win the G_3 game occurs only if the columns $e_{b_1}, \cdots, e_{b_\vartheta}$ of E in the signature outputted by A are correct, but still T is not equal to RM. This implies that $\mathsf{Enc}(T)$ differs from RE in at least $\tau - L$ columns (since the rows are codewords from a code with minimal distance $\tau - L$), but that none of these columns were not chosen by the random oracle H_3. Finding $m||\mathbf{s}_1|| \cdots ||\mathbf{s}_\sigma||T$, such that this happens is a marked element search with marking function

$$\mathsf{mark}_1(m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma||T, b_1||\cdots||b_\vartheta) = \begin{cases} \mathsf{False} & \text{if } T = RM \\ \mathsf{False} & Re_{b_i} \neq \mathsf{Enc}(T)_{\star, b_i} \text{ for some } i \\ \mathsf{True} & \text{otherwise} \end{cases} \quad .$$

Since there are at most L indices for which the columns of $\mathsf{Enc}(T)$ and $R\mathsf{Enc}(E)$ are identical, there are at most $\binom{L}{\vartheta} \leq L^{\vartheta}$ marked elements for a given input. The failure probability is therefore bounded by

$$\mathsf{InSec}^{\mathrm{MES}}_{\mathsf{H}_3,\mathsf{mark}_1}(Q_3) \leq \mathsf{InSec}^{\mathrm{SM-OW}}_{\mathsf{H}_3,L^{\vartheta}}(Q_3)\,.$$

Finally, the event that A wins game G_3 but that B does not win G_4 happens when the errors span a vector space of dimension strictly larger than r (B does not win), but that all these error lie in the kernel of $R = \mathsf{H}_2(m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma)$ (otherwise A does not win). Finding $m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma$ such that this happens is a marked element search for the marking function

$$\mathsf{mark}_2(m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma, R) = \begin{cases} \mathsf{False} & \text{if } R(f(\mathsf{s}_i) - \mathsf{H}_1(m||i)) \neq 0 \text{ for some } i \\ \mathsf{False} & \text{if } \dim(\langle f(\mathsf{s}_i) - \mathsf{H}_1(m||i)\rangle_{i=0,\cdots,\sigma}) > r \\ \mathsf{True} & \text{otherwise} \end{cases} .$$

For a choice of $m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma$ there are only good matrices R if the space spanned by the errors $f(\mathsf{s}_i) - \mathsf{H}_1(m||i)$ has dimension at least $r+1$. If this is the case then the good matrices R are precisely the α-by-k matrices whose kernel contains the error space. Therefore there are at most $q^{\alpha(k-r+1)}$ good matrices for each choice of $m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma$. Therefore the failure probability of the last step is bounded by

$$\mathsf{InSec}^{\mathrm{MES}}_{\mathsf{H}_2,\mathsf{mark}_2}(Q_2) \leq \mathsf{InSec}^{\mathrm{SM-OW}}_{\mathsf{H}_2,q^{\alpha\times(k-r+1)}}(Q_2) \quad . \qquad \square$$

Joining Theorem 1 with Eqns. (5) and (6) gives the following corollaries.

Corollary 1. *In the classical random oracle model,*

$$\begin{aligned}\mathsf{InSec}^{\text{EUF-CMA}}_{\mathsf{NEW}}(Q_s, Q_1, Q_2, Q_3, Q_4; t) \leq{}& \mathsf{InSec}^{(\sigma,\,r)\text{-HSS}}_{f}(Q_s, Q_1; t) + (Q_2+1)q^{-\alpha(r+1)} \\ &+ (Q_3+1)(\ell/\tau)^{\vartheta} + (Q_4+1)(2\tau-1)/2^{\kappa} \ .\end{aligned}$$

Corollary 2. *In the quantum random oracle model,*

$$\begin{aligned}\mathsf{InSec}^{\text{EUF-CMA}}_{\mathsf{NEW}}(Q_s, \hat{Q_1}, \hat{Q_2}, \hat{Q_3}, \hat{Q_4}; t) \leq{}& \mathsf{InSec}^{(\sigma,\,r)\text{-HSS}}_{f}(Q_s, \hat{Q_1}; t) + \Theta\left((\hat{Q_2}+1)^2 q^{-\alpha(r+1)}\right) \\ &+ \Theta\left((\hat{Q_3}+1)^2(\ell/\tau)^{\vartheta}\right) + \Theta\left((\hat{Q_4}+1)^2(2\tau-1)/2^{\kappa}\right) \ .\end{aligned}$$

There are two ways to use the transformation. One can choose $\sigma = 1$ and α large enough such that $q^{\alpha/2}$ reaches the required post-quantum security level, *i.e.*, $q^{\alpha/2} > 2^{\kappa}$. Corollary 2 with $r = 0$ then guarantees that the resulting signature scheme is EUF-CMA secure, provided that the constrained linear trapdoor function f that we started from is $(1, 0)$-HSS. This assumption is equivalent to the EUF-CMA security of the original signature scheme OLD. We also note that in this case the security proof is tight, meaning that no security is lost (in the QROM) by applying the transformation in this way.

One can also use the transformation with $\sigma > r$, and a lower value of α such that $q^{\alpha\cdot(r+1)/2}$ reaches the required security level. This reduces the size of the public keys even further, but this comes at the cost of a stronger security assumption on the constrained linear trapdoor function f. In this case Corollary 2 says that the resulting signature scheme is EUF-CMA secure, if the underlying constrained linear trapdoor function is (σ, r)-HSS.

4.3 Applying the transformation

Table 1 presents a comparison of the transformation applied to the three constrained linear trapdoor signature schemes treated in Sect. 3. For the Rainbow and Micciancio-Peikert schemes part of the public key can be generated with a PRNG to reduce the size of the public key. This trick is compatible with our construction, so we have taken this into account. In all cases, 128 bits of security against quantum computers was targeted for an apples-to-apples comparison.

Table 1: Comparison of constrained linear signature schemes before and after public key compression. Legend: NC = no compression; PS = our provably secure technique based on the assumption that the original hash-and-sign signature scheme is secure; SA = the approach relying on stronger assumptions.

scheme	q	other parameters	α	σ	ϑ	τ	e	$\lvert pk\rvert$	$\lvert sig\rvert$
Rainbow NC	256	$v = 68, o_1 = 36,$ $o_2 = 36$	-	-	-	-	-	0.35 MB	0.14 kB
Rainbow PS			32	1	25	2^{20}	3	64 bytes	0.18 MB
Rainbow SA			2	16	25	2^{20}	3	64 bytes	35.51 kB
CFS NC	2	$m = 26, t = 15$	-	-	-	-	-	3.05 GB	59 bytes
CFS PS			256	1	71	2^{25}	25	32 bytes	2.00 GB
CFS SA			1	256	71	2^{25}	25	32 bytes	8.15 MB
Micciancio-Peikert NC	$2^{26} - 5$	$n = 321, m = 16692,$ $\beta = 112296$	-	-	-	-	-	8.30 MB	34.64 kB
Micciancio-Peikert PS			10	1	37	2^{20}	1	64 bytes	0.35 MB
Micciancio-Peikert SA			5	2	37	2^{20}	1	64 bytes	0.26 MB

The shrinkage is the most striking when $k \gg \alpha \cdot \sigma$, because this is when the largest part of the matrix M is omitted. The mediocre shrinkage of $|pk| + |sig|$ for the provably secure case ($\sigma = 1$) suggests that for the trapdoors considered, k is already quite close to the lower bound $k \geq \kappa/\log_2 q$ needed for κ bits of security. The greater compression factor attained when $\sigma > 1$ is due mostly to the representation of the old signatures in far less than $\ell \cdot \log_2 q$ bits.

5 Conclusion

This paper generalizes the construction of Szepieniec *et al.* [39] to a wide class of signature schemes called constrained linear signature schemes. This construction transforms a constrained linear signature scheme into a new signature scheme with tiny public keys, at the cost of larger signatures and while reducing their combined size. We prove the EUF-CMA security of the resulting signature scheme in the quantum random oracle model, and for a more aggressive parametrization we identify the (σ, r)-hash-and-sign security notion as a sufficient property for security. This improves the understanding of the security of instantiations of this construction, which includes the DualModeMS submission to the NIST PQC standardization project [29,12]. Finally, to showcase the generality and facilitate comparison, the construction is applied to an $\mathcal{MQ}$-based, a code-based and a lattice-based signature scheme, all targeting the same security level. In some cases the combined size of a signature and a public key can be reduced by more than a factor 300.

We close with some notes on the practicality of the transformation. From Table 1 we see that our transformation improves the practicality of state of the art multivariate and code-based signature schemes for applications such as public key infrastructure (PKI), where the metric $|\mathsf{sig}|+|\mathsf{pk}|$ is important and the performance of signing a message is less critical (most signatures in a PKI chain are long-lived and need not be created often). Code-based signature schemes remain not very practical, despite the improvements our construction makes. For example, applying the construction to the CFS scheme results in signatures of 8.15 MB. Still, if better code based signature schemes are developed, the construction will likely to be able to improve the quantity $|\mathsf{sig}|+|\mathsf{pk}|$. For example, even though the pqsigRM [22] proposal to the NIST PQC project does not have a completely unstructured matrix as public key, our construction can still reduce $|\mathsf{sig}| + |\mathsf{pk}|$ by a factor 6 from 329 kB to 60 kB in this case (with $\alpha = 4, \sigma = 64$). Unfortunately, comments on the NIST forum indicate that the pqsigRM proposal might not be secure [2].

State of the art hash-and-sign lattice-based signature schemes are built on structured lattices to achieve smaller public keys (e.g. Falcon relies on NTRU lattices [14]). Therefore, our construction does not improve on state of the art lattice-based schemes. Rather, our construction can be seen as an alternative to using structured lattices that provably does not deteriorate the security of the original schemes. In contrast, it is possible that switching to structured lattices has a negative impact on security.

Acknowledgements. This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work was supported by the European Commission through the EC H2020 FENTEC under grant agreement No 780108. In addition, this work was supported by imec through ICON Diskman and by FWO through SBO SPITE S002417N. Ward Beullens is funded by an FWO SB fellowship. Alan Szepieniec is being supported by a doctoral grant from the Flemish Agency for Innovation and Entrepreneurship (VLAIO, formerly IWT).

References

1. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology 9(3), 169–203 (2015)
2. Alperin-Sheriff, J., Lee, Y., Perlner, R., Lee, W., Moody, D.: Official comments on pqsigRM. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/pqsigRM-official-comment.pdf (2018)
3. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: Albers, S., Marion, J. (eds.) 26th International Symposium on Theoretical Aspects

of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings. LIPIcs, vol. 3, pp. 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2009), https://doi.org/10.4230/LIPIcs.STACS.2009.1832

4. Aumasson, J.P., Endignoux, G.: Improving stateless hash-based signatures. Cryptology ePrint Archive, Report 2017/933 (2017), http://eprint.iacr.org/2017/933
5. Bernstein, D.J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z.: SPHINCS: practical stateless hash-based signatures. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 368–397. Springer (2015), https://doi.org/10.1007/978-3-662-46800-5_15
6. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer (2011), https://doi.org/10.1007/978-3-642-25385-0_3
7. Chen, M., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass $\mathcal{MQ}$-based identification to $\mathcal{MQ}$-based signatures. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10032, pp. 135–165 (2016), https://doi.org/10.1007/978-3-662-53890-6_5
8. Courtois, N., Finiasz, M., Sendrier, N.: How to achieve a McEliece-based digital signature scheme. In: Boyd, C. (ed.) Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2248, pp. 157–174. Springer (2001), https://doi.org/10.1007/3-540-45682-1_10
9. Debris-Alazard, T., Sendrier, N., Tillich, J.: A new signature scheme based on (U|U+V) codes. IACR Cryptology ePrint Archive 2017, 662 (2017), http://eprint.iacr.org/2017/662
10. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Information Theory 22(6), 644–654 (1976), https://doi.org/10.1109/TIT.1976.1055638
11. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3531, pp. 164–175 (2005), https://doi.org/10.1007/11496137_12
12. Faugère, J.C., Perret, L., Ryckeghem, J.: DualModeMS: A dual mode for Multivariate-based signature 20170918 draft. UPMC-Paris 6 Sorbonne Universités; INRIA Paris; CNRS (2017)
13. Finiasz, M., Sendrier, N.: Security bounds for the design of code-based cryptosystems. In: Matsui [24], pp. 88–105, https://doi.org/10.1007/978-3-642-10366-7_6

14. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon (2017), submission to the NIST PQC project.
15. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008. pp. 197–206. ACM (2008), http://doi.acm.org/10.1145/1374376.1374407
16. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Annual International Cryptology Conference. pp. 112–131. Springer (1997)
17. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988), https://doi.org/10.1137/0217017
18. Güneysu, T., Lyubashevsky, V., Pöppelmann, T.: Practical lattice-based cryptography: A signature scheme for embedded systems. In: Prouff, E., Schaumont, P. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7428, pp. 530–547. Springer (2012), https://doi.org/10.1007/978-3-642-33027-8_31
19. Høyer, P., Neerbek, J., Shi, Y.: Quantum complexities of ordered searching, sorting, and element distinctness. In: Orejas, F., Spirakis, P.G., van Leeuwen, J. (eds.) Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2076, pp. 346–357. Springer (2001), https://doi.org/10.1007/3-540-48224-5_29
20. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Cheng, C., Chung, K., Persiano, G., Yang, B. (eds.) Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9614, pp. 387–416. Springer (2016), https://doi.org/10.1007/978-3-662-49384-7_15
21. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. Lecture Notes in Computer Science, vol. 1592, pp. 206–222. Springer (1999), https://doi.org/10.1007/3-540-48910-X_15
22. Lee, W., Kim, Y.S., Lee, Y.W., Jong-Seon: pqsigRM (2017), submission to the NIST PQC project.
23. Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: Matsui [24], pp. 598–616, https://doi.org/10.1007/978-3-642-10366-7_35
24. Matsui, M. (ed.): Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings, Lecture Notes in Computer Science, vol. 5912. Springer (2009), https://doi.org/10.1007/978-3-642-10366-7
25. Matsumoto, T., Imai, H.: Public quadratic polynominal-tuples for efficient signature-verification and message-encryption. In: Günther, C.G. (ed.) Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application

of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings. Lecture Notes in Computer Science, vol. 330, pp. 419–453. Springer (1988), https://doi.org/10.1007/3-540-45961-8_39

26. Merkle, R.C., Charles, R., et al.: Secrecy, authentication, and public key systems (1979)
27. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. IACR Cryptology ePrint Archive 2011, 501 (2011), http://eprint.iacr.org/2011/501
28. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Post-quantum cryptography, pp. 147–191. Springer Berlin Heidelberg (2009)
29. National Institute for Standards and Technology (NIST): Post-quantum crypto standardization (2018), http://csrc.nist.gov/groups/ST/post-quantum-crypto/
30. National Institute of Standards and Technology: FIPS PUB 186-4: Digital Signature Standard (DSS) (2013), http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
31. Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 271–288. Springer (2006)
32. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In: Maurer, U.M. (ed.) Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding. Lecture Notes in Computer Science, vol. 1070, pp. 33–48. Springer (1996), https://doi.org/10.1007/3-540-68339-9_4
33. Petzoldt, A., Bulygin, S., Buchmann, J.A.: CyclicRainbow - A multivariate signature scheme with a partially cyclic public key. In: Gong, G., Gupta, K.C. (eds.) Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6498, pp. 33–48. Springer (2010), https://doi.org/10.1007/978-3-642-17401-8_4
34. Petzoldt, A., Chen, M., Yang, B., Tao, C., Ding, J.: Design principles for HFEv-based multivariate signature schemes. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9452, pp. 311–334. Springer (2015), https://doi.org/10.1007/978-3-662-48797-6_14
35. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (1978), http://doi.acm.org/10.1145/359340.359342
36. Schnorr, C.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings. Lecture Notes in Computer Science, vol. 435, pp. 239–252. Springer (1989), https://doi.org/10.1007/0-387-34805-0_22
37. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994. pp. 124–134. IEEE Computer Society (1994), https://doi.org/10.1109/SFCS.1994.365700

38. Stern, J.: A new paradigm for public key identification. IEEE Trans. Information Theory 42(6), 1757–1768 (1996), https://doi.org/10.1109/18.556672

39. Szepieniec, A., Beullens, W., Preneel, B.: MQ signatures for PKI. In: Lange, T., Takagi, T. (eds.) Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10346, pp. 224–240. Springer (2017), https://doi.org/10.1007/978-3-319-59879-6_13

40. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9057, pp. 755–784. Springer (2015), https://doi.org/10.1007/978-3-662-46803-6_25

41. Wendl, M.C.: Collision probability between sets of random variables. *Statistics & Probability letters 64(3), 249–254 (2003)*

A CFS Parameters

Perhaps surprisingly, the most efficient attack on the CFS cryptosystem is not information set decoding (as is the case for the closely related Niederreiter cryptosystem) but a generalized birthday algorithm credited to Bleichenbacher by Finiasz and Sendrier [13]. The offline phase of this attack consists of building three lists L_0, L_1, L_2 containing sums of respectively w_0, w_1, w_2 columns from H, where $t = w_0 + w_1 + w_2$. Next, L_0 and L_1 are merged and pruned by taking the sum of each pair and keeping it only if it starts with λ zeros; the result of this operation is stored in L_0'. In the online phase a random counter i is appended to the document and the sum of $\mathsf{H}(d\|i)$ with every element of L_2 that agrees on the first λ positions is looked up in L_0' — because if this sum is present then that means that $\mathsf{H}(d\|i)$ equals the sum of $w_1 + w_2 + w_3 = t$ columns of H which can be identified by tracing the origins of the elements from L_0', L_2, L_0, L_1 that were used. Let L_1' denote the list obtained from pruning the sums of elements of L_2 and $\mathsf{H}(d\|i)$.

A single trial is successful if there is a collision between L_0' and L_1'. This is essentially a generalized birthday problem as described by Wendl [41], and the same result shows that the much more easily computed binomial distribution approximates the probability of zero collisions very well when this quantity is overwhelming. The number of pairs to consider is $\#L_0' \times \#L_1'$ and the proportion of pairs representing a collision is $1/2^{k-\lambda}$. All considered pairs fail to collide with probability $(1-2^{\lambda-k})^{\#L_0 \times \#L_1}$. By approximating $\#L_0' \approx \mathrm{E}[\#L_0'] = 2^{-\lambda}\binom{n}{w_0+w_1}$ and $\#L_1' \approx \mathrm{E}[\#L_1'] = 2^{-\lambda}\binom{n}{w_2}$ we have a probability of success of

$$\begin{aligned} \mathrm{P}_s &= 1 - \left(1 - 2^{\lambda-k}\right)^{2^{-2\lambda}\binom{n}{w_0+w_1}\binom{n}{w_2}} & (9) \\ &\approx 2^{-\lambda-k}\binom{n}{w_0+w_1}\binom{n}{w_2} + O(2^{2(\lambda-k)}) \ . & (10) \end{aligned}$$

The online complexity is $O(\mathrm{C} \cdot \mathrm{P}_s^{-1})$. The offline complexity is dominated by sorting the largest list of L_0, L_1 and L_2, as merging L_0 and L_1 can be done in linear time. Therefore, the offline complexity is $O\left(\binom{n}{\lceil w/3 \rceil} \mathsf{log}_2 \binom{n}{\lceil w/3 \rceil}\right)$.

Quantumly, there is no speed-up for sorting, and so the offline phase might as well remain classical. The online phase can be improved by applying Grover's algorithm to the "random" guess for the counter i. While sorted list lookup requires only $\frac{1}{\pi}(\mathsf{ln}(n) - 1)$ operations [19], this speed-up factor is hidden by the big-O. The λ that minimizes the online quantum complexity $O(\mathrm{C} \cdot \mathrm{P}_s^{-1/2})$ is small enough to make the offline complexity the algorithm's bottleneck. All complexities are larger than 2^{128} for the parameter set $m = 26, t = 15$, with $\lambda = 31$ being the smallest such value for which the offline complexity is larger than the quantum online complexity. At this point the public key is a bit matrix of $(15 \cdot 26) \times 2^{26}$ elements, or roughly 3.05 GB. In contrast, a signature represents a bitstring of length 2^{26} and of Hamming weight 15, which can be straightforwardly represented as 15 integers of 26 bits each, by 390 bits in total.

6.4 Short Solutions to Nonlinear Systems of Equations

Publication data

Alan Szepieniec and Bart Preneel, "Short Solutions to Nonlinear Systems Equations" *Number-Theoretic Methods in Cryptology - First International Conference, NuTMiC 2017, Warsaw, Poland, September 11-13, 2017, Revised Selected Papers*, pp. 71–90, 2017.

Contributions

Principal author

Short Solutions to Nonlinear Systems of Equations

Alan Szepieniec and Bart Preneel

imec-COSIC KU Leuven, Belgium
first-name.last-name@esat.kuleuven.be

Abstract. This paper presents a new hard problem for use in cryptography, called Short Solutions to Nonlinear Equations (SSNE). This problem generalizes the Multivariate Quadratic (MQ) problem by requiring the solution be short; as well as the Short Integer Solutions (SIS) problem by requiring the underlying system of equations be nonlinear. The joint requirement causes common solving strategies such as lattice reduction or Gröbner basis algorithms to fail, and as a result SSNE admits shorter representations of equally hard problems. We show that SSNE can be used as the basis for a provably secure hash function. Despite failing to find public key cryptosystems relying on SSNE, we remain hopeful about that possibility.

Keywords: signature scheme, hard problem, post-quantum, MQ, SIS, SSNE, hash function

1 Introduction

The widely deployed RSA and elliptic curve cryptosystems rely on the hardness of the integer factorization and discrete logarithm problems respectively, which are in fact easy to solve on quantum computers by means of Shor's algorithm [29]. These encryption and signature schemes will therefore become insecure once large enough quantum computers are built; and as a result we need to design, develop and deploy cryptography capable of resisting attacks by quantum computers, despite running on classical computers.

A number of hard problems have been proposed to replace integer factorization and discrete logarithms for precisely this purpose of offering *post-quantum* security. For instance, the problem of finding short vectors in high-dimensional lattices relates to normed linear algebra problems such as SIS [1] and LWE [27], which in turn generate many types of public key cryptosystems. Finding satisfying solutions to systems of multivariate quadratic (MQ) systems of equations seems to be hard even if the quadratic map embeds a secret trapdoor allowing only the secret-key holder to generate signatures [14]. Evaluating isogenies between elliptic curves is easy, but finding the isogeny from input and output images is hard; this enables a rather direct adaptation of the Diffie-Hellman key agreement protocol [20]. Even traditionally symmetric problems such as hash function inversion have been used to generate stateless digital signature schemes [5]. However, in nearly all post-quantum cryptosystems to date, either the public key or else the ciphertext or signature is huge — measurable in tens of kilobytes if not megabytes[1]. In the interest of easing the transition away from the quantum-insecure but very low-bandwidth ECDSA, designing a post-quantum signature scheme with short signatures or ciphertexts *and* short public keys is a major open problem.

In this paper, we propose a new cryptographic problem called Short Solutions to Nonlinear Equations (SSNE) and argue that it is likely hard, even for quantum computers. Informally, our new hard problem asks to find a *short* solution to a system of *non-linear* multivariate polynomial equations, and thus generalizes both the Short Integer Solution (SIS) problem where the system is linear, and the Multivariate Quadratic (MQ) problem where the solution need not be short. Adopting both requirements renders standard attack strategies inapplicable or wildly inefficient.

Nevertheless, we show in Section 4 that it is possible to attack SSNE with limited success, in a way that improves over brute force search. We take this attack and its limitations into account and delineate a niche of parameter space in which brute force is the most efficient attack strategy. As a result, SSNE offers a denser encoding of computational hardness than either SIS or MQ, and if it is possible to design public key cryptosystems that rely on this hard problem, it holds promise of generating a smaller public keys, ciphertexts and signatures than their MQ and SIS counterparts without incurring a security cost.

While designing a public key cryptosystem on top of SSNE remains an open problem, designing a hash function whose security relies on SSNE does not, as this problem is solved in Section 5. This result does not merely serve to demonstrate design of cryptographic primitives in lieu of the comparably more difficult end-goal of designing public key functionalities; it has standalone value as well. From the point of view of provable security, very few hash functions come with a security proof showing that finding a solution implies solving a hard problem that is defined independently of the hash function itself. Therefore these not-provably-secure hash functions offer less assurance of security than provably secure hash functions whose underlying hard problems are studied independently. Moreover, it is prudent to diversify the hard problems upon which cryptographic primitives rely, in order to isolate the effects of cryptanalytic breakthroughs.

2 Preliminaries

Notation. We denote by $\mathbb{F}_q$ the finite field of q elements. The integer range $\{a, a+1, \ldots, b-1, b\}$ is denoted by $[a : b]$. Vectors are denoted in boldface, *e.g.*, $\mathbf{x}$ and matrices by capital letters, *e.g.*, A, with indexation starting at zero. The

[1] The curious exception to this rule is the supersingular isogeny Diffie-Hellman key agreement scheme, but even so it does not seem possible to use this construction for small signature schemes.

slice of A consisting of rows i—j and columns k—l is denoted by $A_{[i:j,k:l]}$, and we drop the $, k : l$ when slicing from a vector instead of a matrix.

Lattices. A *lattice* of dimension n and embedding degree m is a discrete n-dimensional subspace of $\mathbb{R}^m$; without loss of generality, we consider subspaces of $\mathbb{Z}^m$. Any such lattice $\mathcal{L}$ can be described as the set of integer combinations of a set of vectors $\mathbf{b_0}, \ldots, \mathbf{b_{n-1}} \in \mathbb{Z}^m$, which is called a *basis* for the lattice and is not unique for a given lattice. A lattice $\mathcal{L}$ is *q-ary* whenever membership of a point $\mathbf{p} \in \mathbb{Z}^m$ is decided by $\mathbf{p} \bmod q$, *i.e.*, with each component reduced modulo q.

The LLL algorithm [24] takes a matrix of integers $A \in \mathbb{Z}^{h \times w}$ whose rows span a lattice, and outputs another matrix $B \in \mathbb{Z}^{h \times w}$ whose rows span the same lattice but are much shorter in length. Without loss of generality we assume the LLL algorithm also outputs a unitary matrix U such that $UA = B$. The shortest basis vector produced by LLL when applied to a lattice spanned by h vectors of w elements, is bounded in length by

$$\|\mathbf{b_0}\|_2 \leq \left(\frac{4}{4\delta - 1}\right)^{(w-1)/4} \mathsf{det}(\mathcal{L})^{1/w} \quad , \tag{1}$$

where $\frac{1}{4} < \delta \leq 1$ is the LLL parameter and where the *determinant of the lattice* is given by $\mathsf{det}(\mathcal{L}) = \mathsf{det}(AA^\mathsf{T})^{1/2} = \mathsf{det}(BB^\mathsf{T})^{1/2}$ if A and B have linearly independent rows.

In the case of q-ary matrices, a basis matrix can be obtained by adjoining the original basis matrix with qI. LLL will return a $(w + h) \times w$ matrix whose first w rows consist of all zeros. The determinant of q-ary lattices of this dimension is q^{w-h} with high probability [26], which means that the length of the shortest nonzero vector in the output of LLL is bounded by

$$\|\mathbf{b_0}\|_2 \leq \left(\frac{4}{4\delta - 1}\right)^{(w-1)/4} q^{(w-h)/w} \quad . \tag{2}$$

The ith *successive minimum* $\lambda_i(\mathcal{L})$ of a lattice $\mathcal{L}$ is the smallest $\rho \in \mathbb{R}$ such that the hypersphere with radius ρ centered at the origin contains at least i independent lattice points. According to the m-dimensional ball argument of Micciancio and Regev [26], the first successive minimum of a random q-ary lattice of dimension h and embedding dimension w can be approximated by

$$\lambda_0(\mathcal{L}) \approx \sqrt{\frac{w}{2\pi e}}\, q^{(w-h)/w} \quad . \tag{3}$$

3 Short Solutions to Nonlinear Equations

Our hard problem generalizes the Multivariate Quadratic (MQ) problem as well as the Short Integer Solution (SIS) problem. After presenting the definitions we consider some straightforward attacks. In the next section we consider a more sophisticated one.

MQ Problem. Given a quadratic map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ consisting of m polynomials in n variables of degree at most 2, find a vector $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathcal{P}(\mathbf{x}) = \mathbf{0}$.

The MQ problem is **NP**-hard in general as well as empirically hard on average whenever $m \approx n$. The best known attack is the hybrid attack [6], which consists of guessing some variables so as to overdetermine the system of equations and then solving it using a Gröbner basis type solver such as F_4 [16] or XL [13]. The reduced cost of solving the overdetermined system compensates for the increased cost of retrying a new guess whenever it leads to no solutions. The complexity of the optimal-trade-off hybrid attack approaches $2^{C_q n}$ as $n \gg q \to \infty$ with $C_q = \omega(1.38 - 0.44\,\omega \log_2 q)$ and where $\omega \geq 2$ is the exponent of matrix multiplication complexity [7]. However, when $q \gg n$, the cost of even one random guess beyond the number of variable-fixes that makes the system a determined one, dominates the attack complexity. In this case the complexity of a purely algebraic attack can be estimated using the *degree of regularity* D_{reg} of the system. For semi-regular quadratic systems [4,3] (which we assume random quadratic systems are), the degree of regularity is equal to the degree of the first term with a non-positive coefficient of the power series expansion of

$$\mathrm{HS}(z) = \frac{(1-z^2)^m}{(1-z)^n} \quad . \tag{4}$$

At this point, the Gröbner basis computation using F_4 or XL boils down to performing sparse linear algebra in the Macaulay matrix whose polynomials have degree D_{reg}. The complexity of this task is $O\left(\binom{n+D_{\mathrm{reg}}+1}{D_{\mathrm{reg}}}^2\right)$ in terms of the number of finite field operations, which in turn are polynomial in $\log q$. In summary, the complexity of solving the MQ problem is *exponential* in $n \approx m$, but barely affected by q.

SIS Problem. Given a matrix $A \in \mathbb{F}_q^{n \times m}$ with $m > n$, find a nonzero vector $\mathbf{x} \in \mathbb{Z}^m \backslash \{\mathbf{0}\}$ such that $A\mathbf{x} = \mathbf{0} \bmod q$ and $\|\mathbf{x}\|_2 \leq \beta$.

While not **NP**-hard, SIS does offer an average-case to worst-case reduction: solving random SIS instances is at least as hard as solving the lattice-based Shortest Independent Vectors Problem (SIVP) up to an approximation factor of $\tilde{O}(\beta\sqrt{n})$ in the worst case [25]. The most performant attack on SIS is indeed running a lattice-reduction algorithm such as BKZ 2.0 [8] to find short vectors in the associated lattice which is spanned by the kernel vectors of A. The complexity of this task is captured by the *root Hermite factor* $\delta > 1$, which approaches 1 for more infeasible computations. For a given δ the optimal number of columns of A to take into account (*i.e.*, by setting the coefficients of $\mathbf{x}$ associated to the other columns to zero) is given by $m = \sqrt{n \log_2 q / \log_2 \delta}$. At this point the average length of the lattice points found is $2^{2\sqrt{n \log_2 q \log_2 \delta}}$ and cryptographic security requires β to be smaller than this number. Albrecht *et al.* estimate the complexity of obtaining lattice points of this quality as $0.009/\log_2^2\delta + 4.1$ in terms of the base-2 logarithm of the number of time steps [2]. The key takeaway is that the complexity of SIS grows *exponentially* in m and n, but *polynomially* in q and β.

SSNE Problem (Short Solutions to Nonlinear Equations) Given a map $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ consisting of m polynomials in n variables over a prime field $\mathbb{F}_q$ and with $\mathsf{deg}(\mathcal{P}) \geq 2$, find a vector $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathcal{P}(\mathbf{x}) = 0 \bmod q$ and $\|\mathbf{x}\|_2 \leq \beta$.

It is clear that the attack strategies that work for MQ and SIS do not apply out of the box to the SSNE problem. The random guess of the hybrid attack on MQ might fix the first few variables to small values, but offers no guarantee that an algebraic solution to the other variables is small. Alternatively, one can drop the random guess and compute a Gröbner basis for the under-determined system. Even if the resulting Gröbner basis consists of a reasonable number of polynomials of reasonable degrees, obtaining a short vector in the variety associated with the Gröbner basis seems like a hard problem in and of itself. Alternatively, one can linearize the system by introducing a new variable for every quadratic term and treat the resulting matrix of coefficients as the matrix of a SIS instance. However, in this case it is unclear how to find the correct length bound β as it now applies to a vector of quadratic monomials. Nevertheless, we now show under which conditions or adaptations an algebraic attack and attack based on lattice reduction are possible.

3.1 Algebraic Attack

The constraint $\|\mathbf{x}\|_2 \leq \beta$ can be formulated algebraically. Assume $\beta < q/2$, and let $b = \lfloor \beta \rfloor$. Then any solution $\mathbf{x}$ to the SSNE problem must consist of coefficients in $[-b : b]$. For any such coefficient x_i, the polynomial $\prod_{j=-b}^{b}(x_i - j)$ must evaluate to zero. Therefore, by appending these polynomials to $\mathcal{P}$, one obtains a less under-determined system and perhaps even a determined one. If that is the case, XL and F_4 will find a short solution; however, the Gröbner basis computation must reach degree $2b$ for the added polynomials to make a difference, and for sufficiently large β even this task is infeasible. It is possible to generalize this strategy so as to require that the sums-of-squares of all subsets of the coefficients of $\mathbf{x}$ are smaller than β. This method cannot work when $\beta > q$, but can be effective when β is small — say, a handful of bits.

Alternatively, it is possible to run down the unsigned bit expansion of every component of $\mathbf{x}$ and introduce a new variable $x_{i,j}$ for each bit and one for each component's sign s_i. This transformation adds n equations of the form $x_i = s_i \sum_{j=0}^{\lceil \log_2 q \rceil} 2^j x_{i,j}$, $n\lceil \log_2 q \rceil$ equations of the form $x_{i,j}(1 - x_{i,j}) = 0$, and n equations of the form $(s_i - 1)(s_i + 1) = 0$. The advantage of having access to this bit expansion is that the constraint $\|x\|_2 \leq \beta$ can now be expressed as $\lceil \log_2 q \rceil$ equations modulo q, *even when* $\beta > q$.

In both cases, the system of equations becomes infeasibly large whenever β grows, which is exactly the intended effect from a design perspective. Phrased in terms of the security parameter κ, we have

Design Principle 1: *β must be large:* $\mathsf{log}_2\beta > \kappa$.

Note that β cannot be larger than $\sqrt{n}(q-1)/2$ because in that case *any* solution vector $\mathbf{x}$ satisfies the shortness criterion, which can therefore be forgotten at no cost in favor of a very fast algebraic solution. In fact, we want a random solution to the system of equations to satisfy $\|\mathbf{x}\|_2 \leq \beta$ with at most a negligible probability. Design principle 2 requires this probability to be at most $2^{-\kappa}$, where κ is the targeted security level.

Design Principle 2: *β must not be too large:* $n\mathsf{log}_2 q \geq \kappa + n\mathsf{log}_2\beta$.

3.2 Lattice Attack

In the relatively small dimensions considered for SSNE, basic lattice reduction algorithms such as LLL [23] manage to find the shortest vector in polynomial time with all but absolute certainty. Moreover, the nonlinear system $\mathcal{P}(\mathbf{x}) = \mathbf{0}$ can always[2] be represented as a linear system $P\bar{\mathbf{x}} = \mathbf{0}$, where P is the Macaulay matrix of $\mathcal{P}$ and $\bar{\mathbf{x}}$ is the vector of all monomials in $\mathbf{x}$ that appear in $\mathcal{P}$. If the solution $\mathbf{x}$ to $\mathcal{P}(\mathbf{x}) = \mathbf{0}$ is short enough, then its expansion into $\bar{\mathbf{x}}$ will also be a solution to $P\bar{\mathbf{x}} = \mathbf{0}$ — and might be found quickly by lattice-reducing any basis for the kernel of P and weighting the columns as necessary.

In fact, the vector $\bar{\mathbf{x}}$ associated with a solution $\mathbf{x}$ to $\mathcal{P}(\mathbf{x}) = \mathbf{0}$ will *always* lie in the kernel of P, although not every kernel vector corresponds to a solution. Since $\bar{\mathbf{x}}$ is necessarily in the lattice spanned by the kernel vectors of P, the only way to hide it from lattice-reduction is to make it long — as long as random lattice vectors taken modulo q. The rationale behind the next design principle is to require that some of the quadratic monomials $\bar{\mathbf{x}}$ are of the order of magnitude of q (possibly after modular reduction).

Design Principle 3: $\mathbf{x}$ *must not be too small:* $\mathsf{log}_2\|\mathbf{x}\|_2^2 \geq \mathsf{log}_2 q$.

A straightforward attack strategy to cope with this design principle is to focus only on those columns of P that correspond to the monomials of degree 1 in $\bar{\mathbf{x}}$. Lattice reduction will then find short kernel vectors for this reduced matrix $\tilde{P}$. The attack runs through linear combinations of these small reduced kernel vectors until it finds a small linear combination $\mathbf{c}$ such that $\mathcal{P}(\mathbf{c}) = \mathbf{0}$. A rigorous argument counts the number of points in this lattice that have the correct length and then computes the proportion of them that solve $\mathcal{P}(\mathbf{x}) = \mathbf{0}$, and infers from this a success probability and hence a running time for the attack. A far simpler but heuristic argument pretends that the nonlinear monomials of $\bar{\mathbf{x}}$ multiply with their matching columns from P and thus generate a *uniformly random*

[2] This assumes that $\mathcal{P}$ has no constant terms, but the same arguments apply with minor modifications even if it does.

offset vector $\mathbf{p}$. The attacker succeeds only when $\mathbf{p} + \tilde{P}\mathbf{x} = \mathbf{0}$, which can be engineered to occur with at most a negligible probability.

Design Principle 4: *The output space must be large enough:* $m\mathsf{log}_2\, q \geq \kappa$.

Lattice-reduction has been used in the past to find small solutions to univariate and multivariate polynomial equations, for instance in the context of factoring RSA moduli $n = pq$ where some of the bits of p or q are known. These applications of LLL were first discovered by Coppersmith [10,9], and were then expanded on by Howgrave-Graham [19], Jutla [21], Coron [11,12], and most recently by Ritzenhofen [28]. The common strategy behind all these attacks is to generate clever algebraic combinations of the polynomials but which must be linearly independent. LLL is run either on the resulting system's Macaulay matrix or on its kernel matrix to find either polynomial factors with small coefficients or else short roots. However, this family of methods is only effective when the targeted solution is short enough. In particular, if $X_i \in \mathbb{Z}$ is a bound on x_i, *i.e.*, $|x_i| \leq X_i$, then success is only guaranteed whenever for every term $t \in \mathbb{F}_q[\mathbf{x}]$ of every polynomial of $\mathcal{P}$ (interpreted as $t \in \mathbb{Z}[\mathbf{x}]$)

$$|t(X_1, \ldots, X_n)| < q \quad . \tag{5}$$

This success criterion is inconsistent with design principle 3.

3.3 Additional Considerations

Note that the shortness constraint $\|\mathbf{x}\|_2 \leq \beta$ does not have to apply to all variables. Even requiring only $\sqrt{\sum_{i \in S} x_i^2} \leq \beta$ where the sum is taken only over a non-empty subset S of the variables suffices to capture the hardness of the problem. More generally, the problem can be defined with respect to any weight matrix $W \in \mathbb{Z}^{n \times n}$, namely by requiring that $\mathbf{x}^\mathsf{T} W \mathbf{x} \leq \beta^2$. Diagonalization of W leads to a partitioning of the variables into one set which should be short and one set whose length does not matter. Nevertheless, one should be careful to ensure that the number of short variables must be larger than the dimension of the variety. Otherwise the shortness constraint is no constraint at all because it is possible to guess the short variables and subsequently solve for the remaining variables using a Gröbner basis algorithm.

Design Principle 5. *There should be more small variables than the dimension of the variety:* $\mathsf{rank}(W + W^\mathsf{T}) > \mathsf{dim}\, V(\mathcal{P}) = n - m$.

Remark. The concise way to capture "the number of variables that must be small after optimal basis change" is indeed $\mathsf{rank}(W + W^\mathsf{T})$. To see this, observe that $\mathbf{x}^\mathsf{T} W \mathbf{x}$ is a quadratic form and therefore equal to $\mathbf{x}^\mathsf{T}(W + A)\mathbf{x}$ for any skew-symmetric matrix A (*i.e.*, square matrix such that $A^\mathsf{T} = -A$). Up to additions of skew-symmetric matrices and up to constant factors we have $W \equiv W + W^\mathsf{T}$. This latter form is preferred for diagonalization, which finds an invertible basis change S such that makes $S^\mathsf{T}(W + W^\mathsf{T})S$ diagonal. The zeros on this diagonal indicate the variables whose size is unconstrained. Moreover, the rank of $W + W^\mathsf{T}$ cannot change under left or right multiplication by invertible matrices such as S^T or S.

3.4 Estimating Hardness

The main selling point of the SSNE problem is that neither the algebraic solvers nor lattice-reduction algorithms seem to apply, and as a result of this immunity it admits a much conciser encapsulation of cryptographic hardness. In MQ problems, the hardness derives from the large number of variables and equations n and m, and is largely independent of the field size q. In SIS problems, the hardness derives mostly from the large lattice dimension n, although the field size q and length constraint β are not entirely independent. Since both Gröbner basis and lattice-reduction algorithms do not apply, the hardness of SSNE problems must be much more sensitive to the size of the search space than their MQ and SIS counterparts. In particular, this sensitivity allows designers to achieve the same best attack complexity while shrinking m and n in exchange for a larger q — a trade-off that makes perfect sense because in all cases the representation of a single problem instance is *linear* in $\mathsf{log}_2\, q$ and *polynomial* in m and n.

All five design principles, including design principle 6 which will be derived in Section 4, have a limited range of applicability. No known algorithm solves SSNE problems for which all six criteria are met, faster than the following brute force search does. In the most optimistic scenario, no such algorithm exists. We invite the academic community to find attacks on SSNE that outperform this brute force search. In Section 5 we propose a hash function whose security relies on the assumption that either such an algorithm does not exist or that if it does, it does not beat brute force by any significant margin.

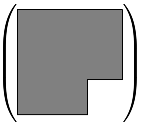
A brute force strategy must only search across $\mathbb{F}_q^{n-m}$. Each guess of the first $n - m$ variables is followed by an algebraic solution to the remaining system of m equations in m variables. If m is not too large then the task of finding this solution algebraically is rather fast, and the complexity of this joint task is dominated by $O(q^{n-m})$. In quantum complexity, Grover's algorithm [18] offers the usual square root speed-up of $O(q^{(n-m)/2})$.

4 An Algebraic-Lattice Hybrid Attack

In this section we describe an attack that applies when $m(m + 1)/2 \leq n$ and manages to produce somewhat short solutions. In a nutshell, the attack treats the polynomial system as a UOV$^-$ public key. A UOV reconciliation attack recovers the secret decomposition and at this point the attacker samples vinegar and oil variables such that the resulting "signature" is small. We consider the various steps separately. This section uses the terms "signature" and "solution" interchangeably because in the context of attacks on UOV they are identical.

4.1 UOV

Unbalanced Oil and Vinegar [22] is an MQ signature scheme with parameters $n = o + v$, $v \approx 2o$ and $m = o$. The public key is a homogeneous quadratic map $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$. The secret key is a decomposition of this public map into $\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ and $S \in \mathsf{GL}_n(\mathbb{F}_q)$ such that $\mathcal{P} = \mathcal{F} \circ S$. While S is a randomly chosen invertible matrix, $\mathcal{F}$ must have a special structure. All m components $f_i(\mathbf{x})$ partition the variables into two sets: vinegar variables $x_0, \ldots, x_{v-1}$, which are quadratically mixed with all other variables; and oil variables $x_v, \ldots, x_{n-1}$. Visually, the matrix representations of these quadratic forms have an all-zero[3] $o \times o$ block:

$$f_i(\mathbf{x}) = \mathbf{x}^\mathsf{T}$$

$$\mathbf{x} \quad . \tag{6}$$

In order to compute a signature for a document $d \in \{0,1\}^*$, the signer computes its hash $\mathbf{y} = \mathsf{H}(d)$. He then chooses a random assignment to the vinegar variables and substitutes these into the system of equations $\mathcal{P}(\mathbf{x}) = \mathbf{y}$, or more explicitly

$$\begin{cases} \qquad\qquad\qquad\qquad \vdots \\ \sum_{j=0}^{v-1} \sum_{k=0}^{j} f_{j,k}^{(i)} \underline{x_j} \underline{x_k} + \sum_{j=0}^{v-1} \sum_{k=v}^{n-1} f_{j,k}^{(i)} \underline{x_j} x_k = y_i \\ \qquad\qquad\qquad\qquad \vdots \end{cases} , \tag{7}$$

where $f_{j,k}^{(i)}$ represents the coefficient of the monomial $x_j x_k$ of the ith component of $\mathcal{F}$. The underlining indicates vinegar variables, which are substituted for their assignments. It is clear from this indication that the system of equations has become linear in the remaining oil variables, and since $m = o$, it has one easily computed solution in the generic case. The signer chooses a different assignment to the vinegar variables until there is one solution. At this point, the signature $\mathbf{s} \in \mathbb{F}_q^n$ is found by computing $\mathbf{s} = S^{-1}\mathbf{x}$. It is verified through evaluation of $\mathcal{P}$, *i.e.*, $\mathcal{P}(\mathbf{s}) \stackrel{?}{=} \mathsf{H}(d)$.

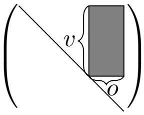
4.2 Reconciliation Attack

The reconciliation attack [15] is essentially an algebraic key recovery attack: the variables are the coefficients of S^{-1} and the equations are obtained by requiring that all the polynomials be of the same form as Eqn. 6. Naïvely, this requires solving a quadratic system of $mo(o + 1)$ equations in n^2 variables. However, the attack relies on the observation that there is almost always a viable S'^{-1}

[3] Or since it represents a quadratic form, skew-symmetric instead of all-zero.

compatible with (6) but of the form

$$S'^{-1} = \qquad . \tag{8}$$



This observation is justified by the fact that only the coefficients of S^{-1} that are located in the rightmost o columns appear as indeterminates in the coefficients that are equated to zero. Moreover, any linear recombination of these columns also maps the oil-times-oil coefficients to zero and therefore we might as well consider only the representative of this equivalence class (equivalence under linear recombination of the rightmost o columns) whose bottom right $o \times o$ block is the identity matrix.

The use of this observation reduces the number of variables to $v \times o$. Moreover, the key observation behind the reconciliation attack is that the o columns of S'^{-1} can be found iteratively, solving a new quadratic system at each step. Moreover, the authors of this attack argue that the complexity of this strategy is dominated by the first step, which requires solving only m equations in v variables [15].

These optimizations are no issue in our attack on SSNE. The parameters m and n are generally small enough to make naïvely solving a quadratic system of $mo(o+1)/2$ equations in n^2 variables feasible. However, for generic systems, whenever $mo(o+1)/2 > n^2$ there might not exist a $S^{-1} \in \mathsf{GL}_n(\mathbb{F}_q)$ that brings $\mathcal{P}$ into the form of Eqn. 6. But choosing o to be different from m might bring a suitable S^{-1} back into existence. This motivates the following definition.

Definition 1 (o-reconcilable). *A system $\mathcal{P}$ of m multivariate quadratic polynomials in n variables over $\mathbb{F}_q$ is o-*reconcilable *iff there exists an $S \in \mathsf{GL}_n(\mathbb{F}_q)$ such that $\mathcal{P} \circ S$ partitions the n variables into $v = n - o$ vinegar variables and o oil variables distinguished by $\mathcal{P} \circ S$ being linear in the oil variables.*

Remark. Clearly, constant and linear terms are linear in all variables under any change of basis. Reconcilability considers only the quadratic part of the polynomials and without loss of generality we may restrict attention to their homogeneous quadratic part.

Theorem 1 (m-reconcilability of UOV). *Let $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be the public key of a UOV cryptosystem. Then $\mathcal{P}$ is m-reconcilable.*

Proof. Trivial: follows from construction of $\mathcal{P} = \mathcal{F} \circ S$. $\mathcal{F}$ induces the required partition into oil and vinegar variables. □

Theorem 2 ($\lfloor n/2 \rfloor$-reconcilability when $m = 1$). *Assume q is odd. Let $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q$ be a single quadratic polynomial. Then $\mathcal{P}$ is $\lfloor n/2 \rfloor$-reconcilable.*

Proof. Let $Q_p \in \mathbb{F}_q^{n \times n}$ be a symmetric matrix representation of $\mathcal{P}(\mathbf{x})$ via $\mathcal{P}(\mathbf{x}) = \mathbf{x}^\mathsf{T} Q_p \mathbf{x}$. Then Q_p is diagonalizable, *i.e.*, there exists an invertible matrix $A \in \mathbb{F}_q^{n \times n}$ such that $A^\mathsf{T} Q_p A$ is nonzero only on the diagonal.

All non-zero elements on the diagonal must be one except for the last which might be the smallest quadratic non-residue in $\mathbb{F}_q$. Now choose a random symmetric matrix $Q_f \in \mathbb{F}_q^{n \times n}$ such that the lower right $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$ block consists of all zeros and such that $\mathsf{rank}(Q_f) = \mathsf{rank}(Q_p)$. It is also diagonalizable: there is an invertible matrix $B \in \mathbb{F}_q^{n \times n}$ such that $B^\mathsf{T} Q_f B$ is a diagonal matrix consisting of all ones except for the last element which might be the smallest quadratic non-residue. If $B^\mathsf{T} Q_f B = A^\mathsf{T} Q_p A$ we are done because $\mathcal{F} = \mathcal{P} \circ B^{-1} \circ A$ induces the required partition. If $B^\mathsf{T} Q_f B \neq A^\mathsf{T} Q_p A$ they must differ in the last diagonal element. So then multiply any one nonzero row of Q_f by any quadratic residue and obtain another diagonalization. Now $B^\mathsf{T} Q_f B = A^\mathsf{T} Q_p A$ must hold and we are done. □

Theorem 3. *In the generic case, a system of m quadratic polynomials in n variables over $\mathbb{F}_q$ is o-reconcilable when $m(o+1)/2 \leq n$.*

Proof. The number of coefficients of S^{-1} that are involved in the $mo(o+1)/2$ equations that set the oil-times-oil coefficients to zero is no, corresponding the rightmost $n \times o$ block of S^{-1}. The other elements of S^{-1} do not affect these coefficients. This leads to a system of $mo(o+1)/2$ quadratic equations in no variables which generically has solutions when $mo(o+1)/2 \leq no$, or equivalently when $m(o+1)/2 \leq n$. □

4.3 Generating Small Solutions

After obtaining an o-reconciliation $(\mathcal{F}, S)$, the task is to obtain a solution $\mathbf{x}$ such that $\mathcal{F}(\mathbf{x}) = \mathbf{0}$ and such that $S^{-1}\mathbf{x}$ is small. The partitioning of $\mathbf{x}$ into the vinegar variables $x_0, \ldots, x_{v-1}$ and the oil variables $x_v, \ldots, x_{n-1}$ separates the shortness objective into two parts. On the one hand, the *vinegar contribution*

$$\left(S^{-1}\right)_{[:,0:(v-1)]} \mathbf{x}_{[0:(v-1)]} \tag{9}$$

must be small; on the other hand, the *oil contribution*

$$\left(S^{-1}\right)_{[:,v:(n-1)]} \mathbf{x}_{[v:(n-1)]} \tag{10}$$

must be small as well. The reason for this separation is not just that the vinegar variables and oil variables are determined in separate steps; in fact, determining vinegar variables that lead to a small vinegar contribution is easy. The form of Eqn. 8 guarantees that small vinegar variables will map onto a small vinegar contribution. Therefore, the only requirement for selecting vinegar variables is that they be small enough, say roughly $q^{1/2}$. By contrast, the process of finding suitable oil variables is far more involved.

A quadratic map where $o > m$ can be thought of as a UOV$^-$ map, *i.e.*, a UOV map with $o-m$ dropped components. This gives the signer, or an attacker who possesses the reconciliation, $o-m$ degrees of freedom for selecting the oil variables. Coupled with the freedom afforded by the choice of vinegar variables, the signer or attacker can generate a vector $\mathbf{x}$ such that $S^{-1}\mathbf{x}$ is short.

The task is thus to find an assignment to the oil variables such that a) $\mathcal{F}(\mathbf{x}) = \mathbf{0}$ is satisfied; and b) $\left(S^{-1}\right)_{[:,v:(n-1)]} \mathbf{x}_{v:(n-1)}$ is small as well. Constraint (a) is satisfiable whenever $m \leq o$ (in the generic case, *i.e.*, assuming certain square matrices over $\mathbb{F}_q$ are invertible). Constraint (b) requires $o > m$ and the resulting vector can be made shorter with growing $o - m$.

The matrix representation of a quadratic form is equivalent under addition of skew-symmetric matrices, which in particular means that it is always possible to choose an upper-triangular representation even of UOV maps such as Eqn. 6. The ith equation of $\mathcal{F}(\mathbf{x}) = \mathbf{0}$ can therefore be described as

$$f_i(\mathbf{x}) = \mathbf{x}^\mathsf{T} \left(\begin{array}{c|c} Q_i & L_i \\ \hline & \end{array} \right) \mathbf{x} + \boldsymbol{\ell}^{(i)\mathsf{T}} \mathbf{x} + c_i = 0 \tag{11}$$

$$\left(\mathbf{x}^\mathsf{T}_{[0:(v-1)]} L_i + \boldsymbol{\ell}^{(i)\mathsf{T}}_{[v:(n-1)]} \right) \mathbf{x}_{[v:(n-1)]} = -\mathbf{x}^\mathsf{T}_{[0:(v-1)]} Q_i \mathbf{x}_{[0:(v-1)]} - \boldsymbol{\ell}^{(i)\mathsf{T}}_{[0:(v-1)]} \mathbf{x}_{[0:(v-1)]} - c_i . \tag{12}$$

All m equations can jointly be described as $A\mathbf{x}_{[v:(n-1)]} = \mathbf{b}$ for some matrix $A \in \mathbb{F}_q^{m \times o}$ and vector $\mathbf{b} \in \mathbb{F}_q^m$, because the vinegar variables $\mathbf{x}_{[0:(v-1)]}$ assume constant values. Let $\mathbf{x}^{(p)}$ be any particular solution to this linear system, and let $\mathbf{x}_0^{(k)}, \ldots, \mathbf{x}_{o-m-1}^{(k)}$ be a basis for the right kernel of A. Any weighted combination of the kernel vectors plus the particular solution, is still a solution to the linear system:

$$\forall (w_0, \ldots, w_{o-m-1}) \in \mathbb{F}_q^{o-m} \, . \, A \left(\mathbf{x}^{(p)} + \sum_{i=0}^{o-m-1} w_i \mathbf{x}_i^{(k)} \right) = \mathbf{b} \ . \tag{13}$$

This means we have $o-m$ degrees of freedom with which to satisfy constraint (b).

In fact, we can use LLL for this purpose in a manner similar to the clever selection of the vinegar variables. The only difference is that the weight associated with the vector $\mathbf{x}^{(p)}$ must remain 1 because otherwise constraint (a) is not satisfied. This leads to the following application of the embedding method.

Identify $\mathbf{x}^{(p)}$ and all $\mathbf{x}_i^{(k)}$ by their image after multiplication by $\left(S^{-1}\right)_{[:,v:(n-1)]}$, thus obtaining $\mathbf{z}^{(p)} = \left(S^{-1}\right)_{[:,v:(n-1)]} \mathbf{x}^{(p)}$ and $\mathbf{z}_i^{(k)} = \left(S^{-1}\right)_{[:,v:(n-1)]} \mathbf{x}_i^{(k)}$. Then append q^2 to $\mathbf{z}^{(p)}$ and 0 to all $\mathbf{z}_i^{(k)}$, and stack all these vectors in column form over a diagonal of q's to obtain the matrix C:

$$C = \left(\begin{array}{ccc|c} — & \mathbf{z}^{(p)\mathsf{T}} & — & q^2 \\ — & \mathbf{z}_0^{(k)\mathsf{T}} & — & 0 \\ & \vdots & & \vdots \\ — & \mathbf{z}_{o-m-1}^{(k)\mathsf{T}} & — & 0 \\ \hline q & & & \\ & \ddots & & \\ & & q & \end{array} \right) . \tag{14}$$

Run LLL on this matrix to obtain a reduced basis matrix $B \in \mathbb{Z}^{(o-m+1+n)\times(n+1)}$ of which the first n rows are zero, and a unimodular matrix U satisfying $B = UC$. The appended q^2 element guarantees that the row associated with the particular solution will never be added to another row because that would increase the size of the basis vectors. As a result, there will be one row in the matrix B that ends in q^2. Moreover, this row will be short because it was reduced by all other rows. We now proceed to derive an upper bound for the size of this vector considering only the first n elements, *i.e.*, without the q^2. Unfortunately, the best upper bound we can prove rigorously is $\lceil \frac{q}{2} \rceil \sqrt{n}$, but we can rely on the following heuristic argument for a meaningful result.

Let s be the index of this targeted row. Without row s and omitting the last column, the nonzero rows of B form an LLL-reduced basis for a q-ary lattice of dimension $o-m$ and embedding dimension n. We approximate the sizes of these vectors using $\lambda_i(\mathcal{L}) \approx \lambda_0(\mathcal{L})$. Coupled with the m-dimensional ball argument of Micciancio and Regev for estimating the first successive minimum [26], this gives

$$\|\mathbf{b}_\ell\|_2 \lesssim 2^{(o-m)/2}\sqrt{\frac{n}{2\pi e}}q^{(n-o+m)/n} \quad . \tag{15}$$

Moreover, row s (considered without the q^2) cannot be much larger than this quantity because it is LLL-reduced with respect to vectors of this size. So $\|\mathbf{b}_s\|_2 \approx \|\mathbf{b}_\ell\|_2$. Our experiments show that this heuristic bound is followed quite closely in practice for small m, n and large q.

The solution $\mathbf{s} = S^{-1}\mathbf{x}$ consists of two parts: the vinegar contribution and the oil contribution. Therefore, we can bound the size of the whole thing.

$$\begin{aligned} \|\mathbf{s}\|_2 &\leq \|S^{-1}_{[:,0:(v-1)]}\mathbf{x}_{[0:(v-1)]}\|_2 + \|S^{-1}_{[:,v:(n-1)]}\mathbf{x}_{[v:(n-1)]}\|_2 && (16) \\ &\lesssim \sqrt{n-o}\cdot q^{1/2} + 2^{(o-m)/2}\sqrt{\frac{n}{2\pi e}}q^{(n-o+m)/n} \quad . && (17) \end{aligned}$$

Or if we treat n, m, o, v as small constants,

$$\|\mathbf{s}\|_2 \in O\left(q^{(n-o+m)/n}\right) \quad . \tag{18}$$

4.4 Summary

Figure 1 shows pseudocode for the algebraic-lattice hybrid attack algorithm.

Line 1 attempts to launch a UOV reconciliation attack, but the algorithm fails when this attack is unsuccessful. In fact, the criterion for success is precisely whether the map $\mathcal{P}$ is o-reconcilable. Generically, this criterion is only satisfied for $m(o+1)/2 \leq n$, as per Theorem 3, although it is certainly possible to construct maps that are o-reconcilable for $m(o+1)/2 > n$ — indeed, standard UOV public keys match this ungeneric description. A prudent strategy for maps whose structure is unknown is to try step 1 for several values of o and to pick the decomposition of $\mathcal{P}$ where o is largest. However, in this case the length of

```
algorithm ALHA
input: $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ — a quadratic map
     : $o \in \mathbb{Z}$ — number of oil variables
output: $\mathbf{s} \in \mathbb{F}_q^n$ such that $\mathcal{P}(\mathbf{s}) = \mathbf{0}$
            and such that $\|\mathbf{s}\|_2 \in O(q^{o/n} + q^{(n-o+m)/(n+1)})$

    ▷ find decomposition $\mathcal{P} = \mathcal{F} \circ S$ where $\mathcal{F}$ is quadratic but linear in
    $x_{n-o}, \ldots, x_{n-1}$, and where $S \in \mathsf{GL}_n(\mathbb{F}_q)$
 1: try: $\mathcal{F}, S \leftarrow \mathsf{UOV\ Reconciliation\ Attack}(\mathcal{P}, o)$

    ▷ get vinegar variables $x_0, \ldots, x_{n-o-1}$
 2: $\mathbf{x}_{[0:n-o-1]} \xleftarrow{\$} [-\lfloor q^{1/2} \rfloor : \lfloor q^{1/2} \rfloor]^{n-o}$

    ▷ get oil variables $x_{n-o}, \ldots, x_{n-1}$
 3: Find $A \in \mathbb{F}_q^{m \times o}$ and $\mathbf{b} \in \mathbb{F}_q^m$ such that $A\mathbf{x}_{[(n-o):(n-1)]} = \mathbf{b} \Leftrightarrow \mathcal{F}(\mathbf{x}) = \mathbf{0}$
 4: Find particular solution $\mathbf{x}^{(p)}$ to $A\mathbf{x}_{[(n-o):(n-1)]} = \mathbf{b}$
 5: Find kernel vectors $\mathbf{x}_0^{(k)}, \ldots, \mathbf{x}_{o-m-1}^{(k)}$ of $A$
 6: $\mathbf{z}^{(p)} \leftarrow \left(S^{-1}\right)_{[:,(n-o):(n-1)]} \mathbf{x}^{(p)}$
 7: for $i \in [0 : (o-m-1)]$ do:
 8:     $\mathbf{z}_i^{(k)} \leftarrow \left(S^{-1}\right)_{[:,(n-o):(n-1)]} \mathbf{x}_i^{(k)}$
 9: end
10: Compile matrix $C$ from $\mathbf{z}^{(p)}$ and $\mathbf{z}_i^{(k)}$          ▷ according to Eqn. 14
11: $U, B \leftarrow \mathsf{LLL}(C)$
12: Find $s$ such that $B_{[s,:]}$ ends in $q^2$
13: $\mathbf{x}_{[(n-o):(n-1)]} \leftarrow \mathbf{x}^{(p)} + \sum_{i=0}^{o-m-1} U_{[s,1+i]} \mathbf{x}_i^{(k)}$

    ▷ join vinegar and oil variables, and find inverse under $S$
14: $\mathbf{s} \leftarrow S^{-1}\mathbf{x}$
15: return $\mathbf{s}$
```

Fig. 1. Algebraic-lattice hybrid attack.

the returned solution is not fixed beforehand but depends on the largest o for which step 1 succeeds.

With this algebraic-lattice hybrid attack in mind, we formulate the last design principle for SSNE instances. The rationale is that the targeted solution should be significantly smaller (*i.e.*, κ bits, spread over n variables) than what the algebraic-lattice hybrid attack can produce.

Design Principle 6: *Let o be the largest integer such that the system is o-reconcilable. If $o > m$ then guarantee that*

$$\frac{\kappa}{n} + \mathsf{log}_2 \beta \leq \frac{n-o+m}{n+1} \mathsf{log}_2 q \ . \tag{19}$$

4.5 Discussion

Equation 15 is an upper bound whereas we actually need a lower bound in order to delineate a portion of the parameter space where the attack does not apply. In practice, the short solutions found by the algebraic lattice hybrid attack are indeed shorter than the heuristic upper bound of Eqn. 17. Nevertheless, the solutions found by the attack have length very close to this bound, to the point where it is a suitable estimate. Fig. 2 plots in full blue the minimum length of solutions found by the algebraic lattice hybrid attack across one hundred trials for various modulus sizes. This graph follows the dashed green line, which represents the estimate or heuristic upper bound of Eqn. 17, quite closely. Both are far apart from the recommendation of design principle 6, which is drawn in full red. This graph represents many random SSNE instances with $m = 2$ and $n = 9$. The same behavior was observed across a wide range of parameter choices.

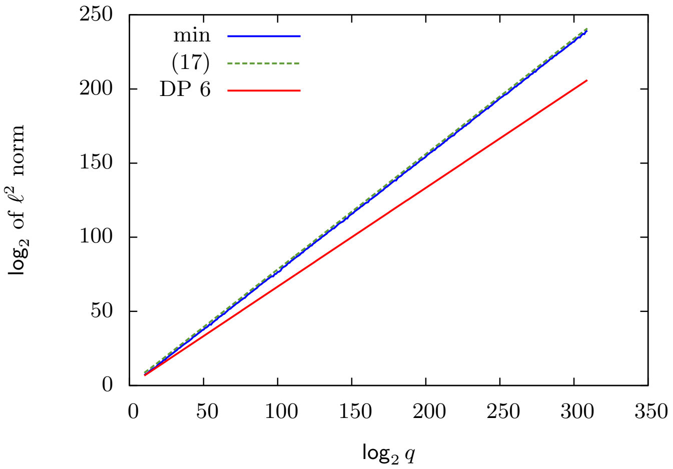


Fig. 2. Comparison of prediced length against experimental length of solutions obtained by the algebraic-lattice hybrid attack.

It is worth stressing that the algebraic-lattice hybrid attack applies only when $o > m$. When $o = m$ it does not produce solutions that are shorter than random vectors in $\mathbb{F}_q^n$, and when $o < m$ there is no guarantee it will find even one solution. Obviously, instead of requiring β to be significantly smaller than the expected length of this attack's solutions, the designer might also choose n and m so as to render the algebraic-lattice hybrid attack inapplicable.

5 Hash Function

At this time we do not know how to use SSNE to generate short-message public key functionalities. The next best option is to generate a hash function, which is what this section is about.

The resulting design does not merely exemplify using the SSNE problem constructively; it has concrete advantages over other hash functions as well. For instance, not only is the SSNE hash function provable secure (in contrast to all widely deployed hash functions), but it also relies on a *different* hard problem, which is likely to be unaffected by potential future breakthroughs in cryptanalysis of other hard problems. Also, our hash function has essentially optimal output length in terms of security: for κ bits of security against collision finders the output is 2κ bits long. This stands in contrast to many other provably secure hash functions which either have larger outputs or else require purpose-defeating post-processing functions to shrink them.

Additionally, because the hash function is built on top of SSNE instances, it requires comparably few finite field multiplications to compute. This property of having low multiplication complexity is interesting from the point of view of multiparty computation, zero-knowledge proofs, and fully homomorphic encryption, where multiplication operations are typically so expensive as to compel minimization at all costs. However, this argument ignores the cost of the bit shuffling, which are nonlinear operations over the finite field.

We note that it is possible to generate digital signature schemes from just hash functions [17,5], although the size and generation time of the signatures scales poorly. Nevertheless, anyone wanting to implement this signature scheme's key generation or signature generation procedures in a distributed manner — for instance, in order to require majority participation — must develop applied multiparty computation protocols and must consequently look to minimize multiplication complexity. Therefore, the SSNE hash function might be a good candidate for instantiating hash-based digital signature schemes with if they must enable distributed key and signature generation.

5.1 Description

We use the Merkle-Damgård construction, which requires dividing the data stream into a sequence of size b blocks. At every iteration, one data block is consumed and it is compressed with the state in order to produce a new state. The hash value is the output of the compression function after the last block has been consumed. The concept is described visually in Fig. 3.

Before applying the sequence of compression functions, the data stream $x \in \{0,1\}^*$ must first be expanded into a multiple of b bits. Let $\ell = |x|$ be the number of bits before padding, and let $\llcorner \ell \lrcorner$ be its expansion and $|\ell|$ the number of bits in this expansion. The expansion function is given by

$$\mathsf{expand} : \{0,1\}^{\ell} \rightarrow \{0,1\}^{\lceil (\ell + |\ell|)/b \rceil b} = x \mapsto x \| 0^{-\ell \bmod b} \| 0^{-|\ell| \bmod b} \| \llcorner \ell \lrcorner \ . \qquad (20)$$

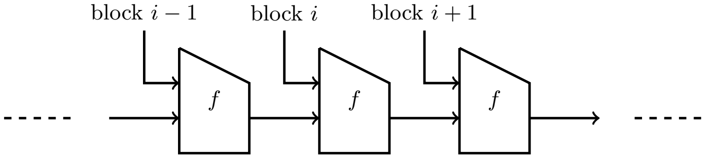


Fig. 3. Merkle-Damgård construction for hash functions.

Let q be the largest prime smaller than $2^{2\kappa}$, where κ is the targeted security level. For the purpose of defining this hash function, the elements of $\mathbb{F}_q$ are $\{0, \dots, q-1\}$. The compression function itself decomposes into $f = \mathcal{P} \circ r$. The purpose of $r : \{0,1\}^b \times \mathbb{F}_q \rightarrow \mathbb{F}_q^2$ is to permute the bits and output two integers inside $[0 : \lceil q^{3/4} \rceil - 1]$, which are then interpreted as small elements of $\mathbb{F}_q$. In particular, on input $(s, e) \in \{0,1\}^b \times \mathbb{F}_q$, this reshuffling function takes the most significant $\frac{1}{4} \lceil \mathsf{log}_2\, q \rceil$ bits of e, appends them to s, and reinterprets this bitstring as an integer. Formally, r maps

$$r : \left(s_{b-1} \| \cdots \| s_0, \sum_{i=0}^{\lceil \mathsf{log}_2 q \rceil - 1} 2^i e_i \right) \mapsto \left(\left(\sum_{i=0}^{b-1} 2^i s_i \right) + \left(\sum_{i=b}^{\lceil \frac{3}{4} \mathsf{log}_2 q \rceil - 1} 2^i e_{i+b/2} \right), \sum_{i=0}^{\lceil \frac{3}{4} \mathsf{log}_2 q \rceil - 1} 2^i e_i \right) . \tag{21}$$

In particular, this implies that $b = \frac{1}{2} \lceil \mathsf{log}_2\, q \rceil$.

The map $\mathcal{P} : \mathbb{F}_q^2 \rightarrow \mathbb{F}_q$ is a single homogeneous cubic polynomial in two variables. There are $\binom{5}{2} = 10$ coefficients which are assigned indices lexicographically from 0 to 9. Then the ith coefficient has a bit expansion equal to the first 2κ bits in the expansion of π^{i+1}, without the leading 1.

The description of the hash function is complete except for one remaining item. The initial state element, *i.e.*, the field element that is fed into the very first compression function must still be specified. For this value we choose the first 2κ bits of π^{-1}, again without the leading 1. The formal description of the algorithm is given in Fig. 4.

5.2 Security

The key property a hash function should possess is collision-resistance, which informally states that it should be difficult to find two different inputs $x, y \in \{0, 1\}$ such that $\mathsf{Hash}(x) = \mathsf{Hash}(y)$. Collision-resistance implies weaker properties such as second preimage resistance and first preimage resistance (also known as one-wayness). Therefore, it suffices to show that collisions are hard to find. We demonstrate this fact by showing that any pair of colliding values implies a collision for $\mathcal{P}$, which should be difficult to find because that task requires solving a hard SSNE instance.

First, consider that expand is injective. To see this, assume there are two different strings x and y that have the same output under expand. Then $|x| \neq |y|$ because otherwise the appended tail is the same and then the difference must

```
algorithm Hash
input: $x \in \{0,1\}^{\ell}$ — bitstring of any length
output: $h \in \{0,1\}^{2\kappa}$ — hash value

$h \leftarrow \lfloor (\pi^{-1} - \frac{1}{4}) 2^{2\kappa+2} \rfloor$
$x' \leftarrow \mathsf{expand}(x)$
for $i \in [0 : |x'|/b]$ do:
    $e_1, e_2 \leftarrow r(x'_{[ib:(ib+b-1)]}, h)$
    $h \leftarrow \mathcal{P}(e_1, e_2)$
end
return $\llcorner h \lrcorner$
```

Fig. 4. Hash function relying on SSNE.

be present in their images under expand as well. However, the last b bits of the images under expand uniquely determine the length of the original strings and this quantity must be the same, which contradicts $|x| \neq |y|$. This argument assumes the length of the inputs is less than $2^b = 2^\kappa$, which is reasonable from a practical point of view. Since expand is injective, it cannot be the source of a collision.

Next, the permutation of bits r is a bijection. It cannot be the source of a collision either.

Therefore, the only source of collisions contained in the description of the hash function is $\mathcal{P}$. Finding a collision means finding a pair of vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^2$ whose elements have at most $\frac{6}{4}\kappa$ bits, such that $\mathcal{P}(\mathbf{a}) = \mathcal{P}(\mathbf{b})$. One can re-write this equation in terms of the difference $\mathbf{d}$ from the mean $\mathbf{c} = (\mathbf{a} + \mathbf{b})/2$. The equation then becomes

$$\mathcal{P}(\mathbf{c} + \mathbf{d}) - \mathcal{P}(\mathbf{c} - \mathbf{d}) = \mathbf{0} \ . \tag{22}$$

This expression is useful because its degree in $\mathbf{c}$ is one less, *i.e.*, 2 instead of 3. Therefore, by choosing a random value for $\mathbf{d}$ the attacker finds $\mathbf{c}$ by solving a *quadratic*, instead of *cubic*, SSNE instance. (In fact, this argument was precisely the motivation for a degree-3 polynomial map $\mathcal{P}$ to begin with; to kill an attack strategy that involves only finding short solutions to *linear* equations.) The parameters of the hash function were chosen to ensure that the SSNE instance of Eqn. 22 (with randomly chosen $\mathbf{d}$) satisfies all design principles.

6 Conclusion

This paper presents a new hard problem called SSNE, which is the logical merger of the SIS and MQ problems. However, in contrast to both the SIS and MQ problems, the hardness of an SSNE instance grows linearly with the size of the modulus q. This linear scaling stands in stark contrast to the quadratic and cubic scaling of the SIS and MQ problems; and therefore, if it is possible to generate post-quantum public key cryptosystems from SSNE as it is from SIS and MQ, then these cryptosystems are very likely to require dramatically less bandwidth for having smaller public keys, ciphertexts, or signatures.

Indeed, the goal of the research that lead to the writing of this paper was to generate *public key* cryptosystems with exactly those properties. Needless to say, we have failed in that endeavor. Some of the design principles came about as a result of a process of design and attack. At least from a superficial point of view, this failure suggests that the design principles are incompatible with strategies for generating public key cryptosystems. Nevertheless, we remain hopeful about the possibility of finding strategies that are compatible with the design principles and leave their discovery as an open problem.

Acknowledgments. The authors would like to thank Fré Vercauteren and Wouter Castryck for useful discussions and references, as well as the anonymous reviewers for helpful comments. Alan Szepieniec is being supported by a Ph.D. grant from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work was supported by the European Commission through the Horizon 2020 research and innovation programme under grant agreement No H2020-ICT-2014-644371 WITDOM and H2020-ICT-2014-645622 PQCRYPTO.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Miller, G.L. (ed.) ACM STOC 1996. pp. 99–108. ACM (1996)
2. Albrecht, M.R., Cid, C., Faugère, J., Fitzpatrick, R., Perret, L.: On the complexity of the BKW algorithm on LWE. Des. Codes Cryptography 74(2), 325–354 (2015), http://dx.doi.org/10.1007/s10623-013-9864-x
3. Bardet, M.: Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie. Ph.D. thesis, Pierre and Marie Curie University, Paris, France (2004), https://tel.archives-ouvertes.fr/tel-00449609
4. Bardet, M., Faugere, J.C., Salvy, B.: On the complexity of gröbner basis computation of semi-regular overdetermined algebraic equations. In: Proceedings of the International Conference on Polynomial System Solving. pp. 71–74 (2004)
5. Bernstein, D.J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z.: SPHINCS: practical stateless hash-based signatures. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 368–397. Springer (2015)
6. Bettale, L., Faugère, J., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. J. Mathematical Cryptology 3(3), 177–197 (2009)
7. Bettale, L., Faugère, J., Perret, L.: Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In: van der Hoeven, J., van Hoeij, M. (eds.) ISSAC'12. pp. 67–74. ACM (2012)
8. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 1–20. Springer (2011)

9. Coppersmith, D.: Finding a small root of a bivariate integer equation; factoring with high bits known. In: Maurer, U.M. (ed.) EUROCRYPT '96. LNCS, vol. 1070, pp. 178–189. Springer (1996)
10. Coppersmith, D.: Finding a small root of a univariate modular equation. In: Maurer, U.M. (ed.) EUROCRYPT '96. LNCS, vol. 1070, pp. 155–165. Springer (1996)
11. Coron, J.: Finding small roots of bivariate integer polynomial equations revisited. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 492–505. Springer (2004)
12. Coron, J.: Finding small roots of bivariate integer polynomial equations: A direct approach. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 379–394. Springer (2007)
13. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer (2000)
14. Ding, J., Yang, B.Y.: Multivariate public key cryptography. In: Post-quantum cryptography, pp. 193–241. Springer (2009)
15. Ding, J., Yang, B., Chen, C.O., Chen, M., Cheng, C.: New differential-algebraic attacks and reparametrization of rainbow. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 242–257. Springer (2008)
16. Faugere, J.C.: A new efficient algorithm for computing gröbner bases (F 4). Journal of pure and applied algebra 139(1), 61–88 (1999)
17. Goldreich, O.: The Foundations of Cryptography - Volume 2, Basic Applications. Cambridge University Press (2004)
18. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: ACM STOC 1996. pp. 212–219. ACM (1996)
19. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) Cryptography and Coding, 6th IMA International Conference. LNCS, vol. 1355, pp. 131–142. Springer (1997)
20. Jao, D., Feo, L.D.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 19–34. Springer (2011)
21. Jutla, C.S.: On finding small solutions of modular multivariate polynomial equations. In: Nyberg, K. (ed.) EUROCRYPT '98. LNCS, vol. 1403, pp. 158–170. Springer (1998)
22. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) EUROCRYPT '99. LNCS, vol. 1592, pp. 206–222. Springer (1999)
23. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen 261(4), 515–534 (Dec 1982), https://doi.org/10.1007/BF01457454
24. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen 261(4), 515–534 (Dec 1982)
25. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. 37(1), 267–302 (2007)
26. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography, pp. 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg (2009), https://doi.org/10.1007/978-3-540-88702-7_5
27. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) ACM STOC 2005. pp. 84–93. ACM (2005)

28. Ritzenhofen, M.: On efficiently calculating small solutions of systems of polynomial equations: lattice-based methods and applications to cryptography. Ph.D. thesis, Ruhr University Bochum (2010), http://www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/RitzenhofenMaike/
29. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: FOCS 1994. pp. 124–134. IEEE Computer Society (1994)

Chapter 7

Unpublished Manuscripts

7.1 A Digital Signature Scheme from Short Solutions to Nonlinear Equations

Publication data

Alan Szepieniec and Aysajan Abidin and Bart Preneel, "A Digital Signature Scheme from Short Solutions to Nonlinear Equations" *This article had been submitted to CT-RSA 2019. Unfortunately, it was rejected due to an attack found by one of the reviewers.*

Contributions

Principal author

Notes (September 2018)

While the SSNE problem had already been introduced as a good candidate hard problem, this paper is the first to construct a public key cryptosystem on top of it. The main contribution of this paper is therefore the possibility result.

Nevertheless, compared to other post-quantum signature schemes it is rather slow. Furthermore, the security argument is rather weak for two reasons. First, it relies on several independent assumptions. Second, even if the assumptions and the heuristic are sound, the concrete bound obtained has a massive security degradation.

Notes (December 2018)

Unfortunately, the paper was rejected from CT-RSA. One of the reviewers found an attack which enables an attacker to forge a signature efficiently after observing only four authentic signatures. The attacker proceeds as follows.

The attacker obtains a signature $(Y, U, \mathbf{r})$ and computes the $n \times m$ matrix $V = \big((P_i + P_i^\mathsf{T})\mathbf{r}\big)_{i=0}^{m-1} \bmod q$. Then $\mathbf{x}^\mathsf{T} V = 2aX + U \bmod q$, thus providing the attacker with $m = 6$ linear equations in the $n = 19$ variables of the secret key $\mathbf{x}$. By collecting the equations from four different signatures, the attacker has a linear system of 24 equations in 19 variables. Solving this equation produces the secret key.

How was this cryptanalysis possible despite security proof? The answer has to do with how a valid proof is only as good as the assumptions that go into it. One of these premises was the *SSNE heuristic*: “upon encountering an SSNE system, and upon failing to identify any particular structure that would make its solution efficient, the system of equations may be assumed to be indistinguishable from a random one.” The security proof invokes this heuristic to argue about the indistinguishability of two SSNE systems. However, the cryptanalysis shows that these systems *do* have a particular structure that can help to find solutions.

In particular, the SSNE system in question is given by Eqn. 14 of the paper:

$$\begin{cases} a \cdot \mathbf{x} + \mathbf{y} = \mathbf{r} \\ (\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1} = X \bmod q \\ (\mathbf{y}^\mathsf{T} P_i \mathbf{y})_{i=0}^{m-1} = Y \bmod q \\ (\mathbf{x}^\mathsf{T}(P_i + P_i^\mathsf{T})\mathbf{y})_{i=0}^{m-1} = U \bmod q \end{cases} .$$

The first line can be taken modulo q as no overflow is guaranteed to occur. For authentic transcripts, there is a short solution $(\mathbf{x}, \mathbf{y})$, where “short” means $\sqrt{\mathbf{x}^\mathsf{T}\mathbf{x} + \mathbf{y}^\mathsf{T}\mathbf{y}} \leq \sqrt{2n} \cdot 2^u$ with $u = 9.60\lambda$ and $\log_2 q \approx 17\lambda$. For forged transcripts no such short solution is guaranteed to exist. Under the SSNE heuristic, determining whether the system of equations has a short solution is hard, which is exactly the same as authentic transcripts being indistinguishable from counterfeit ones.

However, in order to apply the SSNE heuristic, the SSNE system cannot contain any particular structure. The cryptanalysis shows that it does. Indeed, if a short solution exists, the lattice (up to translation) of solutions to the pair of equations $\mathbf{x}^\mathsf{T} V \mathbf{r} = 2aX + U \mod q$ and $\mathbf{y}^\mathsf{T} V \mathbf{r} = 2Y + U \mod q$ must have lattice vectors of length less than or equal to $\sqrt{2n} \cdot 2^{9.60\lambda}$. Whereas a random system of $2m = 12$ linear equations in $n = 19$ variables is only expected to have solutions of length about $q^{2m/n} \approx 2^{17 \cdot \lambda \cdot 12/19} \approx 2^{10.74\lambda} \gg 2^{9.60\lambda}$. Therefore, the question whether the given SSNE system has a short solution can be solved by running a lattice reduction algorithm and testing the length of the shortest vector against $2^{10.74\lambda}$. In other words, this SSNE system exhibits exactly the kind of structure that it is not allowed to have in order for the SSNE heuristic to apply.

This flaw makes the paper effectively unpublishable in its present form. Nevertheless, I am happy to include it as it is in this dissertation based on the merit of the following points.

- This result provides further evidence that generating an efficient signature scheme from SSNE is a challenging problem. The next would-be designer of cryptosystems based on SSNE therefore has one more reference with which to bolster that claim.
- The paper constitutes an instructive and comprehensive example of the full design path from hard problem to zero-knowledge proof to signature scheme.
- A negative result is still a result. It is conceivable that someone else, challenged to generate an efficient signature scheme from SSNE, opts for roughly the same strategy. In this case, they need not repeat the part that fails and they may even recycle the parts that succeed.
- This failure prompts the drawing of valuable lessons regarding provable security in cryptography:
  - Provable security is not a panacea: proofs can fail.
  - Nevertheless, even security proofs with shaky assumptions are useful, because they help to focus the attention of cryptanalysts. It is worth noting that the scheme was broken exactly in one of the links we identified as weak: the SSNE heuristic.
  - However, security proofs also distract from cryptanalysis: looking for a leaked linear relation of the secret key is far a more tangible

and understandable task than looking for non-random structure in a given object.

- Hard problems may be defined in the abstract and may even be hard in general. However, for cryptographic security we do not care about worst-case hardness or even average-case hardness *per se* — we care about the hardness of concrete problem instances arising from their application as a component in a larger cryptosystem.

A Digital Signature Scheme from Short Solutions to Nonlinear Equations

Alan Szepieniec and Aysajan Abidin and Bart Preneel

Dept. Electrical Engineering,
imec-COSIC, KU Leuven, Belgium
{first-name}.{last-name}@esat.kuleuven.be

Abstract. Short Solutions to Nonlinear Equations (SSNE) is a post-quantum hard problem introduced recently in the context of cryptosystem design [30]. By logically merging the SIS and MQ problems, the SSNE problem renders standard solving strategies either obsolete or wildly inefficient, and thus promises a better scaling of hardness to representation size. As a consequence of this conciser encoding, cryptosystems relying on SSNE may induce far smaller bandwidth requirements than their SIS and MQ counterparts. However, until now, no public key constructions based on SSNE have been proposed.
This paper introduces a zero-knowledge proof system for proofs of knowledge of a short solution to a quadratic system of equations. The Fiat-Shamir transform turns the zero-knowledge proof into a signature scheme with a public key and signature of little over 12 kB for the highest security level. A proof of concept implementation in Sage validates the design and indicates that all operations execute in time on the order of seconds.

Keywords: post-quantum cryptography, provable security, random oracle model, zero-knowledge proof, digital signature scheme, SSNE

1 Introduction

Post-Quantum Cryptography. A large number of widely deployed cryptosystems such as RSA [27] and ECC [21, 23] rely on the assumed intractability of number theoretic and elliptic curve problems. However, this assumption is known to be false in the context of quantum computation [29]. In response to the threat posed by future quantum computers, much research is devoted to *post-quantum cryptography* [5], the effort to design, develop and deploy cryptographic algorithms capable of resisting attacks on quantum computers despite running on today's classical hardware.

For instance, the US National Institute of Standards and Technology (NIST) has started a post-quantum standardization project with the purpose to issue a standard for three of the most basic public key functionalities: digital signature schemes, key encapsulation mechanisms, and public key encryption schemes [24]. Their call for proposals has garnered 69 submissions, relying on a variety of mathematical problems and associated computational hardness assumptions for which no efficient quantum algorithm is known.

Unfortunately, any migration towards post-quantum cryptographic standards incurs a bandwidth penalty. No post-quantum cryptosystem is capable of making both the public key and the ciphertext or signature as small as those produced by elliptic curve cryptosystems (ECC). Instead, the more balanced cryptosystems boast public keys and ciphertexts or signatures measurable in *kilobytes*, as opposed to *tens of bytes* for ECC. One of the biggest challenges in post-quantum cryptography is to push this number down further, not only to streamline the anticipated migration but also to make the cryptography accessible to resource-constrained devices.

Short Solutions to Nonlinear Equations Problem. One of the cryptographically useful hard problems that is not represented by any of the 69 NIST submissions is the Short Solutions to Nonlinear Equations (SSNE) Problem [30]. The problem had been studied under various guises in the context of cryptanalysis [9–13, 18, 19, 26]. It was presented only recently as a good candidate for generating small-bandwidth post-quantum cryptosystems, but ultimately this generation was merely conjectured to be possible as the authors could not find a way to do it [30].

Informally, the SSNE Problem asks to find a short solution to a nonlinear system of multivariate polynomial equations. It generalizes both the Short Integers Solution (SIS) Problem [1], where the system of equations is linear; as well as the Multivariate Quadratic (MQ) [14] Problem, where the solution need not be short. The double requirement renders standard attack strategies applying to SIS or MQ obsolete or wildly infeasible, and thus enables a conciser representation of an equally hard problem. In particular, the size of an SSNE instance in a straightforward representation scales *linearly* with the logarithm of the best attack complexity; in contrast to the higher degree (but still polynomial) scaling associated with equally straightforward representation of SIS and MQ instances.

Signatures from Zero-Knowledge Proofs. One common strategy for generating signature schemes is to start with a zero-knowledge proof and apply the Fiat-Shamir transform [16]. This transformation replaces interactive challenges with hash function evaluations to make protocol non-interactive but still secure in the random oracle model (ROM). By also including the document to be signed in the input to the hash function, the non-interactive proof testifies to the involvement of the secret key in its generation while linking it to the document in question. The resulting transcript therefore provides non-repudiation of origin, which is the defining property of signatures.

The key property in this context is witness-extractability, which formalizes the notion that a successful prover could have outputted the witness just as easily. It requires the existence of an extactor machine that is capable of outputting this witness whenever it has black box access to a successful prover. This extractor is traditionally constructed, both in the interactive and non-interactive case, with the Forking Lemma [25]: the extractor records the prover's state just after it made a commitment but before it receives a challenge from the verifier. Then the extractor tricks the prover into generating responses to two different challenges, but both valid with respect to the same commitment. The protocol should guarantee that the witness is efficiently computable from a small number of transcripts with the same beginning but different endings, which the extractor can obtain in this way.

In the context of provable security against quantum adversaries, the preferred notion of *knowledge* is *quantum-witness-extractability* [31], which allows the extractor to be a quantum computer if the successful prover is. In this setting, the Forking Lemma is invalid because it relies on copying information, which is impossible for generic quantum states. Nevertheless, cryptosystems following this design pattern can be classified as "post-quantum" because no quantum attack exploiting this invalidity is known. The classical security proof, though invalid quantumly, is to be interpreted as another argument for the cryptosystem's security.

Contributions. In this paper, we take another look at the SSNE problem for generating post-quantum public key cryptosystems. We answer the challenge posed in the SSNE paper [30] positively, and validate the intuition stated therein about its potential for low bandwidth schemes. In particular:

- We revisit and revise the SSNE problem. Particularly, we briefly sketch an attack that mandates an update to the design principles for guaranteeing a targeted level of security. Moreover, we cast the search and decision problems and their induced hardness assumptions into an exact formal language, thus enabling their usage in security proofs. Our analysis requires the formalization of an additional assumption, which states that finding triples of colliding inputs is hard as well.
- We propose a zero-knowledge proof system for proofs of knowledge based on the SSNE problem, which can double as an identification scheme. Our proof system resembles that of Schnorr for finite field and elliptic curve groups [28], but in contrast to Schnorr proofs, no quantum attack is known to defeat SSNE.
- A straightforward application of the Fiat-Shamir transform generates a post-quantum signature scheme from this protocol. Assuming the average case hardness of the SSNE problem, and heuristically assuming that generic SSNE systems behave as random ones do, the scheme is provably secure in the (classical) random oracle model.
- We present a proof of concept implementation in Sage to validate the design. The scheme produces relatively short public keys and signatures (3.22 kB and 12.09 kB respectively at the highest level of security), and the operational speed of this high-level implementation is on the order of seconds.

We stress that SSNE is a relatively new problem in the context of cryptosystem design, albeit much older in the context of cryptanalysis. It does not and should not at this time inspire the same confidence that other post-quantum problems do, particularly the ones that have received and withstood decades of scrutiny. Nevertheless, the shortest path towards justified confidence is to incentivize cryptanalytic attention. By presenting a signature scheme as we do with performance matching or in excess of its competition, we hope to invite this much-needed examination.

While we are unable to provide a proof of security that is valid in the quantum random oracle model (QROM) as well as in the classical random oracle model, we note that no quantum attack is known to break the security of the Fiat-Shamir transform.[1] Therefore, our signature scheme is justifiably classified as *post-quantum* precisely because no quantum attack is known. The unavailability of a security proof for Fiat-Shamir in the QROM might be merely an artifact of the stronger computational model and of the mortality of the humans writing the proofs, rather than an indication of some inherent weakness. Nevertheless, it remains an interesting open question to find a non-interactivity transform with provable security in the quantum computing model that does not incur a large speed and bandwidth overhead. Indeed, such a solution would be applicable to many other zero-knowledge based signatures schemes beyond our own.

2 Preliminaries

Negligible. A function $\epsilon : \mathbb{N} \rightarrow \mathbb{R}_{>0}$ is *negligible* if for all polynomials $p(x) \in \mathbb{R}[x]$ there is an $N \in \mathbb{N}$ such that for all $x > N$, $\epsilon(x)$ drops faster than the reciprocal of $|p(x)|$. Conversely, a function $\nu : \mathbb{N} \rightarrow \mathbb{R}_{>0}$ is *noticeable* if there exists a polynomial $p(x)$ whose reciprocal drops faster. Formally, we need only consider the dominant monomial of $p(x)$:

$$\forall c > 1 \,.\, \exists N \in \mathbb{N} \,.\, \forall \lambda > N \,.\, \epsilon(\lambda) \leq \frac{1}{\lambda^c} \;\; ;$$
$$\exists c > 1 \,.\, \exists N \in \mathbb{N} \,.\, \forall \lambda > N \,.\, \nu(\lambda) \geq \frac{1}{\lambda^c} \;\; .$$

A probability is *overwhelming* if its distance from 1 is negligible. From here on, any reference to negligible or noticeable functions drops the quantifiers from the notation. They are still implicitly present whenever asymptotic security notions, the functions ϵ and ν, or the *security parameter* λ appear.

Pseudorandom Generator. A pseudorandom generator is a deterministic algorithm that expands a short input seed into a long bitstring that is indistinguishable from uniform. Formally, a function $G : \{0,1\}^a \rightarrow \{0,1\}^b$ with $a < b$ is a peudorandom generator if for all quantum polynomial time distinguishers D the distinguishing advantage $\mathsf{Adv}_G^{\mathsf{PRG}}(\mathsf{D})$ is negligible:

$$\mathsf{Adv}_G^{\mathsf{PRG}}(\mathsf{D}) \triangleq \left| \Pr_{s \xleftarrow{\$} \{0,1\}^a}[\mathsf{D}(G(s)) \Rightarrow 1] - \Pr_{g \xleftarrow{\$} \{0,1\}^b}[\mathsf{D}(g) \Rightarrow 1] \right| \leq \epsilon(a) \;\; . \quad (1)$$

[1] Ambainis, Rosmanis, and Unruh do have a result showing that the Fiat-Shamir construction is classically-secure but quantumly-insecure *relative to an oracle*, which may or may not be realizable [2].

Among other things, pseudorandom generators are useful for *derandomization*, which is the process by which a probabilistic algorithm is made deterministic by exchanging its random coins for pseudorandom ones, and fixing or transmitting the seed somehow. We drop the argument to denote the maximum distinguishing advantage over all polynomial-time quantum adversaries:

$$\mathsf{Adv}_G^{\mathsf{PRG}} \triangleq \mathsf{max}_{\mathsf{D}}\, \mathsf{Adv}_G^{\mathsf{PRG}}(\mathsf{D}) \ . \tag{2}$$

Random Oracle Model. A random oracle is an idealization of a hash function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$ that captures the complete ignorance of an adversary about images of inputs in which the function was not evaluated [4,16]. Formally, a random oracle is a function $\mathsf{RO} : \{0,1\}^* \to \{0,1\}^\lambda$ from arbitrary length bitstrings to fixed length bitstrings drawn uniformly at random from the space of all functions of that type signature: $\mathsf{RO} \xleftarrow{\$} \{f \,|\, f : \{0,1\}^* \to \{0,1\}^\lambda\}$. A proof that holds when all hash functions are replaced by (possibly different) random oracles are said to hold in the *random oracle model.* In the context of post-quantum cryptography, the *quantum random oracle model (QROM)* is preferred [6], because this stronger model captures the realistic capability of the attacker to evaluate the hash function on a quantum superposition of values. Not all proofs that are valid in the classical random oracle model are also valid in the quantum random oracle model.

2.1 Fiat-Shamir Transform

We assume the reader is familiar with the syntax and security notions of signature schemes and zero-knowledge identification schemes. Otherwise, they are referred to Appendix A for a quick recap.

Informally, the Fiat-Shamir transform replaces the public coin challenges from the verifier with the hash of all protocol messages up until that point [16]. The result is a *non-interactive* zero-knowledge proof. In order to turn the protocol into a signature scheme, the message to be signed must be hashed as well. In some cases, the transcript leaks information about the witness and in this case the transformation should abort and try again with new randomness. This is the strategy of *Fiat-Shamir with Aborts* [22]. The parameter κ determines the number of tries before signature generation fails.

Formally, let $\varDelta = (\varDelta.\mathsf{KeyGen}, \varDelta.\mathsf{P}, \varDelta.\mathsf{V})$ be an identification scheme with $\mathsf{leaks}(com, ch, rsp)$, a Boolean function that determines whether the given transcript leaks information about the witness. Furthermore, let H be a hash function and G a pseudorandom generator. Then the *deterministic Fiat-Shamir transform with aborts and derandomization* generates a signature scheme $\varSigma = \mathsf{DFSAD}[\varDelta, \mathsf{leaks}, \kappa, \mathsf{H}, \mathsf{G}] = (\varSigma.\mathsf{KeyGen}, \varSigma.\mathsf{Sign}, \varSigma.\mathsf{Verify})$ with $\varSigma.\mathsf{KeyGen}$, $\varSigma.\mathsf{Sign}$, $\varSigma.\mathsf{Verify}$ defined as follows: $\varSigma.\mathsf{KeyGen} = \varDelta.\mathsf{KeyGen}$ and

```
define $\Sigma$.Sign$(pk, sk, d)$ as:
    $\{coins_{1,i}, coins_{2,i}\}_{i=0}^{\kappa-1} \leftarrow \mathsf{G}(sk\|d)$
    for $i \in \{0, \ldots, \kappa - 1\}$ do:
        $com, st \leftarrow \Delta.\mathsf{P}(sk; coins_{1,i})$
        $ch \leftarrow \mathsf{H}(pk\|com\|d)$
        $rsp \leftarrow \Delta.\mathsf{P}(st, ch; coins_{2,i})$
        if not $\mathsf{leaks}(com, ch, rsp)$ then:
            return $s = (com, rsp)$
return $\perp$ ,
```

```
define $\Sigma$.Verify$(pk, d, s)$ as:
    $(com, rsp) \leftarrow s$
    $ch \leftarrow \mathsf{H}(pk\|com\|d)$
    return $\Delta.\mathsf{V}(pk, com, ch, rsp)$ .
```

The scheme Σ is provably secure in the classical random oracle model [25]. The same is not known to hold in the quantum random oracle model. Unruh shows that the Fiat-Shamir transform retains *soundness* against quantum adversaries [33], but in order for the resulting signature scheme to be secure it must retain *witness-extractability* as well.

3 Zero-Knowledge Proof System based on SSNE

3.1 SSNE

The Short Solutions to Nonlinear Equations (SSNE) problem was introduced in the context of design of cryptographic primitives by Szepieniec and Preneel [30], although essentially the same problem has been known for much longer in the context of cryptanalysis [9–13,18,19,26]. The problem can be seen as the logical merger of the SIS problem, *i.e.*, finding short solutions to linear systems of equations, with the MQ problem, *i.e.*, finding *any* solution to a system of quadratic equations. A formal definition is as follows.

Definition 1 (SSNE Problem). *Given $\mathcal{P} \in (\mathbb{F}_q[\mathbf{x}])^m$, a list of m multivariate polynomials in n variables $\mathbf{x} = (x_1, \ldots, x_n)^\mathsf{T}$ over a finite field with prime order q, find a vector of n integers $\mathbf{x} \in \mathbb{Z}^n$ such that*

$$\mathcal{P}(\mathbf{x}) = 0 \bmod q \quad \text{and} \quad \|\mathbf{x}\| \leq \beta \ ,$$

for some parameter $\beta \in \mathbb{R}_{>0}$ and where $\|\cdot\|$ denotes the ℓ^2 norm.

Szepieniec and Preneel identify six design principles for choosing parameters such that the problem is hard. In particular, when all design principles are satisfied, no known algorithm, classical or quantum, is capable of producing solutions to SSNE faster than brute force. The principles for targeting λ bits of security are:

1. $\log_2 \beta \geq \lambda$
2. $n(\log_2 q - \log_2 \beta) \geq \lambda$
3. $\log_2 \|\mathbf{x}\|^2 \geq \log_2 q$, for all solutions $\mathbf{x}$
4. $m \log_2 q \geq \lambda$
5. $\mathsf{rank}(W + W^\mathsf{T}) \geq \mathsf{dim} V(\mathcal{P})$, for a generalized length criterion $\mathbf{x}^\mathsf{T} W \mathbf{x} < \beta^2$
6. $o > m \implies \frac{n-o+m}{n+1} \log_2 q \geq \lambda/n + \log_2 \beta$, where $o = \mathsf{max}_o\, o$ subject to $m(o+1)/2 \leq n$ and $o < n$.

There is an attack mandating a revision of design principle 2. It is possible to fix the first $n-m$ variables to random but small enough values and then solve the polynomial system for the remaining variables. The solution will be short enough with probability $2^{-m\delta}$, where $\delta = \mathsf{log}_2\, q - \mathsf{log}_2\, \beta$. Making this probability negligible requires setting $m\delta > \lambda$. We revise the design principle to do just that, and note that this change makes principle 4 superfluous because $m\delta > k$ would imply $m \log_2 q > k$.

Design Principle 2':

$$m(\mathsf{log}_2\, q\; - \mathsf{log}_2\, \beta) \geq \lambda\ .$$

In order to make a meaningful assumption about the average-case hardness of SSNE, one must define a probability distribution of problem instances. However, some systems of polynomial equations may be designed to contain a trapdoor allowing the secret key holder to efficiently solve the associated SSNE problem. We solve this problem by requiring that every coefficient of the system of polynomials be chosen at random.

Search and Decisional SSNE Assumptions. Let $\mathcal{P} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ be a list of m random polynomials of degree at most $deg \geq 2$ in n variables over a prime field $\mathbb{F}_q$; and let $\beta \in \mathbb{R}_{>0}$ be a target length, and λ the security parameter. If design principles 1—6 are satisfied for security level λ, then for all polynomial-time quantum algorithms D and S, the success probability in the search SSNE game (Game 1) and the advantage in the decisional SSNE game (Game 2) is negligible:

$$\mathsf{Succ}^{\mathsf{SSSNE}}(\mathsf{S}) \triangleq \Pr[\mathsf{Game}^{\mathsf{S}}_{\mathsf{SSSNE}}(m, n, deg, \beta) \Rightarrow 1] \leq 2^{-\lambda} \tag{3}$$

$$\mathsf{Adv}^{\mathsf{DSSNE}}(\mathsf{D}) \triangleq \Pr[\mathsf{Game}^{\mathsf{D}}_{\mathsf{DSSNE}}(m, n, deg, \beta) \Rightarrow 1] \leq 2^{-\lambda}\ . \tag{4}$$

Game 1: Search SSNE (SSSNE)

```
define $\mathsf{Game}^{\mathsf{S}}_{\mathsf{SSSNE}}(m, n, deg, \beta)$ as:
    $\mathcal{P} \xleftarrow{\$} \mathbb{F}_q[\mathbf{x}^{\leq deg}]$
    $\mathbf{x} \xleftarrow{\$} \{\mathbf{x} \in \mathbb{Z}_q^n \,|\mathbf{x}^\mathsf{T}\mathbf{x} \leq \beta^2\}$
    $\mathbf{z} \leftarrow \mathcal{P}(\mathbf{x})$    ▷ evaluate $\mathcal{P}$ in $\mathbf{x}$
    $\hat{\mathrm{x}} \leftarrow \mathsf{S}(\mathcal{P}, \mathbf{z})$
    return $[\![\mathcal{P}(\hat{\mathrm{x}}) = \mathbf{z}]\!]$
```

Game 2: Decisional SSNE (DSSNE)

```
define $\mathsf{Game}^{\mathsf{D}}_{\mathsf{DSSNE}}(m, n, deg, \beta)$ as:
    $\mathcal{P} \xleftarrow{\$} \mathbb{F}_q[\mathbf{x}^{\leq deg}]$
    $\mathbf{x} \xleftarrow{\$} \{\mathbf{Z} \in \mathbb{F}_q^n \,|\mathbf{x}^\mathsf{T}\mathbf{x} \leq \beta^2\}$
    $\mathbf{z}_0 \leftarrow \mathbf{0}$
    $\mathbf{z}_1 \leftarrow \mathcal{P}(\mathbf{x})$    ▷ evaluate $\mathcal{P}$ in $\mathbf{x}$
    $b \xleftarrow{\$} \{0, 1\}$
    $\hat{b} \leftarrow \mathsf{D}(\mathcal{P} - \mathbf{z}_b)$
    return $[\![b = \hat{b}]\!]$
```

We write $\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\beta}(\mathsf{S})$ and $\mathsf{Adv}^{\mathsf{DSSNE}}_{m,n,q,\beta}(\mathsf{D})$ to capture the success probability and distinguishing advantage of adversaries against generic SSNE instances with the given parameters. We drop the argument to refer to the maximum success probability and advantage across all polynomial-time quantum algorithms:

$$\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\beta} \triangleq \mathsf{max}_{\mathsf{S}}\, \mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\beta}(\mathsf{S})\ ,\ \ \mathsf{Adv}^{\mathsf{DSSNE}}_{m,n,q,\beta} \triangleq \mathsf{max}_{\mathsf{D}}\, \mathsf{Adv}^{\mathsf{DSSNE}}_{m,n,q,\beta}(\mathsf{D})\ . \tag{5}$$

Note that a solver can always be used as a sub-procedure in a distinguisher, so for every solver S there is a distinguisher D^{S} such that $\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\beta}(\mathsf{S}) \leq \mathsf{Adv}^{\mathsf{DSSNE}}_{m,n,q,\beta}(\mathsf{D}^{\mathsf{S}})$. Also, a solution computed by a solver is also a solution for the same system but a with larger length constraint. So for every $\rho > 0$ and every solver S, the success probabilities satisfy $\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\beta}(\mathsf{S}) \leq \mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\beta+\rho}(\mathsf{S})$.

SSNE Heuristic. In addition to making both assumptions, we employ the following heuristic argument: *upon encountering an SSNE system, and upon failing to identify any particular structure that would make its solution efficient, the system of equations may assumed to be indistinguishable from a random one.* While technically speaking invalid, this heuristic argument is still useful inside an otherwise valid proof because the conclusion is true with overwhelming probability if the premises are true; moreover, this heuristic argument pinpoints the locations where the proof may break. We refer to this heuristic as the *SSNE heuristic.*

It is certainly possible to avoid employing the SSNE heuristic altogether and make the proof in which it is used perfectly valid. For instance, one can identify the processes that produce the generic systems, and assume explicitly (or prove, if possible) that this process produces systems that are computationally or perfectly indistinguishable from random. However, we feel that this strategy to eliminate heuristics distracts from the intuition behind the security argument and results in a convoluted proof. We choose to err on the side of simplicity and intuition.

Triple-Collision-Resistance. We require a third hardness property of SSNE systems, specifically for the case where $deg = 2$, namely that it be hard to find $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{Z}_q^n$ such that for all $i \in \{1, 2, 3\}$, $\|\mathbf{x}_i\| \leq \beta$, and $\mathcal{P}(\mathbf{x}_1) = \mathcal{P}(\mathbf{x}_2) = \mathcal{P}(\mathbf{x}_3)$. If this is the case, we call $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ a *triple-collision.*

To see why we consider tuples of at least three items, consider the following procedure for finding colliding pairs. First, choose a small difference $\boldsymbol{\delta} \xleftarrow{\$} \mathbb{Z}_{2^u}^n$ at random. Require that $\mathbf{x}_1 \triangleq \mathbf{x} - \boldsymbol{\delta}$ and $\mathbf{x}_2 \triangleq \mathbf{x} + \boldsymbol{\delta}$ collide, *i.e.*, $\mathcal{P}(\mathbf{x} - \boldsymbol{\delta}) = \mathcal{P}(\mathbf{x} + \boldsymbol{\delta})$. Since $\mathcal{P}(\mathbf{x})$ is quadratic, it can be written $\left(\mathbf{x}^\mathsf{T} P_i \mathbf{x}\right)_{i=0}^{m-1} + L\mathbf{x} + \mathbf{c}$ for some matrices $P_0, \ldots, P_{m-1}, L$ and vector $\mathbf{c}$. Then move all the terms of the collision equation to the right hand side and observe that the quadratic terms cancel:

$$\left((\mathbf{x} - \boldsymbol{\delta})^\mathsf{T} P_i (\mathbf{x} - \boldsymbol{\delta})\right)_{i=0}^{m-1} + L(\mathbf{x} - \boldsymbol{\delta}) + \mathbf{c} - \left((\mathbf{x} + \boldsymbol{\delta})^\mathsf{T} P_i (\mathbf{x} + \boldsymbol{\delta})\right)_{i=0}^{m-1} - L(\mathbf{x} + \boldsymbol{\delta}) - \mathbf{c} \quad (6)$$

$$= -2\boldsymbol{\delta}^\mathsf{T}(P_i + P_i^\mathsf{T})\mathbf{x} - 2L\boldsymbol{\delta} = \mathbf{0} \ . \quad (7)$$

This is a linear system of m equations in n variables. Lattice reduction can reduce the length of the solution to approximately $q^{m/n}$. If β is much smaller than this number, one can rely on regular collision-resistance after all. However, if β is larger, collisions are easy to find.

A similar argument can be used to find triple-collisions. In this case, choose small $\boldsymbol{\delta}, \boldsymbol{\gamma}, \boldsymbol{\eta} \xleftarrow{\$} \mathbb{Z}_{2^u}^n$ and require that $\mathcal{P}(\mathbf{x} + \boldsymbol{\delta}) = \mathcal{P}(\mathbf{x} + \boldsymbol{\gamma}) = \mathcal{P}(\mathbf{x} + \boldsymbol{\eta})$. However, in the resulting system of equations the terms $\left(\boldsymbol{\delta}^\mathsf{T} P_i \boldsymbol{\delta}\right)_{i=0}^{m-1}$, $\left(\boldsymbol{\gamma}^\mathsf{T} P_i \boldsymbol{\gamma}\right)_{i=0}^{m-1}$, and $\left(\boldsymbol{\eta}^\mathsf{T} P_i \boldsymbol{\eta}\right)_{i=0}^{m-1}$ do not cancel. Consequently, the system of equations does not necessarily have a solution. Moreover, there are twice as many equations as variables and in the case we are interested in, $\beta \ll q^{2m/n}$. As far as we can tell, the most straightforward strategy for finding triple-collisions involves solving an SSNE instance with at least m equations and n variables, but we cannot prove it. We therefore assume it explicitly. Together with SSNE heuristic, this bounds the success probability of triple-collision finders. Formally, define the triple-collision-resistance game as follows.

Game 3: Triple-Collision-Resistance (3CR)

```
define $\mathsf{Game}^{\mathsf{A}}_{\mathsf{3CR}}(m, n, \beta)$ as:
  | $\mathcal{P} \xleftarrow{\$} (\mathbb{F}_q[\mathbf{x}^{\leq 2}])^m$   $\triangleright$ $m$-tuple of random polynomials of degree at most 2
  | $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \leftarrow \mathsf{A}(\mathcal{P})$
  | return $[\![\mathcal{P}(\mathbf{x}_1) = \mathcal{P}(\mathbf{x}_2) = \mathcal{P}(\mathbf{x}_3)$
  |        $\wedge \|\mathbf{x}_1\| \leq \beta \wedge \|\mathbf{x}_2\| \leq \beta \| \wedge \|\mathbf{x}_3\| \leq \beta]\!]$
```

The triple-collision-resistance assumption states that for all polynomial-time quantum adversaries A, the success probability is negligible, *i.e.*, $\mathsf{Succ}^{\mathsf{3CR}}_{m,n,q,\beta}(\mathsf{A}) \triangleq \Pr[\mathsf{Game}^{\mathsf{A}}_{\mathsf{3CR}}(m, n, \beta) \Rightarrow 1] \leq 2^{-\lambda}$. For the purpose of estimating security levels we employ the SSNE heuristic and assume that $\mathsf{Succ}^{\mathsf{3CR}}_{m,n,q,\beta}(\mathsf{A}) \leq \mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\beta}$.

3.2 Zero-Knowledge Proof System

The following presents Π, a sigma-protocol for proving knowledge of a short solution $\mathbf{x}$ to a nonlinear system of equations $\mathcal{P}(\mathbf{x}) = X \bmod q$, where $\mathcal{P}(\mathbf{x})$ is a list of quadratic forms, *i.e.*, $\mathcal{P}(\mathbf{x}) = (\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1}$ for some list of matrices $P_i \in \mathbb{F}_q^{n \times n}$. In CS notation [7]: $\mathsf{ZKPoK}\{(\mathbf{x}) : (\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1} = X \bmod q \wedge \mathbf{x}^\mathsf{T}\mathbf{x} \leq \sqrt{n} \cdot 2^\ell\}$. Let λ be a root parameter that determines the prime modulus q, an upper bound $u < \log_2 q$, a lower bound $\ell < u$, and the verifier entropy level $e \leq u - \ell$. The first message of the prover consists of two distinct mathematical objects: $Y \leftarrow (\mathbf{y}^\mathsf{T} P_i \mathbf{y})_{i=0}^{m-1}$, $U \leftarrow (\mathbf{x}^\mathsf{T}(P + P^\mathsf{T})\mathbf{y})_{i=0}^{m-1}$ for some randomly chosen $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_{2^u}$. The challenge is a random e-bit number $a \xleftarrow{\$} \mathbb{Z}_{2^e}$. The response $\mathbf{r} \leftarrow a \cdot \mathbf{x} + \mathbf{y}$ allows the prover to verify a relation involving all variables at his disposal: $\forall i \,.\, \mathbf{r}^\mathsf{T} P_i \mathbf{r} \overset{?}{=} a^2 \cdot X + Y + aU \bmod q$. Moreover, if all goes well $\mathbf{r}$ is sufficiently short: $\|\mathbf{r}\| \overset{?}{\leq} 2\sqrt{n} \cdot 2^u = \beta$. The protocol is presented diagrammatically in Figure 4.

3.3 Security

Lemma 1. *Protocol Π is a complete proof system for the relation $\{(\mathbf{x}) : X = (\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1} \bmod q\}$ with completeness error $\varepsilon = 0$.*

P

secret knowledge:
$\mathbf{x} \in \mathbb{Z}_{2^\ell}^n$ such that
$(\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1} = X \bmod q$

$\mathbf{y} \xleftarrow{\$} \mathbb{Z}_{2^u}^2$

$Y \leftarrow (\mathbf{y}^\mathsf{T} P_i \mathbf{y})_{i=0}^{m-1} \bmod q$

$U \leftarrow (\mathbf{x}^\mathsf{T} (P_i + P_i^\mathsf{T}) \mathbf{y})_{i=0}^{m-1} \bmod q$

V

public knowledge:
$P_i \in \mathbb{F}_q^{n \times n}, X \in \mathbb{F}_q^m$

$$\xrightarrow{\quad Y, U \quad}$$

$a \xleftarrow{\$} \mathbb{Z}_{2^e}$

$$\xleftarrow{\quad a \quad}$$

$\mathbf{r} \leftarrow a \cdot \mathbf{x} + \mathbf{y}$

$$\xrightarrow{\quad \mathbf{r} \quad}$$

$a^2 X + Y + aU \overset{?}{=} (\mathbf{r}^\mathsf{T} P_i \mathbf{r})_{i=0}^{m-1} \bmod q$

$\|\mathbf{r}\| \overset{?}{\leq} 2\sqrt{n} \cdot 2^u$

Fig. 4: Protocol Π: a zero-knowledge proof of knowledge of $\mathbf{x}$ in $(\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1}$.

Proof. By construction:

$$\begin{aligned}
\left(\mathbf{r}^\mathsf{T} P_i \mathbf{r}\right)_{i=0}^{m-1} &= \left((a \cdot \mathbf{x} + \mathbf{y})^\mathsf{T} P_i (a \cdot \mathbf{x} + \mathbf{y})\right)_{i=0}^{m-1} \bmod q & (8)\\
&= \left(a^2 \cdot \mathbf{x}^\mathsf{T} P_i \mathbf{x} + a \cdot \mathbf{x}^\mathsf{T} (P_i + P_i^\mathsf{T}) \mathbf{y} + \mathbf{y}^\mathsf{T} P_i \mathbf{y}\right)_{i=0}^{m-1} \bmod q & (9)\\
&= a^2 \cdot X + Y + a \cdot U \bmod q & (10)
\end{aligned}$$

and

$$\begin{aligned}
\|\mathbf{r}\| = \|a \cdot \mathbf{x} + \mathbf{y}\| &\leq a\|\mathbf{x}\| + \|\mathbf{y}\| & (11)\\
&\leq 2^e \cdot \sqrt{n} \cdot 2^\ell + \sqrt{n} \cdot 2^u \leq 2 \cdot \sqrt{n} \cdot 2^u \quad \square & (12)
\end{aligned}$$

Zero-knowledge is a more complicated matter because it is possible for $\mathbf{r}$ to leak some information on the secret $\mathbf{x}$. For example, if one component of $\mathbf{x}$ happens to be very close to 2^ℓ, then with high probability the matching component of $\mathbf{r}$ will be larger than 2^u. To get around this problem, we only consider executions of the protocol where all components of $\mathbf{r}$ are smaller than 2^u, and employing the Fiat-Shamir *with aborts* strategy later on. Restricting attention to the least significant u bits, it is easy to see that $\mathbf{y}$ is a one-time pad on $a \cdot \mathbf{x}$. Since u is slightly bigger than $e + \ell$, the probability of having to abort is small.

Lemma 2. *With the SSNE heuristic, and conditioned on every component r_i of $\mathbf{r}$ being less than 2^u, Protocol Π is a computational honest verifier zero-knowledge proof system with maximum distinguisher advantage*

$$\mathsf{Adv}_{\Pi}^{\mathsf{ZK}}(\mathsf{D}) \leq \mathsf{Adv}_{3m+n, 2n, q, \sqrt{2n} \cdot 2^u}^{\mathsf{DSSNE}} \quad . \tag{13}$$

Proof. The simulator generates the transcript as in the following algorithm.

```
define $\mathsf{S}(X)$ as:
    $a \xleftarrow{\$} \mathbb{Z}_{2^e}$
    $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_{2^u}^n$
    $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_{2^u}^n$
    $Y \leftarrow (\mathbf{y}^\mathsf{T} P_i \mathbf{y})_{i=0}^{m-1} \bmod q$
    $U \leftarrow \big((\mathbf{r}^\mathsf{T} P_i \mathbf{r})_{i=0}^{m-1} - a^2 \cdot X - Y\big) a^{-1} \bmod q$
    return $(Y, U, a, \mathbf{r})$
```

The task of the distinguisher is to distinguish the output of $\mathsf{S}(X)$ from the transcript of $\langle \mathsf{P}(\mathbf{x}, X) \leftrightarrow \mathsf{V}(X) \rangle$, subject to every component r_i of $\mathbf{r}$ being less than 2^u. Since a can be computed from the remaining values, distinguishing authentic from simulated transcripts based on $(Y, U, \mathbf{r})$ is enough. Individually, each element of the tuple $(Y, U, \mathbf{r})$ is distributed identically across both distributions. This is clear in the case of Y and $\mathbf{r}$; the identical distribution of U follows from the equation $U = \left((\mathbf{r}^\mathsf{T} P_i \mathbf{r})_{i=0}^{m-1} - a^2 \cdot X - Y\right) a^{-1} \bmod q$. Moreover, drop any one element and the remainder of the tuple is identically distributed across both distributions, because each of the tuples $(Y, U), (U, \mathbf{r})$ and $(Y, \mathbf{r})$ can be completed with the missing element in a way that could have been the authentic output of the prover. The distinction therefore lies in the joint probability distribution of all tuple elements. To capture this distinction we must cast them into the language of SSNE.

A tuple $(Y, U, a, \mathbf{r})$ is identifiable with a system of equations

$$\begin{cases} a \cdot \mathbf{x} + \mathbf{y} = \mathbf{r} \\ (\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1} = X \bmod q \\ (\mathbf{y}^\mathsf{T} P_i \mathbf{y})_{i=0}^{m-1} = Y \bmod q \\ (\mathbf{x}^\mathsf{T} (P_i + P_i^\mathsf{T}) \mathbf{y})_{i=0}^{m-1} = U \bmod q \enspace . \end{cases} \tag{14}$$

Crucially, this system has a short solution $(\mathbf{x}, \mathbf{y})$ if the transcript was generated authentically. Since there is no overflow, the first equation can be taken modulo q, in which case we have an SSNE system with $m' = 3m + n$, $n' = 2n$, and length bound $\beta = \sqrt{2n} \cdot 2^u$. Using the SSNE heuristic, we conclude that any distinguisher D between S and $\langle \mathsf{P}(\mathbf{x}, X) \leftrightarrow \mathsf{V}(X) \rangle$ has an advantage bounded by

$$\mathsf{Adv}_{\Pi}^{\mathsf{ZK}}(\mathsf{D}) \leq \mathsf{Adv}_{\mathsf{S}(X), \langle \mathsf{P}(\mathbf{x}, X) \leftrightarrow \mathsf{V}(X) \rangle}^{\mathsf{dist}}(\mathsf{D}) \leq \mathsf{Adv}_{3m+n, 2n, q, \sqrt{2n} \cdot 2^u}^{\mathsf{DSSNE}} \enspace . \quad \square \tag{15}$$

The next property is soundness, *i.e.*, the inability of a computationally bounded adversary to authenticate with respect to a public key that is invalid. By design this property is guaranteed if SSNE is hard.

Lemma 3. *In the SSNE heuristic, protocol Π is sound against quantum polynomial time adversaries, with soundness error $\sigma \leq \mathsf{Succ}_{m,n,q,\sqrt{n} \cdot 2^u}^{\mathsf{SSSNE}}$.*

Proof. Since $\nexists \mathbf{x} \in \mathbb{Z}_q^n \, . \, (\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1} = X \, \wedge \, \|\mathbf{x}\| < \sqrt{n} \cdot 2^\ell$, the task of finding an $\mathbf{r} \in \mathbb{Z}_q^n$ such that $(\mathbf{r}^\mathsf{T} P_i \mathbf{r})_{i=0}^{m-1} = a^2 \cdot X + Y + a \cdot U \bmod q$ for a random a

and $\|\mathbf{r}\| \leq 2\sqrt{n}2^\ell$, implies a solution to search SSNE if it has a solution, and is impossible if it does not. The SSNE problem is defined with respect to $m' = m$ equations, $n' = n$ variables, and length bound $\beta = 2\sqrt{n} \cdot 2^u$. So the soundness error is bounded by the optimal success probability of search SSNE, namely $\sigma \leq \mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,2\sqrt{n}\cdot 2^u}$. □

Witness-extractability is less straightforwardly proven than soundness. Given two *honestly generated* transcripts $T_1 = (Y, U,\ a_1, \mathbf{r}_1)$ and $T_2 = (Y, U, a_2, \mathbf{r}_2)$, with the same first message but different challenges, the witness $\mathbf{x}$ can be found as $\mathbf{x} \leftarrow (\mathbf{r}_2 - \mathbf{r}_1)/(a_2 - a_1)$. However, this extractability holds only for provers that follow the protocol. The verifier should not have to assume that the prover is behaving honestly; rather, he should be convinced of that fact precisely by participating in the protocol. The following lemma shows classical witness-extractability. Quantum witness-extractability remains an open question.

Lemma 4. *In the SSNE heuristic, protocol Π is a classical proof of knowledge: for all polynomial-time adversaries* B*, the success probability is bounded by*

$$\Pr[\mathsf{out}_\mathsf{V}(\langle \mathsf{B} \leftrightarrow \mathsf{V} \rangle) = 1] \leq \sqrt[5]{\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\sqrt{n}\cdot 2^\ell}} + 3\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\sqrt{n}\cdot 2^\ell} \ . \tag{16}$$

Proof. The extractor E proceeds as follows. He simulates the forger B until B outputs the first protocol message (Y, U). At this point, X and Y are fixed. Moreover, the forger *knows* at most two preimages each to X and Y. Formally, *knowledge* in this context means that all polynomial-time extractors F^B that output preimages to X or Y, jointly output a set of at most two preimages for X and at most two for Y. If some extractor F^B did output a third preimage for X or for Y, then it can be used to win the triple-collision-resistance game. The winning probability for this task is $\mathsf{Succ}^{\mathsf{3CR}}_{m,n,q,\sqrt{n}\cdot 2^\ell}(\mathsf{F}^\mathsf{B}) \leq \mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\sqrt{n}\cdot 2^\ell}$. Conditioned on this event not occurring, we can speak of *the* two preimages $\mathbf{x}_1, \mathbf{x}_2$ of X and $\mathbf{y}_1, \mathbf{y}_2$ of Y. The following argument holds even if there is only one preimage for X or Y or both.

The response $\mathbf{r}$ must be of the form

$$\mathbf{r} = a \cdot \mathbf{x}_i + \mathbf{y}_j \quad \text{with} \quad i, j \in \{1, 2\} \ . \tag{17}$$

If it is not, then it is either an invalid response or the forger has managed to find a short solution to an SSNE system that is randomized by a. Since the forger is polynomial-time, its success probability in the latter task is bounded by $\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,2\sqrt{n}\cdot 2^u}$. Therefore, conditioned on this event not occurring, the proof-forger B must respond with an $\mathbf{r}$ of the form of Eqn. 17.

The extractor forks into five branches and feeds a different random challenge a_k to the proof-forger B in each branch. Each proof-forger B outputs a valid response $\mathbf{r}_k$ of the form of Eqn. 17. At least one pair $(\mathbf{x}_i, \mathbf{y}_j)$ must be reused.

For each pair $(\mathbf{r}_k, \mathbf{r}_\ell)$ of responses, the extractor E computes $\mathbf{x} \leftarrow (\mathbf{r}_k - \mathbf{r}_l)/(a_k - a_l)$. At least one such pair uses the same $\mathbf{x}_i$ and $\mathbf{y}_j$ for both $\mathbf{r}_k$ and $\mathbf{r}_\ell$, which guarantees that $\mathbf{x}_i = (\mathbf{r}_k - \mathbf{r}_\ell)/(c_k - c_\ell)$ without modular reduction.

By computing this value for all pairs $(\mathbf{r}_k, \mathbf{r}_\ell)$, the extractor E finds at least one $\mathbf{x}$ such that $\mathcal{P}(\mathbf{x}) = X$.

The success of the extractor E depends on the non-occurrence of the events "B finds a triple-collision for X", "B finds a triple-collision for Y", and "B finds a wholly new solution $\mathbf{r}$ to the SSNE problem". Let $\mathcal{E}$ denote the occurrence of any of these events. The probability of $\mathcal{E}$ is bounded by $2\mathsf{Succ}^{\mathsf{3CR}}_{m,n,q,\sqrt{n}\cdot 2^\ell} + \mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\sqrt{n}\cdot 2^u} \leq 3\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\sqrt{n}\cdot 2^\ell}$.

The event "E outputs the witness $\mathbf{x}$" is equivalent with "B wins rounds 1–5 and not $\mathcal{E}$". This gives the following.

$$\Pr[\mathsf{B}\,\mathit{wins\ rounds}\,1\text{–}5\ \wedge\ \neg\mathcal{E}] = \Pr[\mathsf{B}\,\mathit{wins\ round}\,1\ \wedge\ \cdots\ \wedge\ \mathsf{B}\,\mathit{wins\ round}\,5\ \wedge\ \neg\mathcal{E}] \tag{18}$$

$$= \Pr[(\mathsf{B}\,\mathit{wins\ round}\,1\ \wedge\ \neg\mathcal{E}) \wedge \cdots \wedge (\mathsf{B}\,\mathit{wins\ round}\,5\ \wedge\ \neg\mathcal{E})] \tag{19}$$

For a given random tape, the events ($\mathsf{B}\,\mathit{wins\ round}\ i\ \wedge\ \neg\mathcal{E}$) are independent because they are a deterministic function of an independently drawn variable, a_i. We assume without loss of generality that the random coins for B are subsumed into its state and hence replicated each round.

$$\ldots = \sum_{\mathit{coins}} \Pr[\mathit{coins}] \cdot \prod_{i=1}^{5} \Pr[\mathsf{B}\,\mathit{wins\ round}\,i\ \wedge\ \neg\mathcal{E} \mid \mathit{coins}] \tag{20}$$

$$= \sum_{\mathit{coins}} \Pr[\mathit{coins}] \cdot \left(\Pr[\mathsf{B}\,\mathit{wins}\ \wedge\ \neg\mathcal{E} \mid \mathit{coins}]\right)^5 \tag{21}$$

$$\geq \left(\sum_{\mathit{coins}} \Pr[\mathit{coins}] \cdot \Pr[\mathsf{B}\,\mathit{wins}\ \wedge\ \neg\mathcal{E} \mid \mathit{coins}] \right)^5 \tag{22}$$

$$= \Pr[\mathsf{B}\,\mathit{wins}\ \wedge\ \neg\mathcal{E}]^5 \tag{23}$$

The inequality holds due to Jensen's inequality, which states that for any convex function f, coefficients $\lambda_1, \ldots, \lambda_n \in [0,1]$ with $\sum_{i=1}^n \lambda_i = 1$, $f(\sum_i^n \lambda_i t) \leq \sum_i^n \lambda_i f(t)$.

Since a successful extractor is finding a preimage under $\mathcal{P}(\mathbf{x})$ of X, the extractor's success probability is bounded by $\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,2\sqrt{n}\cdot 2^\ell}$, meaning that

$$\Pr[\mathsf{B}\,\mathit{wins}] \leq \Pr[\mathsf{B}\,\mathit{wins}\ \wedge \neg\mathcal{E}] + \Pr[\mathcal{E}] \tag{24}$$

$$\leq \sqrt[5]{\Pr[\mathsf{E}\,\mathit{success}]} + 3\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,2\sqrt{n}\cdot 2^\ell} \tag{25}$$

$$\leq \sqrt[5]{\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,\sqrt{n}\cdot 2^\ell} + 3\mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,2\sqrt{n}\cdot 2^\ell}}\ . \tag{26}$$

□

The security bound involves a fifth-root security degradation, in contrast to the square-root degradation in the analysis of Bellare and Neven [3, §3]. This exacerbated degradation is due to the need to fork into five branches, whereas two branches are enough in the standard case. Nevertheless, the bound is asymptotically sound, meaning that if no polynomial-time algorithm solves search-SSNE with more than a negligible success probability, then no polynomial-time adversary fools the verifier with more than a negligible probability.

3.4 Quantum Soundness

The previous discussion covers classical security only, even though the protocol is presented as a system for *post-quantum* security. Correctness and honest-verifier zero-knowledgeness are straightforwardly lifted to the quantum adversarial model, as is soundness. However, knowledge-soundness, or witness-extractability, is much trickier. We survey here two strategies for generating a provably secure (in the QROM) signature scheme from a zero-knowledge proof, and argue in each case that its application to our protocol is inadvisable. We are not unique in such an argument in favor of the Fiat-Shamir transform at the cost of a QROM proof. The same motivation appears explicitly in Dilithium [15] and MQDSS [8], and implicitly in many other proposals.

Unruh Transform. The Unruh transform [32] turns a Σ-protocol into a non-interactive zero-knowledge quantum proof of knowledge in the quantum random oracle model. It achieves this by relying on length-preserving commitments which, when instantiated by the extractor, can be efficiently inverted with the forger being none the wiser.

The prover commits to many complete branches of protocol executions, where the path is determined by the verifier's challenge. The hash of this commitment then determines which branch will be opened, thus revealing only one randomly chosen transcript. However, the extractor who provides the forger with a backdoored view of the random oracle, can obtain all transcripts in the entire tree. If the original protocol has *computational special soundness*, then the extractor can compute the witness from these transcripts.

However, the Unruh transform comes at a significant cost. The motivating promise of using the SSNE problem in the first place is the small size of the transcript and the high speed of operations. The speed objective is undermined by the need of the Unruh transform to run enough protocol executions to reduce the soundness error to a cryptographically insignificant quantity; and the size objective is undermined when all transcripts are committed to with a length-preserving hash function.

UKLS Deterministic Fiat-Shamir. Unruh shows that the Fiat-Shamir transform preserves soundness in the quantum random oracle model and proposes to generate post-quantum signature schemes based on plain soundness instead of knowledge-soundness [33]. In particular, the security proof can bypass the need to show witness-extractability when no forger can feasibly generate a proof for a fake public key. In addition to that, fake public keys must exist and they must be computationally infeasible to distinguish from authentic ones. The strategy received a concrete treatment by Kiltz, Lyubashevsky and Schaffner (KLS) in the context of lossy identification schemes [20].

In the case of our protocol, we find that it is unclear whether fake public keys can exist. A public key is a tuple of finite field elements $X \in \mathbb{F}_q^m$ and it might have been produced authentically by a key generation algorithm when $\exists \mathbf{x} \in \mathbb{Z}_{2^\ell}^n \, . \, (\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1} = X$. Taking the length constraint into account, there are ℓn indeterminate bits in $\mathbf{x}$, and $m \log_2 q < 2m\ell$ constraining bitwise equations, and so there are roughly $2^{\ell n - m \log_2 q}$ different values for $\mathbf{x}$ such that $X = \left(\mathbf{x}^\mathsf{T} P_i \mathbf{x}\right)_{i=0}^{m-1} \bmod q$. Changing the parameters so that $\ell n < m \log_2 q$ (which implies $n < 2m$) is not compatible with the requirement for zero-knowledge, namely $n' = 2n > m' = 3m + n$ (which implies $n > 3m$). So the UKLS technique cannot be made to work because it would eventually generate a signature scheme whose signatures betray knowledge of the secret key.

4 Signature Scheme

4.1 Description

We instantiate the protocol with parameters $m = 6$ and $n = 19$. Moreover, we set q, ℓ, u, e as a function of the security parameter via $\log_2 q \approx 17\lambda$, $\ell = 8.55\lambda$, $u = 9.60\lambda$, $e = \lambda$. For these parameters the soundness and zero-knowledge properties achieve a security level of λ bits against classical attacks. To see this, consider each of the terms in the security statements. With respect to the knowledge-soundness property, we are content with the asymptotic security, implicitly assuming that an attack on knowledge-soundness requires solving the SSSNE instance associated with breaking soundness.

- $\mathsf{Succ}^{\mathsf{SSSNE}}_{6,19,q,2\sqrt{19}\cdot 2^u}$. This term captures the soundness. All design principles are satisfied. With respect to design principle number 6, we find that $o = 5$ and hence $\frac{n-o+m}{n+1} \log_2 q \approx \frac{20}{20} 17\lambda = 17\lambda > \frac{\lambda}{n} + \log_2 \beta = (\lambda/19 + u\lambda + 1)\lambda \approx 9.65\lambda$. So design principle 6 is satisfied.
  The margin is smallest with respect to design principle number 3. In this case we have the squared length of the secret key $\mathbf{x}$ is $\|\mathbf{x}\|^2 \approx \sqrt{n} \cdot 2^{2u} \approx 2^{19.23\lambda}$, which is only a little bit larger than $q \approx 2^{17\lambda}$. The first approximation comes from the fact that the components of $\mathbf{x}$ are sampled from $\mathbb{Z}_{2^u}$, and so design principle 3 is satisfied with overwhelming probability.
- $\mathsf{Adv}^{\mathsf{SSNE}}_{37,38,q,\sqrt{38}\cdot 2^u}$. This term captures the security of the zero-knowledge property. Design principle 6 does not apply and we find that the smallest margin is in fact for design principle 3. However, since both $\|\mathbf{x}\| > \sqrt{q}$ and $\|\mathbf{y}\| > \|\mathbf{x}\|$ with overwhelming probability, the length of the solution $(\mathbf{x}, \mathbf{y})$ is larger than $\sqrt{q}$ with overwhelming probability as well.
- The challenge coming from the verifier consists of e bits of entropy. This term is hidden by $\mathsf{Succ}^{\mathsf{SSSNE}}_{6,19,q,\sqrt{19}\cdot 2^u}$ but it should be large enough to make the case for the SSNE heuristic in the knowledge-soundness proof compelling. Moreover, the term 2^{-e} appears explicitly in the Bellare-Neven formula for the security degradation as a consequence of applying the Forking lemma.

The signature scheme Σ follows directly from applying the deterministic Fiat-Shamir transform with aborts and derandomization to the identification scheme Π, along with a hash function H and a pseudorandom generator G. Additionally, we assume access to a function $\mathsf{sample}(\cdot,\cdot)$ that deterministically samples from the space given as first argument using the coins given as the second. Symbolically we have $\Sigma = \mathsf{DFSAD}[\Pi, \mathsf{leaks}, \kappa, \mathsf{H}, \mathsf{G}]$ and functions defined as follows. In this pseudocode, G is a PRG; $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$ is a hash function; and leaks is defined as

```
1.  define $\Sigma.\mathsf{Sign}(sk, m)$ as:
2.  |  $coins_1, coins_2 \leftarrow \mathsf{G}(sk)$
3.  |  $\{coins_{3,j}\}_{j=0}^{\kappa-1} \leftarrow \mathsf{G}(sk \| m)$
4.  |  $\{P_i\}_{i=0}^{m-1} \leftarrow \mathsf{sample}((\mathbb{F}_q^{n\times n})^m,$
    |                                  $coins_1)$
5.  |  $\mathbf{x} \leftarrow \mathsf{sample}(\mathbb{Z}_{2^\ell}^n, coins_2)$
6.  |  $X \leftarrow (\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1} \bmod q$
7.  |  for $j \in \{0, \dots, \kappa-1\}$ do:
8.  |  |  $\mathbf{y} \leftarrow \mathsf{sample}(\mathbb{Z}_{2^u}^n, coins_{3,j})$
9.  |  |  $Y \leftarrow (\mathbf{y}^\mathsf{T} P_i \mathbf{y})_{i=0}^{m-1} \bmod q$
10. |  |  $U \leftarrow \mathbf{x}^\mathsf{T}(P_i + P_i^\mathsf{T})\mathbf{y} \bmod q$
11. |  |  $a \leftarrow \mathsf{H}(X \| Y \| U \| m)$
12. |  |  $\mathbf{r} \leftarrow a \cdot \mathbf{x} + \mathbf{y}$
13. |  |  $s \leftarrow (Y, U, \mathbf{r})$
14. |  |  if not $\mathsf{leaks}(Y, U, a, \mathbf{r})$ then:
15. |  |  |  return $s$
16. |  return $\perp$
```

```
1. define $\Sigma.\mathsf{KeyGen}(1^\lambda)$ as:
2. |  $seed \xleftarrow{\$} \{0,1\}^\lambda$
3. |  $coins_1, coins_2 \leftarrow \mathsf{G}(seed)$
4. |  $\{P_i\}_{i=0}^{m-1} \leftarrow \mathsf{sample}((\mathbb{F}_q^{n\times n})^m, coins_1)$
5. |  $\mathbf{x} \leftarrow \mathsf{sample}(\mathbb{Z}_{2^\ell}^n, coins_2)$
6. |  $X \leftarrow (\mathbf{x}^\mathsf{T} P_i \mathbf{x})_{i=0}^{m-1} \bmod q$
7. |  $pk \leftarrow (coins_1, X)$
8. |  $sk \leftarrow seed$
9. |  return $sk, pk$
```

```
1. define $\Sigma.\mathsf{Verify}(pk, m, s)$ as:
2. |  $coins_1, X \leftarrow pk$
3. |  $\{P_i\}_{i=0}^{m-1} \leftarrow \mathsf{sample}((\mathbb{F}_q^{n\times n})^m, coins_1)$
4. |  $(Y, U, \mathbf{r}) \leftarrow s$
5. |  $a \leftarrow \mathsf{H}(X \| Y \| U \| m)$
6. |  return $[\![\|\mathbf{r}\| \leq 2 \cdot \sqrt{n} \cdot 2^u]\!] \wedge$
   |     $[\![ a^2 X + Y + aU = (\mathbf{r}^\mathsf{T} P_i \mathbf{r})_{i=0}^{m-1} \bmod q ]\!]$
```

$$\mathsf{leaks}(Y, U, a, \mathbf{r}) = \begin{cases} 1 & \text{if for all } i \in \{0, \dots, n\}, r_i < 2^u \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

The parameter κ is set so as to make the probability of signature generation failure cryptographically negligible, *i.e.*, less than $2^{-\lambda}$. This probability is determined as follows. Since ax_i is at most $\ell + e$ bits and y_i consists of u bits, a failure for this component occurs when the top $u - \ell - e \approx 0.05\lambda$ bits of y_i are set and a carry chain flips the next one. The probability of this event is $2^{e+\ell-u}$. A single test of line 14 in $\Sigma.\mathsf{Sign}$ fails if any one of the $n = 19$ components triggers failure. Conservatively modeling them as independent events, we find that the probability of a single successful leaks test is at least $(1 - 2^{e+\ell-u})^n$, and the probability of a single failing leaks test is at most one minus this quantity. Signature generation failure entails κ individual leaks test failures, so we find

$$\Pr[\Sigma.\mathsf{Sign}(sk, m) \Rightarrow \perp] \leq (1 - (1 - 2^{e+\ell-u})^n)^\kappa \ . \tag{28}$$

Rather than finding an exact formula for the appropriate value of κ such that $\Pr[\Sigma.\mathsf{Sign}(sk, m) \Rightarrow \perp]$ is smaller than $2^{-\lambda}$, we chose to write a script to compute it numerically in terms of the concrete parameter values. The resulting plot is shown in Fig. 5. In this figure, κ drops to 60 at around $\lambda \approx 125$, after which

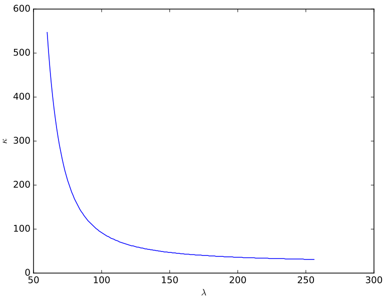


Fig. 5: Value of κ for cryptographically negligible failure rate.

point it continues to drop but very slowly. Therefore, setting $\kappa = 60$ regardless of λ guarantees that for all targeted security levels the signature generation failure probability is small enough.

4.2 Security

Theorem 1. *In the SSNE heuristic, the signature scheme* $\Sigma = \mathsf{DFSAD}[\Pi, \mathsf{leaks}, \kappa, \mathsf{H}, \mathsf{G}]$ *with the parameters set as* $\lceil \log_2 q \rceil = 17\lambda$*,* $m = 6$*,* $n = 19$*,* $u = 9.60\lambda$*,* $\ell = 8.55\lambda$*,* $e = \lambda$*, is secure in the SUF-CMA and random oracle models. In particular, for any polynomial time adversary* A *making* Q_H *queries to the hashing oracle in the SUF-CMA game, the insecurity is bounded by*

$$\begin{aligned}\mathsf{InSec}^{\mathsf{SUF\text{-}CMA}}_{\Sigma}(\mathsf{A}) \leq \mathsf{Succ}^{\mathsf{SSSNE}}_{m,n,q,2\sqrt{n}\cdot 2^u} + \mathsf{Adv}^{\mathsf{PRG}}_{\mathsf{G}} + \mathsf{Adv}^{\mathsf{DSSNE}}_{3m+n,2n,q,\sqrt{2n}\cdot 2^u} + 2^{-e} + 2^{-\lambda} \\ + Q_\mathsf{H}\left(\sqrt[5]{\mathsf{Succ}^{\mathsf{SSNE}}_{m,n,q,\sqrt{n}\cdot 2^\ell}} + 3\mathsf{Succ}^{\mathsf{SSNE}}_{m,n,q,\sqrt{n}\cdot 2^\ell}\right) \ . \qquad (29)\end{aligned}$$

A security proof can be found in Appendix B. The obtained bound is rather loose due to the Forking Lemma and the resulting fifth-root degradation. Nevertheless, we know of no better attack on knowledge-soundness than solving one of the search-SSNE problems. It seems reasonable, therefore, to assert that the fifth-root degradation is an artifact of the proof technique rather than an indication of inherent insecurity. We note that this bound, like the bound on knowledge-soundness, remains asymptotically sound.

4.3 Performance

The parameters have been chosen to guarantee λ bits of security against attacks that involve search-SSNE or decision-SSNE. Therefore, setting $\lambda = 128, 192, 256$ instantly gives us concrete parameter sets targeting exactly those security levels against classical attacks. Assuming a quadratic speedup on quantum computers due to Grover, we obtain half this security level in a post-quantum setting.

Table 1 compares the public key and signature size of our signature scheme with those of several representative submission to the NIST competition [24]. On the first three lines, the omitted parameters are as defined at the start of Sect. 4.1.

Table 1: Comparison of our signature scheme to several NIST proposals.

scheme	PQ security level	parameters	public key size	signature size
ours	64	$q = 2^{17\cdot 128} - 1833$	1.61 kB	6.04 kB
ours	96	$q = 2^{17\cdot 192} - 1703$	2.41 kB	9.07 kB
ours	128	$q = 2^{17\cdot 256} - 9663$	3.22 kB	12.09 kB
Dilithium	125	recommended	1.44 kB	2.64 kB
SPHINCS+	128	sphincs-sha256-256s	64 bytes	29.09 kB
LUOV	128	LUOV-8-117-404	98.6 kB	521 bytes
LUOV	128	LUOV-80-86-399	39.3 kB	4.7 kB

As a proof of concept we implemented the signature scheme in SageMath. We omit a comparison to the C implementations of the NIST candidates as such a comparison would fail to match apples to apples. While much slower, a Sage implementation is capable of validating the design in terms of functionality and its rudimentary timing results can provide some indication as to whether the scheme can be made practicable. These timing results are shown in Table 2. Additionally, we observe a constant abortion rate of zero across all parameter ranges, indicating that perhaps setting $\kappa = 60$ is overkill.

Table 2: Timing results from a Sage implementation of the signature scheme.

PQ security level	KeyGen	Sign	Verify
64	2.59 s	3.89 s	2.62 s
96	3.92 s	5.83 s	4.02 s
128	5.22 s	7.78 s	6.30 s

5 Conclusion

This paper presents a zero-knowledge proof system and signature scheme whose security relies on the short solutions to nonlinear equations (SSNE) problem. The zero-knowledge proof resembles the protocol of Schnorr [28], but in contrast to Schnorr's protocol, no quantum attack is known against it. While the signature scheme's security proof is valid only in the classical random oracle model, no quantum attack is known to defeat the construction. From this point of view, our signature scheme ought to be classified as post-quantum.

The most important difference from other post-quantum signature schemes is the reliance on different hard problems. It is therefore unlikely that generic attacks on other post-quantum cryptosystems and hard problems will affect the security of our scheme. The construction of a signature scheme relying on SSNE answers a question posed by Szepieniec and Preneel in their discussion about the SSNE problem [30], which merely conjectured that possibility.

The main motivation for the construction of cryptosystems from SSNE was to improve on their bandwidth requirements. Since generic attacks on MQ and SIS fail against SSNE, no attack is known to outperform brute force for appropriately chosen parameters. It is therefore possible to obtain a security level that scales linearly with the size of the problem's representation. In particular, this means that for our signature scheme, both public key and signature size scale linearly with the security parameter. We are unaware of other post-quantum signature schemes that attain this optimal asymptotic behavior. Nevertheless, for Earthly security levels, our scheme's bandwidth requirements are at best comparable to those of other post-quantum signature schemes.

The chief reason for this large hidden constant factor is the protocol's reliance on SSNE for *both* zero-knowledge *and* soundness properties. This simultaneous requirement forces us to choose large values for both m and n. Moreover, after fixing m and n, the ALHA attack [30] forces the solutions' size differential $u-\ell$ to remain within relatively slim margins. Fitting $e = \lambda$ bits of entropy between these extremes in turn mandates a large q. While an SSNE problem with parameters as small as $n = 2, m = 1, q = 2^{256}$ can be enough to target a sizeable security level, it does not seem possible to generate a secure zero-knowledge proof system with parameters this small. A interesting question for future research is therefore whether the SSNE problem can be exchanged for another hard problem for the security of *either* the zero-knowledge property *or* soundness.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: ACM STOC 1996. pp. 99–108. ACM (1996)
2. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: FOCS 2014. pp. 474–483. IEEE Computer Society (2014)
3. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: ACM CCS 2006. pp. 390–399. ACM (2006)
4. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS '93. pp. 62–73. ACM (1993)
5. Bernstein, D.J.: Post-quantum cryptography. In: Encyclopedia of Cryptography and Security, 2nd Ed., pp. 949–950 (2011)
6. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: ASIACRYPT 2011. pp. 41–69. LNCS, Springer (2011)
7. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups (extended abstract). In: CRYPTO '97. pp. 410–424. LNCS, Springer (1997)
8. Chen, M., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass MQ -based identification to MQ -based signatures. In: ASIACRYPT 2016 II. pp. 135–165. LNCS, Springer (2016)

9. Coppersmith, D.: Finding a small root of a bivariate integer equation; factoring with high bits known. In: EUROCRYPT '96. pp. 178–189. LNCS, Springer (1996)
10. Coppersmith, D.: Finding a small root of a univariate modular equation. In: EUROCRYPT '96. pp. 155–165. LNCS, Springer (1996)
11. Coppersmith, D.: Finding small solutions to small degree polynomials. In: CaLC 2001. pp. 20–31. LNCS, Springer (2001)
12. Coron, J.: Finding small roots of bivariate integer polynomial equations revisited. In: EUROCRYPT 2004. pp. 492–505. LNCS, Springer (2004)
13. Coron, J.: Finding small roots of bivariate integer polynomial equations: A direct approach. In: CRYPTO 2007. pp. 379–394. LNCS, Springer (2007)
14. Ding, J., Yang, B.Y.: Multivariate Public Key Cryptography. Springer Berlin Heidelberg (2009), Post-Quantum Cryptography, ch. 6, pp 193-241
15. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR TCHES 2018(1), 238–268 (2018)
16. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: CRYPTO '86. pp. 186–194. LNCS, Springer (1986)
17. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988)
18. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Cryptography and Coding IMA. pp. 131–142. LNCS, Springer (1997)
19. Jutla, C.S.: On finding small solutions of modular multivariate polynomial equations. In: EUROCRYPT '98. pp. 158–170. LNCS, Springer (1998)
20. Kiltz, E., Lyubashevsky, V., Schaffner, C.: A Concrete Treatment of Fiat-Shamir Signatures in the Quantum Random-Oracle Model. Cryptology ePrint Archive, Report 2017/916 (2017), https://eprint.iacr.org/2017/916
21. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of computation 48(177), 203–209 (1987)
22. Lyubashevsky, V.: Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: ASIACRYPT 2009. pp. 598–616. LNCS, Springer (2009)
23. Miller, V.S.: Use of elliptic curves in cryptography. In: CRYPTO '85. pp. 417–426. LNCS, Springer (1985)
24. NIST: Post-quantum crypto standardization (2018), http://csrc.nist.gov/groups/ST/post-quantum-crypto/
25. Pointcheval, D., Stern, J.: EUROCRYPT '96. pp. 387–398. LNCS, Springer (1996)
26. Ritzenhofen, M.: On efficiently calculating small solutions of systems of polynomial equations: lattice-based methods and applications to cryptography. Ph.D. thesis, Ruhr University Bochum (2010)
27. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (1978)
28. Schnorr, C.: Efficient identification and signatures for smart cards. In: CRYPTO '89. pp. 239–252. LNCS, Springer (1989)
29. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: FOCS 1994. pp. 124–134. LNCS, Springer (1994)
30. Szepieniec, A., Preneel, B.: Short solutions to nonlinear systems of equations. In: NuTMiC 2017. pp. 71–90. LNCS, Springer (2017)
31. Unruh, D.: Quantum proofs of knowledge. In: EUROCRYPT 2012. pp. 135–152. LNCS, Springer (2012)
32. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: EUROCRYPT 2015 II. pp. 755–784. LNCS, Springer (2015)
33. Unruh, D.: Post-quantum security of Fiat-Shamir. In: ASIACRYPT 2017 I. pp. 65–95. LNCS, Springer (2017)

A Zero-Knowledge and Digital Signature Schemes

A.1 Zero-Knowledge Proofs

An *interactive proof system* $\mathit{\Pi} = (\mathsf{P}, \mathsf{V})$ for a language $\mathcal{L} \in \mathbf{NP}$ is a protocol between a pair of interactive polynomial-time Turing machines (called the *prover* P and *verifier* V, respectively) whose common input is a string $\ell \in \{0,1\}^*$. The secret information for the prover is a witness $v \in \{0,1\}^*$ that certifies that $\ell \in \mathcal{L}$, *i.e.*, $\mathcal{R}_{\mathcal{L}}(\ell, v) = 1$. After running the protocol (we denote this event by $\mathsf{V}(\ell) \leftrightarrow \mathsf{P}(v, \ell)$), the verifier outputs a single bit $b \leftarrow \mathsf{out}_{\mathsf{V}}(\mathsf{V}(\ell) \leftrightarrow \mathsf{P}(\ell, v))$, which is 1 if he accepts and 0 if he rejects. The *transcript* $T \leftarrow \langle \mathsf{P}(v, \ell), \mathsf{V}(\ell) \rangle$ consists of all messages sent between the two parties and we denote whether it is an *accepting* transcript for ℓ by the predicate $V(\ell, T)$ and by definition $V(\ell, T) = b \leftarrow \mathsf{out}_{\mathsf{V}}(\mathsf{V}(\ell) \leftrightarrow \mathsf{P}(\ell, v))$. For the purpose of this paper, we aim to satisfy the following three properties:

1. *Completeness.* For every $\ell \in \mathcal{L}$ and matching witness v, an honest prover will likely convince an honest verifier:

$$\begin{aligned} &\forall \ell \in \{0,1\}^*, v \in \{0,1\}^* \, . \\ &\mathcal{R}_{\mathcal{L}}(\ell, v) = 1 \implies \Pr[b = 1 \,|\, b \leftarrow \mathsf{out}_{\mathsf{V}}(\mathsf{V}(\ell) \leftrightarrow \mathsf{P}(\ell, v))] \geq 1 - \varepsilon \ . \end{aligned}$$

   In this expression ε represents the *completeness error* and should be a negligible function of $|\ell|$, *i.e.*, $\varepsilon \leq \epsilon(|\ell|)$.

2. *Soundness.* For every $\ell \notin \mathcal{L}$ no prover B is likely to convince the verifier:

$$\forall \ell \notin \mathcal{L} \, . \, \forall \mathsf{B} \, . \Pr[b = 1 \,|\, b \leftarrow \mathsf{out}_{\mathsf{V}}(\mathsf{V}(\ell) \leftrightarrow \mathsf{B}(\ell))] \leq \sigma \ .$$

   The quantity σ represents the *soundness error* and should be small but not necessarily negligible.

2⋆ *Witness-extractability or knowledge-soundness.* There is a polynomial-time extractor machine E who, given black-box access to any successful prover B, can compute the witness v with noticeable probability.

$$\exists \mathsf{E} \, . \, \forall \mathsf{B} \, . \Pr[\mathsf{out}_{\mathsf{V}}(\mathsf{V}(\ell) \leftrightarrow \mathsf{B}(\ell)) = 1] \geq \varsigma \implies \Pr[\mathcal{R}_{\mathcal{L}}(\ell, v) = 1 \,|\, v \leftarrow \mathsf{E}^{\mathsf{B}}()] \geq \nu(\lambda) \ .$$

   Phrased differently, if the extractor fails to produce the witness, then the prover's success probability is upper-bounded by the *knowledge error* ς, which should also be small but not necessarily negligible.

3. *Honest-verifier zero-knowledge (HVZK).* There exists a polynomial-time simulator S capable of producing a transcript $T \leftarrow \mathsf{S}(\ell)$ of the protocol without knowledge of the witness v such that T is indistinguishable from authentic transcripts. Indistinguishability is defined with respect to all polynomial-time distinguishers D having at most a negligible *distinguishing advantage*, *i.e.*, $\mathsf{Adv}^{\mathsf{ZK}}_{\mathit{\Pi}}(\mathsf{D}) \leq \epsilon(\lambda)$, where

$$\mathsf{Adv}^{\mathsf{ZK}}_{\mathit{\Pi}}(\mathsf{D}) \triangleq \left| \Pr[\mathsf{D}(T) = 1 \,|\, T \leftarrow \langle \mathsf{P}(\ell, v), \mathsf{V}(\ell) \rangle] - \Pr[\mathsf{D}(T) = 1 \,|\, T \leftarrow \mathsf{S}(\ell)] \right| \ .$$

An *identification scheme* is a zero-knowledge proof system that satisfies the above properties and is furthermore adjoined with a public key generator algorithm KeyGen for $\mathcal{L}$ that outputs a pair of keys (sk, pk) such that sk is a witness for $pk \in \mathcal{L}$, *i.e.*, $\mathcal{R}_{\mathcal{L}}(pk, sk) = 1$. In this context, an attack on knowledge-soundness is known as an *impersonation attack*.

A.2 Signature Schemes

A *digital signature scheme* Σ is a triple of polynomial-time algorithms $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ with the following properties.

- $\mathsf{KeyGen}(1^\lambda)$ outputs a secret and public key pair (sk, pk).
- $\mathsf{Sign}(sk, d)$ takes a secret key sk and a message (or document) $d \in \{0,1\}^*$ and outputs a signature s on that message.
- $\mathsf{Verify}(pk, d, s)$ takes a public key pk, message d, and signature s; and outputs True or False depending on whether the signature is valid or not.
- For all messages $d \in \{0,1\}^*$ and when $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, a signature on d generated with sk will be valid under pk with overwhelming probability:

$$\Pr[\mathsf{Verify}(pk, d, s) \Rightarrow \mathsf{True} \mid s \leftarrow \mathsf{Sign}(sk, d)] \geq 1 - \varepsilon \ .$$

  In this expression we call ε the *correctness error* and we require this quantity to be negligible $\varepsilon \leq \epsilon(\lambda)$.

Security of a signature scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is defined with respect to the *existential unforgeability under chosen message attack (EUF-CMA)* [17] game, or even with respect to the strictly stronger *strong unforgeability under chosen message attack (SUF-CMA)* game. Informally, the adversary A in the EUF-CMA game, who is allowed to query a signature oracle, wins if he can produce a valid message-signature pair where the message was not queried. The SUF-CMA game relaxes the winning condition by considering the adversary to win when the message-signature pair output is valid and not identical to any query-response pair. The games are formally defined by the pseudocode of Games 6 and 7.

Game 6: EUF-CMA

```
define $\mathsf{Game}^{\mathsf{A}}_{\mathsf{EUF\text{-}CMA}}(1^\lambda)$ as:
    $(sk, pk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
    $\mathcal{S} \leftarrow \varnothing$
    define $\mathsf{S}(d)$ as:
        $s \leftarrow \mathsf{Sign}(sk, d)$
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{d\}$
        return $s$
    $(d, s) \leftarrow \mathsf{A}^{\mathsf{S}}(pk)$
    return $[\![\mathsf{Verify}(pk, d, s) \ \wedge \ d \notin \mathcal{S}]\!]$
```

Game 7: SUF-CMA

```
define $\mathsf{Game}^{\mathsf{A}}_{\mathsf{SUF\text{-}CMA}}(1^\lambda)$ as:
    $(sk, pk) \leftarrow \mathsf{KeyGen}(1^\lambda)$
    $\mathcal{S} \leftarrow \varnothing$
    define $\mathsf{S}(d)$ as:
        $s \leftarrow \mathsf{Sign}(sk, d)$
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(d, s)\}$
        return $s$
    $(d, s) \leftarrow \mathsf{A}^{\mathsf{S}}(pk)$
    return $[\![\mathsf{Verify}(pk, d, s) \ \wedge \ (d, s) \notin \mathcal{S}]\!]$
```

A signature scheme Σ is secure in the EUF-CMA model if for all polynomial-time quantum adversaries A, their winning probability in the EUF-CMA game is negligible. The definition is analogous with respect to the SUF-CMA model.

$$\mathsf{InSec}_{\Sigma}^{\mathsf{EUF\text{-}CMA}}(\mathsf{A}) \triangleq \Pr[\mathsf{Game}_{\mathsf{EUF\text{-}CMA}}^{\mathsf{A}}(1^{\lambda}) \Rightarrow 1] \leq \epsilon(\lambda)$$
$$\mathsf{InSec}_{\Sigma}^{\mathsf{SUF\text{-}CMA}}(\mathsf{A}) \triangleq \Pr[\mathsf{Game}_{\mathsf{SUF\text{-}CMA}}^{\mathsf{A}}(1^{\lambda}) \Rightarrow 1] \leq \epsilon(\lambda)$$

A winning adversary for EUF-CMA is also a winning adversary for SUF-CMA, but not necessarily the other way around, so SUF-CMA is the stronger notion. However, it is not clear whether this distinction can lead to a meaningful attack, and for many purposes EUF-CMA is sufficient.

B Proof of Theorem 1

Proof. The inequality follows from a sequence of games. Each term in the inequality arises from one game hop.

- $\mathsf{Game0}$ is the SUF-CMA game. By definition we have
$$\mathsf{InSec}_{\Sigma}^{\mathsf{SUF-CMA}}(\mathsf{A}) \triangleq \Pr[\mathsf{Game0}^{\mathsf{A}}(1^{\lambda}) \Rightarrow 1] \ . \tag{30}$$
- $\mathsf{Game1}$ is a hybrid between the SUF-CMA and EUF-CMA games where the list $\mathcal{S}$ drops $\mathbf{r}$ and in particular consists of pairs $(d,(Y,U))$, where (Y,U) was drawn from the signature $s=(Y,U,\mathbf{r})$. An adversary wins $\mathsf{Game0}$ but not $\mathsf{Game1}$ if he can find a second signature $s_2=(Y,U,\mathbf{r}_2)$ for a message d that has already been signed with $s_1=(Y,U,\mathbf{r}_1)$, both of which are valid and such that $\mathbf{r}_1 \neq \mathbf{r}_2$. Finding such an $\mathbf{r}$ amounts to an SSSNE instance with $m'=m$, $n'=n$, and length bound $\beta = 2\sqrt{n}\cdot 2^u$. So
$$|\Pr[\mathsf{Game0}^{\mathsf{A}}(1^{\lambda}) \Rightarrow 1] - \Pr[\mathsf{Game1}^{\mathsf{A}}(1^{\lambda}) \Rightarrow 1]| \leq \mathsf{Succ}_{m,n,q,2\sqrt{n}\cdot 2^u}^{\mathsf{SSSNE}} \ . \tag{31}$$
- $\mathsf{Game2}$ is the EUF-CMA game for Σ. The adversary wins $\mathsf{Game1}$ but not $\mathsf{Game2}$ when he produces two signatures $s_1=(Y_1,U_1,\mathbf{r}_1)$, $s_2=(Y_2,U_2,\mathbf{r}_2)$ for the same message d. Given such an adversary A, it is possible to build another adversary B^{A} that wins the EUF-CMA game for Σ with the same probability. The simulator B^{A} maintains a dictionary[2] $\mathcal{Q}$ of random oracle query-response pairs and presents A with the following view of the random oracle, in terms of his own random oracle H'.

```
define H(q) as:
|   if $q \notin \mathcal{Q}$.keys
|   |   try parse $q$ as $q = X\|Y\|U\|d$
|   |   if parse success then:
|   |   |   $\mathcal{Q}[q] \leftarrow \mathsf{H}'(X\|Y\|U\|Y\|U\|d)$
|   |   else:
|   |   |   $\mathcal{Q}[q] \leftarrow \mathsf{H}'(q)$
|   return $\mathcal{Q}(q)$
```

[2] We conceive of dictionaries in the sense of the python programming language as mapping keys to values. For a dictionary $\mathcal{D}$, we write $\mathcal{D}$.keys to refer to the list of keys, and $\mathcal{D}[k]$ to refer to the value indicated by the key k.

Additionally, B modifies A's view of the signature oracle accordingly. Specifically, A's view S is given in terms of B's signature oracle S' as follows.

```
1. define $\mathsf{S}(d)$ as:
2.     $i \leftarrow 0$
3.     repeat:
4.         $(Y, U, \mathbf{r}) \leftarrow \mathsf{S}'(d\|i)$
5.         $i \leftarrow i + 1$
6.     until $X\|Y\|U\|d \notin \mathcal{Q}.\mathsf{keys}$
7.     $\mathcal{Q}[X\|Y\|U\|d] \leftarrow \mathsf{H}'(X\|Y\|U\|d\|i)$
8.     return $(Y, U, \mathbf{r})$
```

When A queries the signature oracle he will obtain a signature that is valid with respect to $\mathsf{H}'(X\|Y\|U\|d\|i) = \mathsf{H}(X\|Y\|U\|d)$, for some i. However, if he produces a signature without querying the signature oracle, it must be valid with respect to $\mathsf{H}(X\|Y\|U\|d) = \mathsf{H}'(X\|Y\|U\|Y\|U\|d)$. In the latter case B has obtained a valid signature on a different message, namely $Y\|U\|d$. So whenever A wins $\mathsf{Game1}$, then B^{A} wins $\mathsf{Game2}$, and

$$\Pr[\mathsf{Game1}^{\mathsf{A}}(1^{\lambda}) \Rightarrow 1] = \Pr[\mathsf{Game2}^{\mathsf{B}^{\mathsf{A}}}(1^{\lambda}) \Rightarrow 1] \ . \tag{32}$$

The simulator B does incur a time penalty compared to the simulated adversary A, owing to the loop of lines 3–5 in $\mathsf{S}(d)$. The number of iterations of this loop is bounded by $\#\mathcal{Q}.\mathsf{keys}$, which in turn is bounded by Q_{H}. So if A runs in polynomial time, then so does B.

- $\mathsf{Game3}$ is the same EUF-CMA game but with respect to Σ', a modification of the scheme that drops derandomization. In particular, Σ' is identical to Σ except for line 3 of $\Sigma'.\mathsf{Sign}$, which is:

  ```
  3.     $\{coins_{3,j}\}_{j=0}^{m-1} \xleftarrow{\$} \{0,1\}^{\star}$     ,
  ```

  where $\star$ is a stand-in for a large enough integer. We build a simulator B^{A} that wins $\mathsf{Game3}$ with probability related to A's winning probabilty in $\mathsf{Game2}$. The level of indirection is necessary because B must maintain a list of signatures $\mathcal{S}$, which is initialized to $\varnothing$ and is updated with every signature query. In particular, B presents the following view of the signature oracle S to the adversary A, in terms of his own view S'.

  ```
  define $\mathsf{S}(d)$ as:
      if $d \notin \mathcal{S}.\mathsf{keys}$ then:
          $\mathcal{S}[d] \leftarrow \mathsf{S}'(d)$
      return $\mathcal{S}[d]$
  ```

  If there is an adversary A such that $\Pr[\mathsf{Game2}^{\mathsf{A}}(1^{\lambda}) \Rightarrow 1] \neq \Pr[\mathsf{Game3}^{\mathsf{B}^{\mathsf{A}}}(1^{\lambda}) \Rightarrow 1]$, then it is possible to use A in the construction of a distinguisher D for the PRG, so

$$\left|\Pr[\mathsf{Game2}^{\mathsf{A}}(1^{\lambda}) \Rightarrow 1] - \Pr[\mathsf{Game3}^{\mathsf{B}^{\mathsf{A}}}(1^{\lambda}) \Rightarrow 1]\right| \leq \mathsf{Adv}_{\mathsf{G}}^{\mathsf{PRG}}(\mathsf{D}^{\mathsf{A}}) \tag{33}$$

$$\leq \mathsf{Adv}_{\mathsf{G}}^{\mathsf{PRG}} \ . \tag{34}$$

- $\mathsf{Game4}$ is a Key-Only Attack (KOA) with respect to Σ'. In particular, this game is identical to $\mathsf{Game3}$ except that the adversary has no access to the signing oracle. Given an adversary A that wins $\mathsf{Game3}$, it is possible to build an adversary B that wins $\mathsf{Game4}$ with almost as good probability.
  In particular, B simulates A and whenever A queries the signature oracle for a signature on a message d, B responds by running the HVZK simulator of $\mathit{\Pi}$ to produce a transcript $(Y, U, a, \mathbf{r})$, and he repeats the simulation at most κ times until $\mathsf{leaks}(Y, U, a, \mathbf{r}) = 0$. He then reprograms the random oracle to respond with a when queried on $X\|Y\|U\|d$. Moreover, the simulator B maintains a list of queries made to the random oracle and samples a new transcript whenever $X\|Y\|U\|d$ was already queried.
  The event where the simulator fails to answer a signature query occurs exactly when all κ tests $\mathsf{leaks}(Y, U, a, \mathbf{r})$ fail. For $\lambda > 125$, κ was chosen to make this probability at most $2^{-\lambda}$. The view of A is identical across both worlds except for the simulated or authentic transcripts. So any A that gives rise to a distinguisher between games 3 and 4, conditioned on all signature queries being successful, can be turned into a distinguisher D^{A} between authentic and simulated transcripts:

$$|\Pr[\mathsf{Game3}^{\mathsf{A}}(1^\lambda) \Rightarrow 1] - \Pr[\mathsf{Game4}^{\mathsf{B}^{\mathsf{A}}}(1^\lambda) \Rightarrow 1]| \leq \mathsf{Adv}^{\mathsf{ZK}}_{\mathit{\Pi}}(\mathsf{D}^{\mathsf{A}}) + 2^{-\lambda} \tag{35}$$

$$\leq \mathsf{Adv}^{\mathsf{DSSNE}}_{3m+n,2n,q,\sqrt{2n}\cdot 2^u}(\mathsf{D}^{\mathsf{A}}) + 2^{-\lambda} \ . \tag{36}$$

- $\mathsf{Game5}$ is the impersonation game for the identification scheme composed of the proof system $\mathit{\Pi}$ adjoined with key generation algorithm $\Sigma.\mathsf{KeyGen}$. The adversary wins if he can convince the verifier $\mathit{\Pi}.\mathsf{V}$ in the interactive zero-knowledge proof $\mathit{\Pi}$. Specifically, the game is defined as follows.

```
define $\mathsf{Game5}^{\mathsf{A}}(1^\lambda)$ as:
|   $sk, pk \leftarrow \Sigma.\mathsf{KeyGen}(1^\lambda)$
|   return $\mathsf{out}_{\mathit{\Pi}.\mathsf{V}}(\mathsf{A}(pk) \leftrightarrow \mathit{\Pi}.\mathsf{V}(pk))$
```

  Given an adversary A for $\mathsf{Game4}$, build an adversary B for $\mathsf{Game5}$ as follows. Choose a random query index $i \xleftarrow{\$} \{1, \ldots, Q_{\mathsf{H}}\}$ and present A with the following view of the random oracle, which uses a state variable j initialized at $j = 0$.

```
define $\mathsf{H}(q)$ as:
|   $j \leftarrow j + 1$
|   if $j = i$ then:
|   |   try parse $q$ as $q = X\|Y\|U\|d$
|   |   if parse success then:
|   |   |   send $(U, V)$ to $\mathit{\Pi}.\mathsf{V}$
|   |   |   receive $a$ from $\mathit{\Pi}.\mathsf{V}$
|   |   |   $\mathcal{Q}[q] \leftarrow a$
|   if $q \notin \mathcal{Q}.\mathsf{keys}$
|   |   $\mathcal{Q}[q] \xleftarrow{\$} \mathbb{Z}_{2^e}$
|   return $\mathcal{Q}[q]$
```

7.2 Key Encapsulation from Noisy Key Agreement in the Quantum Random Oracle Model

Publication data

Alan Szepieniec and Reza Reyhanitabar and Bart Preneel. “Key Encapsulation from Noisy Key Agreement in the Quantum Random Oracle Model” *to be submitted to* CRYPTO *2019*.

Contributions

Principal author

Notes

The germ for this paper came from developing the Ramstake submission. In particular, the transformation from shared noisy one-time pad (“snow-tipi”) to IND-CCA secure KEM seemed rather independent of the underlying mathematical structure and I was wondering if it could be proved independently. This train of thought led to the consideration of noisy key agreement (NKA) protocols as a standalone primitive. Additionally, this paper introduces and uses new tools for proofs in the quantum random oracle model, some of which were summarized already in Part I, Ch. 3.

Key Encapsulation from Noisy Key Agreement in the Quantum Random Oracle Model

Alan Szepieniec[1], Reza Reyhanitabar[2], and Bart Preneel[1]

[1] imec-COSIC KU Leuven, Belgium
alan.szepieniec@esat.kuleuven.be, bart.preneel@esat.kuleuven.be
[2] Elektrobit Automotive GmbH, Germany
reza.reyhanitabar@elektrobit.com

Abstract. A multitude of post-quantum key encapsulation mechanisms (KEMs) and public key encryption (PKE) schemes *implicitly* rely on a protocol by which Alice and Bob exchange public messages and converge on secret values that are identical *up to some small noise*. By our count, 24 out of 49 KEM or PKE submissions to the NIST Post-Quantum Cryptography Standardization project follow this strategy. Yet the notion of a *noisy key agreement* (NKA) protocol lacks a formal definition as a primitive in its own right. We provide such a formalization by defining the syntax and security for an NKA protocol. This formalization brings out four generic problems, called A and B State Recovery, Noisy Key Search, and Noisy Key Distinguishing (NKD), whose solutions must be hard in the quantum computing model. Informally speaking, these can be viewed as noisy, quantum-resistant counterparts of the problems arising from the classical Diffie-Hellman type protocols. We show that many existing proposals contain an NKA component that fits our formalization and we reveal the induced concrete hardness assumptions. The question arises whether considering NKA as an independent primitive can help provide modular designs with improved efficiency and/or proofs. As the second contribution of this paper, we answer this question positively by presenting a generic transform from a secure NKA protocol to an IND-CCA secure KEM in the quantum random oracle model, with a security bound related to the insecurity of the NKD problem. This transformation is essentially the same as that of the NIST candidate Ramstake. While establishing the security of Ramstake was our initial objective, the collection of tools that came about as a result of this journey is of independent interest.

Keywords: Post-quantum, key encapsulation, public key encryption, quantum random oracle model, noisy key agreement.

1 Introduction

Post-Quantum Cryptography. Most of the standard public key cryptosystems in use, including Diffie-Hellman and derivatives thereof, RSA, DSA, ECDSA, and ElGamal cryptosystems, rely on the computational hardness of number theoretic problems. For these problems, in particular factoring and discrete log (DLOG) problems, quantum computers offer exponential speedups compared to classical computers. Shor's factoring and discrete logarithm algorithms [57] render these cryptosystems insecure in the quantum computing era.

The anticipation of this threat is what drives the development and deployment of *post-quantum cryptography*—cryptographic algorithms that despite running on classical computers promise to resist quantum attacks—well before large-scale quantum computers arrive.

In contrast to the aforementioned public key schemes, symmetric key algorithms such as AES and its various modes of operations, as well as hash functions such as SHA2 and SHA3 remain relatively unaffected by quantum computers. The best known quantum attack on these primitives is Grover's generic search algorithm [33] and it offers only a square root speed-up, meaning that the same security level is attained against quantum computers by merely doubling the key or output length. In this line, NIST has initiated a competition for post-quantum cryptography standardization [49]. Out of 69 complete and proper submissions, 22 proposals achieve signature scheme functionality and 49 achieve key encapsulation mechanisms (KEMs) or public key encryption (PKE) or both (with some overlap) [50].

Key Exchange (KE). KE protocols enable two parties who communicate over an adversarially-controlled channel to obtain a secret session key. Starting with the seminal work of Diffie and Hellman [28], there is now a rich body of work on this topic in the literature containing several security models and design paradigms [12,20,41,42,24]. By convention, we consider Key Agreement (KA) protocols as a subset of KE protocols in which both parties influence the generation of the resulting session key; for instance, Diffie-Hellman (DH) type protocols are classic examples of KA.

Key Encapsulation Mechanism. Cramer and Shoup [22,23] provided, among other contributions, a formal treatment of hybrid Public Key Encryption (PKE) secure against adaptive chosen ciphertext attacks (CCA) [56]. The approach, known as the KEM/DEM (Key Encapsulation Mechanism/Data Encapsulation Mechanism) framework, rigorously captures the folklore method for building a hybrid encryption scheme, namely by using public key cryptography to *encapsulate* a symmetric *session key*, followed by symmetric-key encryption.

While the original and main application of KEM has been in hybrid PKE, it has turned out that pure KEM can be a useful cryptographic tool in its own right in other applications; for example, to build schemes for identification [9] and authenticated key exchange [19,31,66].

Design Strategies. We identify three binary design choices that partition the design space of KEMs and PKEs. They are *noisy versus noise-free*, *convergence versus inversion*, and *reconciliation versus transmission.* The last choice only makes sense in the case of noisy convergence.

Noisy versus noise-free considers the nature of the underlying mathematical hard problems. Multivariate quadratic (MQ) equations and supersingular isogenies (SI) achieve computational hardness without adding random noise, whereas lattice- and code-based problems are computationally difficult precisely because they rely on the addition of noise. The newest member of the latter class is the family of problems based on sparse integers and arithmetic modulo (pseudo-) Mersenne primes [1,50].

Convergence versus inversion looks at the strategy to achieve the targeted KEM or PKE functionality. The earlier MQ, code- and lattice-based cryptosystems relied on trapdoor inversion [45,46,51,34], in which the public operation amounts to evaluating a trapdoor function and the secret operation amounts to inverting it. In contrast, newer proposals implicitly rely on a noisy key agreement protocol in which two parties obtain *roughly* the same key which is hard for the passive eavesdropper to approximate [29,7,16,26]. The exception to this rule is the supersingular isogeny Diffie-Hellman (SIDH) cryptosystem [38], and its brother CSIDH [21], both of which converge on identical keys and hence might be termed an *exact key agreement* (EKA) protocol but nevertheless amounts to a special case of NKA. To date, SIDH and CSIDH are the only post-quantum cryptosystem capable of achieving *static key agreement* (SKA) functionality, whereby any pair of participants who know each other's public key can derive the same shared symmetric key *without interaction*, opening up the possibility for bypassing the exchange of public key messages and instead communicating over the symmetric channel immediately.

Reconciliation versus transmission deals with the details of obtaining identical keys after similar keys were obtained through a noisy convergence strategy. Reconciliation entails sending helper data to enable the receiver to correct the errors or otherwise extract an identical template from the noisy views of the shared key. There are many subtle variants, all of which rely on the specific mechanics of the underlying mathematics [29,54,16,62]. In contrast, transmission[3] uses the shared noisy key to mask a new message entirely; this new message must then contain enough redundancy to be decodable after being masked and unmasked with two approximately equal one-time pads. Transmission is arguably less prone to error, but does come with a bandwidth penalty [6,40].

Our Contribution. This paper presents two main contributions. The first is a formal syntax and security definition to capture the notion of a noisy key agreement (NKA) protocol as a new useful primitive. The second is a generic transformation to turn an NKA protocol into an IND-CCA secure KEM in the quantum random oracle model. Based on the previous categorization of design strategies, our transformation applies to noisy convergence based protocols, and uses the transmission strategy.

The syntax of NKA protocols captures the intuition where, after an initialization phase that generates public parameters, Alice and Bob generate a state

[3] Also called the *encryption-based approach* in NewHopeSimple [6], and an *asymmetric key consensus* in the context of OKCN/AKCN [40].

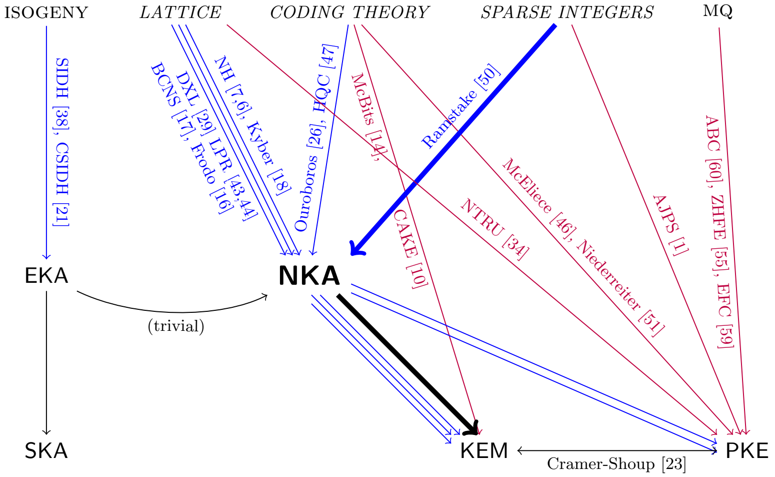


Fig. 1: Map of post-quantum KEM and PKE. The bold objects indicate the contributions of this paper. Italics denotes noisy mathematics; blue arrows denote convergence, red ones denote inversion, and black ones represent generic transforms.

and contribution pair. They then exchange their protocol contributions and use their own state and the other party's contribution to converge on approximately the same value. An explicit treatment of protocol failure events resulting from excessive noise, which may cause decryption or decapsulation errors, is built in to our formalism.

This syntax naturally lends to four attack vectors, which we formulate as generic problems called *A State Recovery (ASR)*, *B State Recovery (BSR)*, *Noisy Key Search (NKS)*, and *Noisy Key Distinguishing (NKD)*, mirroring the DLOG, computational Diffie-Hellman (CDH) and decisional Diffie-Hellman (DDH) Problems in Diffie-Hellman protocols. While the classical DLOG, CDH and DDH problems are efficiently solvable by quantum algorithms, these new generic problems arising from formalization of *noisy* key agreement must remain hard in the quantum computing model. Hence, instantiations of NKA must rely on concrete hardness assumptions that guarantee infeasibility of these generic problems even in the face of quantum solvers. Many existing proposals contain an NKA component that fits our formalization; we identify the induced concrete hardness assumptions.

Security of an NKA protocol is defined with respect to the NKD problem. Specifically, an NKA protocol is secure if and only if its NKD problem is hard on average. We justify this definition in several ways.

- The hardness of NKD implies the hardness of NKS, ASR and BSR; therefore the NKD Assumption is the strongest assumption.

- It is analogous to regular Diffie-Hellman, where the protocol is secure if and only if the DDH problem is hard (assuming authenticated links).
- We consider an example from the NIST PQC project that fits the NKA framework and where ASR and BSR are hard, but where NKD is easy, which led to the submission's prompt cryptanalysis.
- We consider in the appendix an alternate definition of security based on a suitable adaptation of the well-known Canetti-Krawczyk session-key security (SK-security) notion [20]. We find that this security notion is equivalent to the average-case hardness of the NKD problem.

These results indicate that the average-case hardness of the NKD problem is essential in the context of secure NKA-based KEMs and PKEs.

As our second and main contribution, we provide a generic NKA-to-KEM transformation for *noisy*, *convergence*-based protocols, applying the *transmission* strategy, and featuring an IND-CCA security proof in the quantum random oracle model. The main feature in this context is its *genericity*: it applies regardless of the mathematics of the underlying NKA protocol and as such enables a modular design workflow. We note that the Ramstake submission [50] uses essentially the same transformation but was presented without proof; this paper therefore proves the security of Ramstake, assuming the appropriate NKD problem is hard on average.

In comparison to other IND-CCA transforms in the literature, the most obvious difference is that the starting point of our transform is an NKA protocol, whereas other IND-CCA transforms start from an IND-CPA secure PKE or KEM. We include the key-confirmation hash of Targhi-Unruh in the ciphertext [61] and follow the *derandomization and re-encryption* approach so named by Hofheinz *et al.* [35]. We note that a recent result by Jiang *et al.* [39] suggests that this additional hash might not be necessary, but we leave open for the time being the question whether dropping it affects the security of our particular construction. In contrast to these related results [61,35,39], our session key is computed from *bipartite contribution*, *i.e.*, as a function of both the public key and the encapsulator's randomness; this property prevents Bob from establishing the same symmetric key for separate channels, one with Alice and one with Charlie.

An outstanding feature of our proof is the tighter security bound: the insecurity of the underlying primitive (NKD of the NKA protocol in our case; IND-CPA security of the PKE or KEM elsewhere) undergoes a square-root degradation, similar to the result by Jiang *et al.* and in stark contrast to the quartic root degradation of Targhi-Unruh and Hofheinz *et al.*. This improved bound is the result of treating the extendable output function that is used for derandomization as a random oracle; this enables an argument about the queries that are made to it. While our bound does feature fourth-roots, they apply only to the hash function insecurity.

Central to our security proof is a new technique for lifting classically-valid random oracle model security proofs to the quantum random oracle model. We introduce, define, and use, the *aggregate quantum query amplitude*, which behaves similar to the expected number of times a particular query was made by an adversary throughout the entire computation. We use this notion as a starting point to derive lemmata that enable refined argumentation about adversarial query behavior, as well as to derive a multi-target generalization of Unruh's One-Way to Hiding Lemma [64]. These lemmata are used in the security proof to capture the intuition that a quantum adversary does not know the random oracle's output on inputs that were not queried. We believe this notion and our proof technique to be of independent interest as a useful tool in security analysis of other PQC schemes.

Ramstake and the NIST PQC Project. While our starting point was the establishment of a security proof for Ramstake, this journey has led to many independently useful tools for the analysis and provable security of post-quantum cryptosystems. Nevertheless, we stress that despite the detour we were successful in this endeavor. The main contribution of this paper remains the establishment of a security proof reducing the IND-CCA security of Ramstake to solving the appropriate version of the NKD problem — called the Low Hamming Diffie-Hellman Decision (LHDHD) Problem in the context of Ramstake [50].

The ongoing NIST PQC project, as a design-focused project with a somewhat fixed timeframe, has boosted research on PQC and has attracted 69 proposals, which are the subject of intense scrutiny. Nevertheless —or perhaps accordingly— it is compelling and timely to revisit the foundations of security notions and of design paradigms for next-generation PQC schemes in order to stay ahead of emerging threats and to prevent past failures from being transmuted in future. This paper aims to be a step forward in this direction.

Organization of the Paper. Section 2 provides notations, conventions and definitions used throughout the paper. In Sect. 3 we present our *noisy key agreement* formalism, including syntax, abstract hard problems, and security definition. Section 4 presents our NKA-to-KEM transformation, and we follow up in Sect. 5 with a discussion on proof techniques (including the aggregate quantum query amplitude) before presenting the security proof. Section 6 concludes the paper.

2 Preliminaries

Notation and conventions. We use $a \leftarrow b$ to denote the assignment of the value b to the variable a, and $a \xleftarrow{\$} A$ to denote the assignment of a uniformly random element from the set A. Algorithms are denoted in sans-serif font and the event that an algorithm A, on input x, outputs y is written as $\mathsf{A}(x) \Rightarrow y$ and $\mathsf{A}(x) \not\Rightarrow y$ when it does not output y. A long double right arrow ($\Longrightarrow$) denotes logical implication, and $\triangleq$ denotes equality by definition. Superscript, *e.g.*, A^{O} denotes an algorithm A having oracle access to O, meaning that A can query O and receive responses in a black box manner but he cannot study the oracle's code or composition.

A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}_{>0}$ is *negligible* if for all polynomials $p(x) \in \mathbb{R}[x]$ there is an $N \in \mathbb{N}$ such that for all $x > N$, $\mathsf{negl}(x)$ drops faster than the reciprocal of $|p(x)|$. Formally, we need only consider the dominant monomial of $p(x)$:

$$\forall c > 1 \,.\, \exists N \in \mathbb{N} \,.\, \forall \lambda > N \,.\, \mathsf{negl}(\lambda) \leq \frac{1}{\lambda^c} \enspace .$$

Quantum Computation. The state of a quantum system of k qubits is given by a unit-length vector in *ket* notation, *e.g.* $|\Psi\rangle \in \mathcal{H}$, where $\mathcal{H} \subset \mathbb{C}^{2^k}$; where $\langle\Psi|$ is its complex conjugate transpose, and $\langle\Psi|\Phi\rangle$ is the standard inner product. The composition of two quantum systems is described by the tensor product $|\Psi\rangle \otimes |\Phi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$, which is the vector of all multilinear products. However, sometimes quantum systems of more than one qubit cannot be factored into the tensor product of independent systems; in this case the two systems are *entangled.* Except for measurements, all quantum computations are unitary transforms on the state space. Measurement of a system $|\Psi\rangle$ is defined with respect to a set of orthonormal basis vectors $|b_0\rangle, |b_1\rangle, \ldots, |b_{2^k-1}\rangle$ and affects the system by collapsing it to $|b_i\rangle$ with probability $\langle b_i|\Psi\rangle\langle\Psi|b_i\rangle$. Any bitstring $s \in \{0,1\}^k$ has an associated basis vector $|s\rangle = |b_i\rangle$ for some i. Whenever a state is a non-trivial sum of basis vectors, *i.e.*, with weights different from 0, -1 and 1, it represents a *superposition* of values. Except for measurement, all quantum operations are reversible. Moreover, it is possible to transform any quantum circuit into an equivalent circuit where all the measurement operators are located at the end.

An equivalent characterization of quantum computation is in terms of a system's density operator or density matrix $\rho \in \mathbb{C}^{2^k \times 2^k}$, as opposed to its state vector $|\Psi\rangle \in \mathcal{H} \subset \mathbb{C}^{2^k}$. The density operator associated with a pure state $|\Psi\rangle$ is $\rho = |\Psi\rangle\langle\Psi|$. When the density operator has a higher rank it represents a probability ensemble: the density matrix $\rho = \sum p_i |\psi_i\rangle\langle\psi_i|$ represents a system that has a probability p_i of having state $|\psi_i\rangle$. The density operator is especially useful for its characterization of parts of a complex quantum system because this operator, together with the partial trace operator, leads to the correct determination of observable statistics. The *reduced density operator* ρ_A of a subsystem A of a composite system $A + B$ with density matrix $\rho_{A,B}$ is obtained by "tracing out" the Hilbert space $\mathcal{H}_B$ associated with B, *i.e.*, by applying the *partial trace* operator $\rho_A = \mathsf{Tr}_B \rho_{A,B}$ which is defined by $\forall |a\rangle \in \mathcal{H}_A, |b\rangle \in \mathcal{H}_B \,.\, \mathsf{Tr}_B(|a\rangle \otimes |b\rangle\langle a| \otimes \langle b|) = |a\rangle\langle a|\langle b|b\rangle$. For more details on quantum computation and quantum information we refer the reader to a comprehensive treatment of the subject by Nielsen and Chuang [52].

We use capital letters without ket notation to denote quantum registers, *i.e.*, the sets of qubits assigned to a variable. We use lowercase letters in ket notation to denote computational basis vectors with unspecified index, and Greek letters in ket notation to denote non-trivial superpositions of computational basis states.

Quantum Random Oracle Model. Our security proof relies on the modeling of hash functions as *random oracles* [30,13], which are uniformly random functions $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$ with a fixed output length, typically equal to the security parameter. If necessary, the random oracle's output space can be lifted to any finite set X. We use subscripts to differentiate the random oracles associated with different output spaces. The adversary has no access to the function's full description or source code. Security proofs of this type are said to hold in the *random oracle model* (ROM).

Boneh *et al.* show that the random oracle model is not a suitable model when attacks on quantum computers are to be considered [15]. Instead, adversaries have access to a black box that operates on a query-response register pair (Q, R) by sending $|q, r\rangle \mapsto |q, r \oplus \mathsf{H}(q)\rangle$. In this model, quantum adversaries are capable of querying the random oracle on superpositions of bit strings and should receive a superposition answer back. Many classically-valid random-oracle constructions fail to account for this capability and rely in their security proofs on notions or behaviors which become ill-defined when quantum access is considered, such as the list of queries or lazy sampling. As a result, the security proof is valid in the classical random oracle model but invalid in the *quantum random oracle model* (QROM). Many subsequent works elaborate on the notion either by lifting constructions or proofs to the QROM [58,63,65,61], or by showing that such a lift is impossible [25,8].

Derandomization. Our construction relies on derandomization. While pseudorandom generators are usually sufficient for this task, in our case the adversary has quantum oracle access to the function. We thus opt for an extendable-output function (XOF) [48], which we model as a random oracle in the security proof.

In derandomization, probabilistic polynomial-time algorithms are made deterministic. In particular, let A be a probabilistic polynomial-time algorithm and $s \in \{0,1\}^\lambda$ a seed. We write $\mathsf{A}(x)$ to denote that A is run on input $x \in \{0,1\}^*$, and $\mathsf{A}(x; r)$ to make the contents of its random tape $r \in \{0,1\}^R$ explicit. Then A is derandomized by invoking $\mathsf{A}(x; \mathsf{H}_3(s, R))$ for some s. In fact, in our construction we make abstraction of the output length R and instead use denote by H_3 the function that takes a short input and outputs "enough" random bits.

Key Encapsulation Mechanism. A Key Encapsulation Mechanism (KEM) $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ is a triple of probabilistic polynomial-time algorithms, where

- KeyGen takes a security parameter λ (in unary representation) and outputs two objects: a secret key sk and a public key pk;
- Encaps takes a public key pk and outputs two objects: a symmetric key k from a symmetric key space $\mathsf{SKSpace}$ and a ciphertext c;
- Decaps takes a secret key sk and a ciphertext c and outputs a session key k from the symmetric key space $\mathsf{SKSpace}$, or returns $\perp$ if a failure has occurred.

A KEM's *failure probability* ϵ is defined as

$$\epsilon = \Pr\left[k_e \neq k_d \,\middle|\, \begin{array}{l} sk, pk \leftarrow \mathsf{KeyGen}(1^\lambda) \\ k_e, c \leftarrow \mathsf{Enc}(pk) \\ k_d \leftarrow \mathsf{Dec}(sk, c) \end{array} \right] \quad , \tag{1}$$

and should be small or else the scheme is not usable.

Security of KEMs is defined using the following IND-CCA[4] game, defined with respect to an adversary $\mathsf{A}^{\mathsf{D}(\cdot)}$ who who has black box access to a decapsulation oracle. The IND-CPA game relaxes this notion by disallowing decapsulation queries, but is otherwise identical.

Game 2: IND-CCA

```
$sk, pk \leftarrow \mathsf{KeyGen}(1^\kappa)$
$b \xleftarrow{\$} \{0,1\}$
$k_0 \xleftarrow{\$} \mathsf{SKSpace}$
$c, k_1 \leftarrow \mathsf{Encaps}(pk)$
$\mathcal{S} \leftarrow \varnothing$
define $\mathsf{D}(q)$ as:
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{q\}$
    return $\mathsf{Decaps}(sk, q)$
$b' \leftarrow \mathsf{A}^{\mathsf{D}(\cdot)}(pk, k_b, c)$
return $[\![b = b' \wedge c \notin \mathcal{S}]\!]$
```

Game 3: IND-CPA

```
$sk, pk \leftarrow \mathsf{KeyGen}(1^\kappa)$
$b \xleftarrow{\$} \{0,1\}$
$k_0 \xleftarrow{\$} \mathsf{SKSpace}$
$c, k_1 \leftarrow \mathsf{Encaps}(pk)$
$b' \leftarrow \mathsf{A}(pk, k_b, c)$
return $[\![b = b']\!]$
```

The Iverson brackets $[\![\cdot]\!]$ evaluate to 1 if the logical expression is true and to 0 otherwise. A KEM is *secure* if for all polynomial-time quantum adversaries $\mathsf{A}^{\mathsf{D}(\cdot)}$ with *classical* black box query access to a decapsulation oracle D, their advantage $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A}^{\mathsf{D}(\cdot)})$ is negligible:

$$\mathsf{Adv}_{\mathcal{E}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A}^{\mathsf{D}(\cdot)}) \triangleq \left| \Pr\left[\mathsf{Game}_{\mathsf{IND\text{-}CCA}}^{\mathsf{A}^{\mathsf{D}(\cdot)}}(1^\lambda) \Rightarrow 1\right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda) \ . \qquad (2)$$

Most proposals for post-quantum KEMs claim only to satisfy the strictly weaker indistinguishability under chosen plaintext attack (IND-CPA) security notion and emphasize targeting the exchange of *ephemeral keys* only, being a scenario in which chosen ciphertext attacks are unrealistic. Nevertheless, there are several notable exceptions that do meet the stronger IND-CCA requirement [10,18,4]. Moreover, there are generic conversions from IND-CPA secure KEMs and PKEs to IND-CCA secure ones in the classical and quantum random oracle models [32,27,61,36,35].

Error-Correcting Codes. A linear $[n, k, d]$-code $\mathcal{C}$ is a subspace $\mathbb{F}_q^n$ of dimension k. We consider here only bitstrings in which case the *symbol field* $\mathbb{F}_q = \mathbb{F}_2$ and codewords are elements of $\mathbb{F}_2^n \cong \{0,1\}^n$ but encode elements of $\mathbb{F}_2^k \cong \{0,1\}^k$ with $k < n$. The *minimum distance* d of a code is the Hamming weight of its smallest nonzero codeword: $d = \mathsf{min}_{\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}} \textsc{hw}(\mathbf{c})$. The code is capable of finding the nearest codeword $\mathbf{c} \in \mathcal{C}$ to a noisy word $\mathbf{c}'$ as long as the Hamming weight of the distance is at most d: $\textsc{hw}(\mathbf{c}' - \mathbf{c}) \leq d/2$. This process is called *error correction*. This paper abstractly assumes the availability of two functionalities:

[4] The pseudocode of Game 2 follows the IND-CCA-OP notion of Bellare, Hofheinz and Kiltz [11], who prove equivalence between this and five other common IND-CCA notions for KEMs.

- $\mathcal{C}$.encode : $\mathbb{F}_2^k \to \mathbb{F}_2^n$, which generates codewords from messages;
- $\mathcal{C}$.decode : $\mathbb{F}_2^n \to \mathbb{F}_2^k$, which corrects the errors and returns the associated message, or returns $\perp$ if there are more than some threshold t of errors.

3 Noisy Key Agreement

The common theme in all constructions relying on what we call noisy key agreement is the distinction between "small" and "large" elements in compatible spaces. Before the protocol starts, Alice and Bob agree publicly on a random large element G. When the protocol starts, both generate small secrets a, b and c, d respectively. They then exchange messages $aG + b$ and $cG + d$, and obtain views $acG + ad$ and $acG + cb$ of a noisy shared secret which differ only by a still-small term $ad - cb$.

To the best of our knowledge, the first use of the term "Noisy Diffie-Hellman" traces back to a pair of presentations given by Gaborit in 2010 [2,3], although the underlying strategy was already folklore knowledge by that point[5]. We prefer to reserve the term Diffie-Hellman for noise-free key agreement protocols involving square-and-multiply or double-and-add procedures to compute commutative actions on group elements.

The purpose of this section is to abstract out the mathematics and find a syntax that contains all instances of this principle. We call the resulting formalism *noisy key agreement* (NKA). Its desirable properties are: (i) NKA should contain standard Diffie-Hellman-based key agreement protocols for noise level zero. (ii) NKA should come with a usable security definition. (iii) NKA should be identifiable inside the constructions that are supposedly based on it.

3.1 Syntax

We formalize the above intuition as follows. Before the protocol starts, Alice and Bob must agree on a set of *instance parameters iparams*, which is the output of the *initialization function* Init when run on the security parameter λ (provided in unary notation). Alice's and Bob's tasks during the protocol are divided into two algorithms each. In the *contribute algorithms* AContr and BContr, they each generate a *state*, *A_state* and *B_state*, in addition to *contributions A_contr* and *B_contr*. The contributions are sent to the other party, whereas the states are kept secret. In the *converge algorithms* AConv and BConv, Alice and Bob use their own proper state and the other party's contribution to obtain a view of the shared noisy key: $S_A \leftarrow \mathsf{AConv}(A\_state, B\_contr)$ and $S_B \leftarrow \mathsf{BConv}(B\_state, A\_contr)$. Without loss of generality, we assume that S_A and S_B are bit strings of length ℓ. If all goes well, the two views of the session key are *close*, or specifically, different in at most t bits: $\textsc{hw}(S_A \oplus S_B) \leq t$.

[5] Consider for instance Peikert's invited talk at TCC 2009 [53] or Alekhnovich's FOCS 2003 paper [5].

Definition 1 (noisy key agreement protocol). *A* noisy key agreement *protocol between two parties A and B is a tuple $\mathit{\Pi} = (\mathsf{Init}, \mathsf{AContr}, \mathsf{BContr}, \mathsf{AConv}, \mathsf{BConv})$ of five polynomial-time algorithms where the first three are probabilistic and the last two are deterministic. The algorithms are associated with spaces ParSp, $\mathsf{ContrSp}$, $\mathsf{StateSp}$, $\{0,1\}^\ell$ and have type signatures as follows (omitting the random coins and where λ is the security parameter).*

- $\mathsf{Init} : \{1^\lambda\} \to \mathsf{ParSp}$
- $\mathsf{AContr}, \mathsf{BContr} : \mathsf{ParSp} \to \mathsf{StateSp} \times \mathsf{ContrSp}$
- $\mathsf{AConv}, \mathsf{BConv} : \mathsf{StateSp} \times \mathsf{ContrSp} \to \{0,1\}^\ell$

The algorithms are such that, with respect to a noise level $t \leq \ell/2$ *and* correctness error ϵ,

$$\Pr\left[\textsc{hw}(S_A \oplus S_B) \leq t \;\middle|\; \begin{array}{l} \mathit{iparams} \leftarrow \mathsf{Init}(1^\lambda) \\ \mathit{A\_state}, \mathit{A\_contr} \leftarrow \mathsf{AContr}(\mathit{iparams}) \\ \mathit{B\_state}, \mathit{B\_contr} \leftarrow \mathsf{BContr}(\mathit{iparams}) \\ S_A \leftarrow \mathsf{AConv}(\mathit{A\_state}, \mathit{B\_contr}) \\ S_B \leftarrow \mathsf{BConv}(\mathit{B\_state}, \mathit{A\_contr}) \end{array} \right] \geq 1 - \epsilon \ , \quad (3)$$

where $\textsc{hw}\colon \{0,1\}^* \to \mathbb{N}$ *is the Hamming weight function.*

3.2 Generic Problems

The NKA syntax defines three attackable secrets whose recovery is sufficient to undermine the security of the protocol. Also, since the shared secret is what is used in a subsequent module, we note that distinguishing it from random may be a fourth viable attack in many circumstances. We capture these attack strategies in the language of generic problems whose average-case hardness is a necessary condition for security. Any instantiation of NKA therefore defines concrete instantiations of of these hard problems, which then induce concrete average-case hardness assumptions which are necessary for that protocol's security.

The first pair of problems is to recover Alice's secret state from their protocol contribution. If AContr and BContr are identical, then so are these two problems. In the standard Diffie-Hellman key agreement protocol, these problems boil down to the discrete logarithm problem: to obtain a from p, g, and $g^a \bmod p$.

A State Recovery (ASR).
Input: $\mathit{iparams}$, $\mathit{A\_contr}$
Task: find $\mathit{A\_state}$.

B State Recovery (BSR).
Input: $\mathit{iparams}$, $\mathit{B\_contr}$
Task: find $\mathit{B\_state}$.

The next problem captures the task of finding the agreed-upon session key, or a similar enough bit string, from all public information. In the standard Diffie-Hellman key agreement protocol, this problem is essentially the computational Diffie-Hellman problem, *i.e.*, asking to obtain g^{ab} from g, g^a and g^b (all $\bmod\, p$).

Noisy Key Search (NKS).
Input: $\mathit{iparams}$, $\mathit{A\_contr}$, $\mathit{B\_contr}$
Task: find $S \in \{0,1\}^\ell$ such that $\textsc{hw}(S \oplus S_A) \leq t$ and $\textsc{hw}(S \oplus S_B) \leq t$.

Like the state recovery problems, the noisy key problem comes with a decisional variant. This problem captures the task of determining whether a candidate session key is close enough to Alice's and Bob's views.

Noisy Key Distinguishing (NKD).
Input: $\mathit{iparams}$, A_contr, B_contr, S; where if $b = 0$, $S \xleftarrow{\$} \{S \,|\, \textsc{hw}(S \oplus S_A) \leq t \wedge \textsc{hw}(S \oplus S_B) \leq t\}$, and if $b = 1$, $S \xleftarrow{\$} \{0,1\}^\ell$
Task: output 1 if $\textsc{hw}(S \oplus S_A) \leq t$ and $\textsc{hw}(S \oplus S_B) \leq t$; and 0 otherwise.

Clearly, a solver for ASR or for BSR can be used to solve NKS; and a solver for NKS can be used to solve NKD. Therefore, the strongest assumption associated to these problems is assuming that NKD is hard.

Assumption 1 (NKD assumption). *The given NKA protocol* $\Pi = (\mathsf{Init}, \mathsf{AContr}, \mathsf{BContr}, \mathsf{AConv}, \mathsf{BConv})$ *with noise level* t *and correctness error* ϵ *is such that for all polynomial time adversaries* A *in the* NKD *game (Game 4), their advantage* $\mathsf{Adv}_\Pi^{\mathsf{NKD}}(\mathsf{A})$ *is negligible:*

$$\mathsf{Adv}_\Pi^{\mathsf{NKD}}(\mathsf{A}) \triangleq \left| \Pr[\mathsf{Game}_{\mathsf{NKD}}^{\mathsf{A}}(1^\lambda) \not\Rightarrow 0] - \frac{1+\epsilon}{2} \right| \leq \mathsf{negl}(\lambda) \ . \tag{4}$$

When the argument is omitted, the expression denotes the maximum of this quantity across all quantum polynomial-time adversaries: $\mathsf{Adv}_\Pi^{\mathsf{NKD}} \triangleq \max_{\mathsf{A}} \mathsf{Adv}_\Pi^{\mathsf{NKD}}(\mathsf{A})$.

Game 4: $\mathsf{NKD}^{\mathsf{A}}(1^\lambda)$

```
$\mathit{iparams} \leftarrow \mathsf{Init}(1^\lambda)$
$A\_\mathit{state}, A\_\mathit{contr} \leftarrow \mathsf{AContr}(\mathit{iparams})$
$B\_\mathit{state}, B\_\mathit{contr} \leftarrow \mathsf{BContr}(\mathit{iparams})$
$S_A \leftarrow \mathsf{AConv}(A\_\mathit{state}, B\_\mathit{contr})$
$S_B \leftarrow \mathsf{BConv}(B\_\mathit{state}, A\_\mathit{contr})$
if $\textsc{hw}(S_A \oplus S_B) > t$ then:
    return $\perp$
$b \xleftarrow{\$} \{0,1\}$
if $b = 1$ then:
    $S \xleftarrow{\$} \{x \in \{0,1\}^\ell \,|\textsc{hw}(x \oplus S_A) \leq t \wedge \textsc{hw}(x \oplus S_B) \leq t\}$
else:
    $S \xleftarrow{\$} \{0,1\}^\ell$
$\hat{b} \leftarrow \mathsf{A}(\mathit{iparams}, A\_\mathit{contr}, B\_\mathit{contr}, S)$
return $[\![\hat{b} = b]\!]$
```

An interesting problem arises in the formalization of this assumption when the two parties' views of the session key is *too* different. In other words, whenever $\textsc{hw}(S_A \oplus S_B) > t$. Assumption 1 deals with this issue by aborting and ignoring the adversary in this case, but conservatively counting these events as wins for the adversary.

Whether or not to count these aborts as wins for the adversary is a matter of context. In one extreme, when a failure event occurs all bets are off in terms of security. In the other extreme, security is *only* compromised when the adversary successfully attacks a *successful* session. We choose the first option as it is more conservative and as the alternative implies complex design constraints. Note that an adversary whose strategy is random guess has success probability $\Pr[\mathsf{Game}_{\mathsf{NKD}}^{\mathsf{A}}(1^\lambda) \not\Rightarrow 0] = \epsilon + (1-\epsilon) \cdot \frac{1}{2} = \frac{1+\epsilon}{2}$ and hence advantage 0.

3.3 Security

We define the security of an NKA protocol in terms of the NKD game. This follows the regular Diffie-Hellman case in the *authenticated links model*, where security is based on the DDH assumption.

Definition 2 (security of NKA protocols). *An NKA protocol Π is* secure *if and only if the NKD Assumption holds for Π.*

So far, the identification of security with the NKD game has been justified by two arguments. First, the hardness of NKD implies the hardness of NKS, ASR, and BSR. Second, this identification mirrors the case of regular Diffie-Hellman. We supplement this justification with two more arguments. The next section studies a cryptosystem where ASR/BSR are hard, but which failed because NKD is not. Appendix B considers an alternate definition of security called *noisy key security (NK-security)*, along the lines of the session-key security (SK-security) notion in the authenticated links model of Canetti and Krawcyzk [20]. The conclusion there is that NK-security and the NKD Assumption are equivalent, up to a polynomial factor related to the number of sessions started and corrupted in the NK-security game. These indications strongly suggest that the NKD game is not merely a useful formalism, but an essential point of consideration in the context of noisy key agreement protocols.

3.4 Case Study: CFPKM

CFPKM [50] was a KEM proposal based on polynomial system solving with noise (PoSSoWN) submitted to the NIST project. Despite featuring a proof of security, the cryptosystem was broken within days. Since it implicitly relies on a noisy key agreement protocol, it is worthwhile to study what went wrong through the lens of the generic problems described above. The following description is simplified for clarity.

A CFPKM public key consists of a seed *seed* and a vector $\mathbf{b_1} \in \mathbb{Z}_q^m$, where *seed* is expanded into a list of m quadratic polynomials $\mathcal{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ with small coefficients in n variables $\mathbf{x} = (x_1, \ldots, x_n)$ over $\mathbb{Z}_q$ with q a power of 2. The secret key is a short vector $\mathbf{sa} \in \mathbb{Z}_q^n$ and the vector $\mathbf{b_1}$ is found as $\mathbf{b_1} = \mathcal{F}(\mathbf{sa}) + \mathbf{e_1}$ with $\mathbf{e_1} \in \mathbb{Z}_q^m$ a vector of small random errors. To encapsulate, the user chooses a random short vector $\mathbf{sb} \in \mathbb{Z}_q^n$. The ciphertext is then $\mathcal{F}(\mathbf{sb}) + \mathbf{e_2}$, where $\mathbf{e_2}$ is also a vector of small random errors, in addition to some reconciliation information. The key is obtained as the most significant bits of $\mathbf{b_1} \odot \mathcal{F}(\mathbf{sb})$, where $\odot$ is the component-wise product. The decapsulator obtains the same key by computing $\mathcal{F}(\mathbf{sa}) \odot (\mathcal{F}(\mathbf{sb}) + \mathbf{e_2})$ and taking the most significant bits of this vector's components, and by correcting occasional errors when necessary. We identify the underlying noisy key agreement protocol with functionalities and noisy key views as follows. We use $\mathsf{msb}(\cdot)$ to denote the function that takes the most significant bits from each component of its vector argument.

Init: generate $\mathcal{F}$ from *seed*
AContr: sample $\mathbf{sa}, \mathbf{e_1}$ and transmit $\mathbf{b_1} = \mathcal{F}(\mathbf{sa}) + \mathbf{e_1}$
BContr: sample $\mathbf{sb}, \mathbf{e_2}$ and transmit $\mathbf{b_2} = \mathcal{F}(\mathbf{sb}) + \mathbf{e_2}$
AConv: compute $v_1 = \mathsf{msb}(\mathbf{b_2} \odot \mathcal{F}(\mathbf{sa}))$
BConv: compute $v_2 = \mathsf{msb}(\mathbf{b_1} \odot \mathcal{F}(\mathbf{sb}))$
S_A: v_1
S_B: v_2

This description gives rise to the following instantiations of the abstract hard problems. The state recovery problems are instances of PoSSoWN.

A State Recovery (ASR).
Input: $\mathcal{F}, \mathbf{b_1}$ s.t. $\mathbf{b_1} = \mathcal{F}(\mathbf{sa}) + \mathbf{e_1}$ for some small $\mathbf{e_1}, \mathbf{sa}$
Task: find $\mathbf{sa}, \mathbf{e_1}$ s.t. $\mathbf{b_1} = \mathcal{F}(\mathbf{sa}) + \mathbf{e_1}$

B State Recovery (BSR).
Input: $\mathcal{F}, \mathbf{b_2}$ s.t. $\mathbf{b_2} = \mathcal{F}(\mathbf{sb}) + \mathbf{e_2}$ for some small $\mathbf{e_2}, \mathbf{sb}$
Task: find $\mathbf{sb}, \mathbf{e_2}$ s.t. $\mathbf{b_2} = \mathcal{F}(\mathbf{sb}) + \mathbf{e_2}$

Noisy Key Search (NKS).
Input: $\mathcal{F}, \mathbf{b_1}, \mathbf{b_2}$ such that $\mathbf{b_1} = \mathcal{F}(\mathbf{sa}) + \mathbf{e_1}$ and $\mathbf{b_2} = \mathcal{F}(\mathbf{sb}) + \mathbf{e_2}$ for some short $\mathbf{sa}, \mathbf{sb}, \mathbf{e_1}, \mathbf{e_2}$
Task: find $S \in \{0,1\}^\ell$ such that $\textsc{hw}(S \oplus v_1) \leq t$ and $\textsc{hw}(S \oplus v_2) \leq t$, where $v_1 = \mathsf{msb}(\mathcal{F}(\mathbf{sa}) \odot \mathbf{e_1})$ and $v_2 = \mathsf{msb}(\mathcal{F}(\mathbf{sb}) \odot \mathbf{e_2})$.

Noisy Key Distinguishing (NKD).
Input: $\mathcal{F}, \mathbf{b_1}, \mathbf{b_2}, S$ such that $\mathbf{b_1} = \mathcal{F}(\mathbf{sa}) + \mathbf{e_1}$ and $\mathbf{b_2} = \mathcal{F}(\mathbf{sb}) + \mathbf{e_2}$ for some short $\mathbf{sa}, \mathbf{sb}, \mathbf{e_1}, \mathbf{e_2}$
Task: decide whether $\textsc{hw}(S \oplus v_1) \leq t$ and $\textsc{hw}(S \oplus v_2) \leq t$, where $v_1 = \mathsf{msb}(\mathcal{F}(\mathbf{sa}) \odot \mathbf{e_1})$ and $v_2 = \mathsf{msb}(\mathcal{F}(\mathbf{sb}) \odot \mathbf{e_2})$.

The parameters of CFPKM are chosen to guarantee that the solution of the ASR/BSR problems have an infeasible target complexity. However, our analysis suggests that the hardness of ASR and BSR is not enough. Instead, one must look at NKD and tragically, it turns out that in this case NKD is not hard at all. In fact, the attack actually solves the NKS problem for a large proportion of instances by computing $v = \mathsf{msb}(\mathbf{b_1} \odot \mathbf{b_2})$.

Appendix A presents a similar analysis of several KEMs chosen as suitable representatives for their proper branches of mathematics, and identifies the induced hard problems and associated hardness assumptions. This demonstrates that our syntax and hard problems are generic and indeed capable of capturing a multitude of noisy key agreement based schemes. The examples treated there are not known to be insecure. That is to say: there are no known attacks on the induced NKD problems.

4 NKA to KEM: Generic Construction

This section presents a transformation to obtain a KEM from an NKA protocol. In a nutshell, the public key is one contribution to the protocol. The random coins of the encapsulation algorithm are deterministically derived from its seed $s \in \{0,1\}^\lambda$ via a XOF. This algorithm generates the other protocol contribution and uses his view S_B of the shared noisy key as a one-time pad to mask an encoding (using some error-correcting code) of the seed $s \in \{0,1\}^\lambda$. The decapsulation algorithm derives its own view S_A of the shared noisy session key to undo the one time pad up to some errors, after which it can decode the noisy codeword and obtain the seed s. At this point, the decapsulation algorithm simulates the encapsulation algorithm with the exact same deterministic parameters and verifies that the produced ciphertext is identical to the received one.

The resulting KEM is shown in Algorithms 5, 7, and 8. (For a syntactically correct presentation we split the probabilistic portion of the encapsulation from the deterministic portion.) The transformation's parameters are

- Π, the noisy key agreement protocol with session key length ℓ, noise level t, and correctness error ϵ;
- $\mathcal{C}$, the error-correcting coder and decoder for a $[n \leq \ell, k = \lambda, d > t]$-code;
- $\mathsf{H}_1, \mathsf{H}_2 : \{0,1\}^* \rightarrow \{0,1\}^\lambda$, hash functions;
- $\mathsf{H}_3 : \{0,1\}^\lambda \rightarrow \{0,1\}^*$, a cryptographically secure variable output length function whose output is long enough to derandomize any polynomial-time probabilistic algorithm; this may be instantiated by a XOF but we make abstraction of the output length.

We denote the resulting tuple of algorithms as $\mathcal{K} = \mathsf{SNOTP}(\Pi, \mathcal{C}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3)$.

```
algorithm KeyGen
input: $1^\lambda$ — security parameter
output: $sk$ — secret key
        $pk$ — public key

  $iparams \leftarrow \Pi.\mathsf{Init}(1^\lambda)$
  $A\_state, A\_contr \leftarrow \Pi.\mathsf{AContr}(iparams)$
  $pk \leftarrow (iparams, A\_contr)$
  $sk \leftarrow (A\_state, pk)$
  return $sk, pk$
```

Algorithm 5: Key Generation of the KEM.

```
algorithm DetEncaps
input: $pk = (iparams, A\_contr)$ — public key
       $s \in \{0,1\}^\lambda$ — random seed
output: $k$ — symmetric key
        $c$ — ciphertext

1: $B\_state, B\_contr \leftarrow \Pi.\mathsf{BContr}(iparams; \mathsf{H}_3(s))$
2: $S_B \leftarrow \Pi.\mathsf{BConv}(B\_state, A\_contr)$
3: $e \leftarrow \mathcal{C}.\mathsf{encode}(s)$
4: $c \leftarrow (B\_contr, e \oplus S_B, \mathsf{H}_2(s))$
5: $k \leftarrow \mathsf{H}_1(pk \| s)$
6: return $k, c$
```

Algorithm 6: Deterministic encapsulation algorithm of the KEM.

```
algorithm Encaps
input: $pk = (iparams, A\_contr)$ — public key
output: $k$ — symmetric key
        $c$ — ciphertext

$s \xleftarrow{\$} \{0,1\}^\lambda$
return DetEncaps$(pk, s)$
```

Algorithm 7: Encapsulation algorithm of the KEM.

```
algorithm Decaps
input: $sk = (A\_state, pk)$ — secret key
input: $c = (B\_contr, E, h)$ — ciphertext
output: $k$ — symmetric key if successful, or $\perp$ indicating failure

$S_A \leftarrow \Pi.\mathsf{AConv}(A\_state, B\_contr)$
$s \leftarrow \mathcal{C}.\mathsf{decode}(E \oplus S_A)$
if $s = \perp$ or $\mathsf{H}_2(s) \neq h$ then:
    return $\perp$
end
$k, c' \leftarrow \mathsf{DetEncaps}(pk, s)$
if $c' \neq c$ then:
    return $\perp$
end
return $k$
```

Algorithm 8: Decapsulation of the KEM.

Ramstake uses a slight variant of this transformation [50]. The change there is in line 5 of DetEncaps where k is computed as $k \leftarrow \mathsf{H}_1(pk \| coins)$ instead, where

$coins = \mathsf{H}_3(s)$, *i.e.*, the same coins with which $\mathsf{DetEncaps}$ was derandomized. It is clear that this change does not degrade security, for example by setting $\mathsf{H}_1(pk\|s) = \mathsf{H}_1'(pk\|\mathsf{H}_3(s))$.

4.1 Decapsulation Injectivity

Our construction actually achieves something in addition to IND-CCA security: *decapsulation injectivity.* In other words, for any given secret key sk, and for every key k there is (with overwhelming probability) at most one ciphertext c such that $\mathsf{Decaps}(sk, c) = k$. This might sound alarming at first, for instance because it is well known that a public key encryption scheme where every message maps onto one ciphertext cannot be IND-CPA secure, let alone IND-CCA secure.

However, the crucial distinction is that the ciphertexts of KEMs represent encapsulations of *uniformly random* keys. In contrast, PKEs must encrypt *arbitrary* messages, thus enabling the attacker to engineer repeat queries or another attack scenario that requires choosing precisely which messages to encrypt.

Decapsulation injectivity addresses benign malleability, which is the ability of an attacker to modify ciphertexts only if the encapsulated key remains intact. Schemes based on noisy key agreement are inherently resilient to noise and as a result, a ciphertext with added noise may still decapsulate correctly. Also, in some cases the mathematical objects on which the protocol relies, do not have a unique bit-level representation; in this case an adversary can switch representations to obtain a ciphertext that decapsulates to the same key. IND-CCA alone is not sufficient to preclude benign malleability or attacks exploiting it.

Theorem 1 (correctness). *Let Π be an NKA protocol with failure probability ϵ. The failure probability of the KEM $\mathcal{K} = \mathsf{SNOTP}(\Pi, \mathcal{C}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3)$ is*

$$\Pr\left[k_c \neq k_d \;\middle|\; \begin{array}{l} sk, pk \leftarrow \mathsf{KeyGen}(1^\lambda) \\ k_e, c \leftarrow \mathsf{Encaps}(pk) \\ k_d \leftarrow \mathsf{Decaps}(sk, c) \end{array} \right] = \epsilon \;\; . \tag{5}$$

Proof. By construction, we have $pk = (iparams, A\_contr)$, $sk = (A\_state, pk)$ and $c = (B\_contr, S_B \oplus \mathcal{C}.\mathsf{encode}(s), \mathsf{H}_2(s))$, where c is deterministically generated from s and pk. Moreover, the encapsulator finds $k = \mathsf{H}_1(pk\|s)$. The decapsulator then computes $S_A = \Pi.\mathsf{AConv}(A\_state, B\_contr)$ and with probability ϵ, the strings S_A and S_B will lie too far apart for correct decoding. However, if $\textsc{hw}(S_A \oplus S_B) \leq t$, then the decapsulator obtains the correct s from which he can produce the exact same ciphertext as well as $k = \mathsf{H}_1(pk\|s)$. In other words, there is a KEM decapsulation failure only when there is an NKA protocol failure. □

5 NKA to KEM: Security Analysis

5.1 Techniques

We first explain some tools used in the proof before presenting the proof itself.

Inversion. The task of the simulator is to find a preimage x for an output image $y = \mathsf{H}(x)$ that was also output by the simulated algorithm H. In the classical random oracle model, the simulator B can peruse the list of queries made by A to H and test each such query x_i for $\mathsf{H}(x_i) \overset{?}{=} y$.

In the quantum random oracle model, this list of queries is ill-defined because the queries themselves may be represented by quantum superposition states. Instead, it is possible to accomplish the same thing by replacing the random function with a random polynomial $\mathsf{H} \in \mathbb{F}_{2^\ell}[x]$ of degree $2\hat{Q} - 1$, where $\hat{Q}$ is the number of queries made by A to H. Given the output image y, the simulator can factor $\mathsf{H}(x) - y$ in polynomial time to obtain the a list of at most $2\hat{Q} - 1$ candidates $\{x_i\}_{i=0}^{2\hat{Q}-2}$. By selecting one at random, the simulator obtains the correct preimage with probability $\frac{1}{2\hat{Q}-1}$. Zhandry shows that $2\hat{Q}$-wise independent functions (such as this polynomial) are perfectly indistinguishable from a random function [67]. To the best of our knowledge, this technique for inversion in the quantum random oracle model was first used by Unruh for his non-interactivity transform [63].

Insecurity of One-Wayness. Recall that in the One-Wayness game, the challenger samples a random preimage x and runs the adversary on input $\mathsf{H}(x)$. The adversary wins if he outputs a y such that $\mathsf{H}(x) = y$. To capture the hardness of this task, we use a result by Unruh [63]. Here the adversary is given access to a random Bernoulli-distributed function $\mathsf{F} : \{0,1\}^* \rightarrow \{0,1\}$ and each $\mathsf{F}(x)$ is independently Bernoulli-distributed with $\Pr[\mathsf{F}(x) = 1] = \gamma$. For any quantum adversary A making at most $\hat{Q}$ queries, $\Pr[\mathsf{F}(\mathsf{A}^{\mathsf{F}}()) = 1] \leq 2(\hat{Q}+1)\sqrt{\gamma}$. An adversary finding a preimage x of y in the One-Wayness game is simultaneously finding a preimage x of 1 for the Bernoulli-distributed function $\mathsf{F}(x) = [\![\mathsf{H}(x) = y]\!]$, and so $\Pr[\mathsf{H}(\mathsf{A}^{\mathsf{H}}(\mathsf{H}(x))) = \mathsf{H}(x)] \leq 2(\hat{Q} + 1)\sqrt{2^{-n}}$, where n is the output length of H.

Insecurity of Collision Resistance. In the collision resistance game, the adversary oracle access to a function H and is tasked with finding a pair of colliding preimages x_1 and x_2, *i.e.*, such that $\mathsf{H}(x_1) = \mathsf{H}(x_2)$. The success probability of any adversary making at most $\hat{Q}$ queries is bounded by $\Pr[\mathsf{H}(x_1) = \mathsf{H}(x_2) \,|\, x_1, x_2 \leftarrow \mathsf{A}^{\mathsf{H}}()] \leq C(\hat{Q}+1)^3 2^{-n}$, for some universal constant C and where n is the output length of H [68].

Aggregate quantum query amplitude. Our proof relies in part on the indistinguishability of two worlds predicated on a certain value s not being queried to the random oracle. Classically, we can define $b_{k,s} \in \{0,1\}$ as the Boolean value that takes the value 1 in the worlds where the value of query k is s, and 0 in the worlds where it is not, and then proceed to make a distinction depending on whether the aggregation $a_s = \bigvee_k b_{k,s}$ equals 1. In the quantum case, however, these variables are ill-defined because each query does not have an associated value but an associated quantum state, which might be a superposition of many values with possibly non-uniform amplitudes. Nevertheless, we show that the argument can be made to work (even in the quantum random oracle model) when we look instead at these variables' expectation value $\mathrm{E}[b_{k,s}] \in \mathbb{R}_{\geq 0}$. To this end, we define the *quantum query amplitude* $\hat{b}_{k,s} \in \mathbb{C}$ at the kth query associated with a set $\mathcal{S}$ of potential values, and its aggregate across all queries $\hat{a}_s$, in a way that mirrors (but does not capture) the classical notion.

Definition 3 (aggregate quantum query amplitude). *Let A^{H} be a quantum algorithm with oracle access to H making $\hat{Q}$ queries. In particular, A consists of $\hat{Q}+1$ unitary transforms $U_0, \ldots, U_{\hat{Q}}$ operating on a triple of quantum registers S, Q, R, and interleaved with unitaries H operating only on Q, R and sending $|q, r\rangle \mapsto |q, r \oplus \mathsf{H}(q)\rangle$. Let ρ_k^Q represent the reduced density matrix with respect to Q immediately after query k, with query indexation starting at zero. Then the* aggregate quantum query amplitude *$\hat{a}_{\mathcal{S}}$ associated with a set $\mathcal{S}$ of potential queries is*

$$\hat{a}_{\mathcal{S}} = \sum_{k=0}^{\hat{Q}-1} \sqrt{\sum_{s \in \mathcal{S}} \langle s | \rho_k^Q | s \rangle} \ . \tag{6}$$

The aggregate quantum query amplitude is useful as a standalone concept because it enables arguments that consider the degree to which an adversary is querying some member of a set $\mathcal{S}$ and how this quantity changes as this set is modified. The following two lemmas illustrate this fact.

Lemma 1. *For any two sets $\mathcal{S}_1, \mathcal{S}_2 \subseteq \{0,1\}^*$, $\hat{a}_{\mathcal{S}_1} \leq \hat{a}_{\mathcal{S}_1 \cup \mathcal{S}_2}$.*

Proof. Since $\langle s | \rho_k^Q | s \rangle$ is a positive quantity for any s, increasing the range of the sum from $\mathcal{S}_1$ to $\mathcal{S}_1 \cup \mathcal{S}_2$ can only make the sum larger. □

Lemma 2. *For any two sets $\mathcal{S}_1, \mathcal{S}_2 \subset \{0,1\}^*$, if $\hat{a}_{\mathcal{S}_1} \leq 1$ and $\hat{a}_{\mathcal{S}_2} \leq 1$ then $\hat{a}_{\mathcal{S}_1 \cup \mathcal{S}_2} \leq \hat{a}_{\mathcal{S}_1} + \hat{a}_{\mathcal{S}_2}$.*

Proof. Overload "$\backslash$" such that $\mathcal{S}_2 \backslash \mathcal{S}_1 \stackrel{\triangle}{=} \mathcal{S}_2 \backslash (\mathcal{S}_2 \cap \mathcal{S}_1)$. Then we have

$$\hat{a}_{\mathcal{S}_1 \cup \mathcal{S}_2} = \sum_{k=0}^{\hat{Q}-1} \sqrt{\sum_{s \in \mathcal{S}_1 \cup \mathcal{S}_2} \langle s | \rho_k^Q | s \rangle} = \sum_{k=0}^{\hat{Q}-1} \sqrt{\sum_{s \in \mathcal{S}_1} \langle s | \rho_k^Q | s \rangle + \sum_{s \in \mathcal{S}_2 \backslash \mathcal{S}_1} \langle s | \rho_k^Q | s \rangle} \tag{7}$$

$$\leq \sum_{k=0}^{\hat{Q}-1} \sqrt{\sum_{s \in \mathcal{S}_1} \langle s | \rho_k^Q | s \rangle} + \sum_{k=0}^{\hat{Q}-1} \sqrt{\sum_{s \in \mathcal{S}_2 \backslash \mathcal{S}_1} \langle s | \rho_k^Q | s \rangle} = \hat{a}_{\mathcal{S}_1} + \hat{a}_{\mathcal{S}_2 \backslash \mathcal{S}_1} \tag{8}$$

$$\leq \hat{a}_{\mathcal{S}_1} + \hat{a}_{\mathcal{S}_2} \ . \tag{9}$$

The first inequality holds because the terms in the square root are smaller than 1 because $\hat{a}_{\mathcal{S}_1} \leq 1$ and $\hat{a}_{\mathcal{S}_2 \backslash \mathcal{S}_1} \leq \hat{a}_{\mathcal{S}_2} \leq 1$. The second holds due to lemma 1. □

We now upper-bound the trace distance of any pair of quantum distinguishers $\mathsf{D}^{\mathsf{H}_b}$ with oracle access to H_b for some $b \in \{0,1\}$, where $\mathsf{H}_0(x) \neq \mathsf{H}_1(x) \implies x \in \mathcal{S}$, in terms of $\hat{a}_{\mathcal{S}}$. This trace distance in turn upper bounds the maximum distinguishing advantage across all adversaries. The following proof draws in large part on [8, Lemma 37].

Lemma 3. *Let* D *be a quantum distinguisher making at most* $\hat{Q}$ *queries to one of two oracles* $\mathsf{H}_0, \mathsf{H}_1$*, whose outputs differ only on a set* $\mathcal{S}$ *of inputs. Then the trace distance of the distinguishers' final states is bounded by*

$$\mathrm{TD}(\mathsf{D}^{\mathsf{H}_1}(), \mathsf{D}^{\mathsf{H}_2}()) \leq 2\hat{a}_{\mathcal{S}} \ . \tag{10}$$

Proof. Without loss of generality, D uses three registers S, Q, R for its state, and consists of unitary transformations $\{U_k\}_{k=0}^{\hat{Q}}$ operating on all three registers interleaved with oracle queries, which are also unitary transformations H_b but which operate only on Q, R and map $|q, r\rangle \mapsto |q, r \oplus \mathsf{H}_b(q)\rangle$. So if $|\Psi_0\rangle$ is the adversary's initial state, then its final state is given by $|\Psi_b^{\hat{Q}}\rangle = \left(\prod_{k=0}^{\hat{Q}-1} U_{\hat{Q}-k} H_b\right) |\Psi_0\rangle$.

Let $|\Psi_b^i\rangle = \left(\prod_{k=0}^{i} U_{i-k} H_b\right) |\Psi_0\rangle$ be the state before query number i (with indexation of queries starting at 0), and let $|\Psi_b^{\hat{Q}}\rangle$ denote the final state. Define the trace distance at stage i as

$$D_i = \mathrm{TD}(|\Psi_0^i\rangle, |\Psi_1^i\rangle) \ . \tag{11}$$

And then

$$\begin{aligned}
D_i &= \mathrm{TD}(|\Psi_0^i\rangle, |\Psi_1^i\rangle) && (12)\\
&= \mathrm{TD}(U_i H_0 |\Psi_0^{i-1}\rangle, U_i H_1 |\Psi_1^{i-1}\rangle) && (13)\\
&= \mathrm{TD}(H_0 |\Psi_0^{i-1}\rangle, H_1 |\Psi_1^{i-1}\rangle) && (14)\\
&\leq \mathrm{TD}(H_0 |\Psi_0^{i-1}\rangle, H_1 |\Psi_0^{i-1}\rangle) + \mathrm{TD}(H_1 |\Psi_0^{i-1}\rangle, H_1 |\Psi_1^{i-1}\rangle) && (15)\\
&= \mathrm{TD}(H_0 |\Psi_0^{i-1}\rangle, H_1 |\Psi_0^{i-1}\rangle) + \mathrm{TD}(|\Psi_0^{i-1}\rangle, |\Psi_1^{i-1}\rangle) && (16)\\
&= \mathrm{TD}(H_0 |\Psi_0^{i-1}\rangle, H_1 |\Psi_0^{i-1}\rangle) + D_{i-1} \ , && (17)
\end{aligned}$$

where the triangle inequality is used (15). Moreover, since $D_0 = 0$, we have

$$\mathrm{TD}(\mathsf{D}^{\mathsf{H}_0}(), \mathsf{D}^{\mathsf{H}_1}()) = D_{\hat{Q}} \leq \sum_{i=0}^{\hat{Q}-1} \mathrm{TD}(H_0 |\Psi_0^i\rangle, H_1 |\Psi_0^i\rangle) \ . \tag{18}$$

Now consider the projection operator $P_{\mathcal{S}}$ which operates on Q and projects onto the span of all $|s\rangle$ where $s \in \mathcal{S}$. Formally, $P_{\mathcal{S}} = \sum_{s \in \mathcal{S}} I_{(S)} \otimes |s\rangle\langle s| \otimes I_{(R)}$. Let $P_{\bar{\mathcal{S}}}$ be its complement, *i.e.*, $P_{\bar{\mathcal{S}}} = \sum_{s \notin \mathcal{S}} I_{(S)} \otimes |s\rangle\langle s| \otimes I_{(R)}$. We use the symbol z to represent values contained in register S; r for values in R; and both q and s for values in Q.

$$\begin{aligned}
\mathrm{TD}(H_0 |\Psi_0^i\rangle, H_1 |\Psi_0^i\rangle) &= \mathrm{TD}((P_{\mathcal{S}} + P_{\bar{\mathcal{S}}}) H_0 |\Psi_0^i\rangle, (P_{\mathcal{S}} + P_{\bar{\mathcal{S}}}) H_1 |\Psi_0^i\rangle) && (19)\\
&= \mathrm{TD}(P_{\mathcal{S}} H_0 |\Psi_0^i\rangle + P_{\bar{\mathcal{S}}} H_0 |\Psi_0^i\rangle, P_{\mathcal{S}} H_1 |\Psi_0^i\rangle + P_{\bar{\mathcal{S}}} H_1 |\Psi_0^i\rangle) && (20)\\
&= \mathrm{TD}(P_{\mathcal{S}} H_0 |\Psi_0^i\rangle + P_{\bar{\mathcal{S}}} H_0 |\Psi_0^i\rangle, P_{\mathcal{S}} H_1 |\Psi_0^i\rangle + P_{\bar{\mathcal{S}}} H_0 |\Psi_0^i\rangle) && (21)
\end{aligned}$$

$$\begin{aligned}
&\leq 2\|P_{\mathcal{S}}H_0|\Psi_0^i\rangle\| && (22)\\
&= 2\sqrt{\langle\Psi_0^i|H_0^\dagger P_{\mathcal{S}}^\dagger P_{\mathcal{S}}H_0|\Psi_0^i\rangle} && (23)\\
&= 2\sqrt{\sum_{s\in\mathcal{S}}\langle\Psi_0^i|H_0^\dagger(I_{(S)}\otimes|s\rangle\langle s|\otimes I_{(R)})H_0|\Psi_0^i\rangle} && (24)\\
&= 2\sqrt{\sum_{s\in\mathcal{S}}\sum_{z,r}(\langle z|\otimes\langle s|\otimes\langle r|)H_0|\Psi_0^i\rangle\langle\Psi_0^i|H_0^\dagger(|z\rangle\otimes|s\rangle\otimes|r\rangle)} && (25)\\
&= 2\sqrt{\sum_{s\in\mathcal{S}}\langle s|\rho_i^Q|s\rangle}\ . && (26)
\end{aligned}$$

Equation 21 holds because H_0 and H_1 are only different when $q\in\mathcal{S}$, so their effect is the same when projecting onto $\mathsf{span}(\{|s\rangle\}_{s\notin\mathcal{S}})$. The inequality (22) holds due to [8, lemma 35] (with $|\Phi\rangle = P_{\bar{\mathcal{S}}}H_0|\Psi_0^i\rangle$). Equation 26 holds because the reduced density operator of $H_0|\Psi_0^i\rangle = \sum_{z,q,r}\alpha_{z,q,r}|z,q,r\oplus\mathsf{H}_0(q)\rangle$ with respect to register Q is given by

$$\begin{aligned}
\rho_i^Q &= \mathsf{Tr}_{S,R}\left(H_0|\Psi_0^i\rangle\langle\Psi_0^i|H_0^\dagger\right) && (27)\\
&= \sum_{z_1,z_2}\sum_{q_1,q_2}\sum_{r_1,r_2}\alpha_{z_1,q_1,r_1}\alpha^\dagger_{z_2,q_2,r_2}\langle z_1|z_2\rangle\langle r_1\oplus\mathsf{H}_0(q_1)|r_2\oplus\mathsf{H}_0(q_2)\rangle|q_1\rangle\langle q_2| && (28)
\end{aligned}$$

$$\begin{aligned}
&= \sum_{z}\sum_{q_1,q_2}\sum_{r_1,r_2}\alpha_{z,q_1,r_1}\alpha^\dagger_{z,q_2,r_2}\langle z|z\rangle\langle r_1\oplus\mathsf{H}_0(q_1)|r_2\oplus\mathsf{H}_0(q_2)\rangle|q_1\rangle\langle q_2| && (29)\\
&= \sum_{z}\sum_{q_1,q_2}\left(\sum_{r_1,r_2\ |\ r_1\oplus\mathsf{H}_0(q_1)=r_2\oplus\mathsf{H}_0(q_2)}\alpha_{z,q_1,r_1}\alpha^\dagger_{z,q_2,r_2}\right)|q_1\rangle\langle q_2| && (30)\\
&= \sum_{z}\sum_{q_1,q_2}\sum_{r}\alpha_{z,q_1,r\oplus\mathsf{H}_0(q_1)}\alpha^\dagger_{z,q_2,r\oplus\mathsf{H}_0(q_2)}|q_1\rangle\langle q_2|\ . && (31)
\end{aligned}$$

In particular, this means that

$$\begin{aligned}
\sum_{s\in\mathcal{S}}\langle s|\rho_i^Q|s\rangle &= \sum_{s\in\mathcal{S}}\langle s|\left(\sum_{z}\sum_{q_1,q_2}\sum_{r}\alpha_{z,q_1,r\oplus\mathsf{H}_0(q_1)}\alpha^\dagger_{z,q_2,r\oplus\mathsf{H}_0(q_2)}|q_1\rangle\langle q_2|\right)|s\rangle && (32)\\
&= \sum_{s\in\mathcal{S}}\left(\sum_{z}\sum_{q_1,q_2}\sum_{r}\alpha_{z,q_1,r\oplus\mathsf{H}_0(q_1)}\alpha^\dagger_{z,q_2,r\oplus\mathsf{H}_0(q_2)}\langle s|q_1\rangle\langle q_2|s\rangle\right) && (33)\\
&= \sum_{s\in\mathcal{S}}\sum_{z}\sum_{r}\alpha_{z,s,r\oplus\mathsf{H}_0(s)}\alpha^\dagger_{z,s,r\oplus\mathsf{H}_0(s)} && (34)\\
&= \sum_{s\in\mathcal{S}}\sum_{z}\sum_{r}\alpha_{z,s,r\oplus\mathsf{H}_0(s)}\alpha^\dagger_{z,s,r\oplus\mathsf{H}_0(s)}\left(\left(\langle z|\otimes\langle s|\otimes\langle r\oplus\mathsf{H}_0(s)|\right)\left(|z\rangle\otimes|s\rangle\otimes|r\otimes\mathsf{H}_0(s)\rangle\right)\right) && (35)\\
&= \sum_{s\in\mathcal{S}}\sum_{z}\sum_{r}\left(\langle z|\otimes\langle s|\otimes\langle r\oplus\mathsf{H}_0(s)|\right)H_0|\Psi_0^i\rangle\langle\Psi_0^i|H_0^\dagger\left(|z\rangle\otimes|s\rangle\otimes|r\otimes\mathsf{H}_0(s)\rangle\right)\ . && (36)
\end{aligned}$$

Consequently,

$$\mathrm{TD}(\mathsf{D}^{\mathsf{H}_0}(), \mathsf{D}^{\mathsf{H}_1}()) = D_{\hat{Q}} \leq 2 \sum_{k=0}^{\hat{Q}-1} \sqrt{\sum_{s \in \mathcal{S}} \langle s | \rho_k^Q | s \rangle} = 2 \hat{a}_{\mathcal{S}} \ . \qquad \square \quad (37)$$

This theorem shows that if an algorithm A is capable of making a distinction between H_0 and H_1, where H_0 and H_1 differ only on a set $\mathcal{S}$, then $\hat{a}_{\mathcal{S}}$ must be large. The next lemma completes the reasoning by lower-bounding the success probability of an extractor machine who, given black-box access to A, H_0, and H_1, attempts to output some $s \in \mathcal{S}$.

Lemma 4 (Multi-target one-way to hiding). *Let H_0 and H_1 be oracle functions that differ only on input set $\mathcal{S}$, and let A be a quantum adversary that makes at most $\hat{Q}_{\mathsf{H}}$ queries to either H_0 or H_1. Let E be the following algorithm: select $b \xleftarrow{\$} \{0,1\}$ and $k \xleftarrow{\$} \{0, \ldots, \hat{Q}_{\mathsf{H}} - 1\}$ at random, simulate $\mathsf{A}^{\mathsf{H}_b}$ until the kth query, measure the query register in the computational basis, and output the result. Then*

$$\Pr[\mathsf{E}^{\mathsf{A}, \mathsf{H}_0, \mathsf{H}_1}() \Rightarrow s \in \mathcal{S}] \geq \left(\frac{\hat{a}_{\mathcal{S}}}{\hat{Q}_{\mathsf{H}}} \right)^2 \geq \left(\frac{1}{2 \hat{Q}_{\mathsf{H}}} \mathrm{TD}(\mathsf{A}^{\mathsf{H}_0}(), \mathsf{A}^{\mathsf{H}_1}()) \right)^2 \ . \qquad (38)$$

Proof. The probability that E outputs a member of $\mathcal{S}$ is given by

$$\begin{aligned} \Pr[\mathsf{E}^{\mathsf{A}, \mathsf{H}_0, \mathsf{H}_1}() \Rightarrow s \in \mathcal{S}] &= \sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \sum_{s \in \mathcal{S}} \Pr[\mathsf{E}^{\mathsf{A}, \mathsf{H}_0, \mathsf{H}_1}() \Rightarrow s \ \wedge \ \mathsf{E} \, \textit{chooses} \, k] && (39) \\ &= \sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \sum_{s \in \mathcal{S}} \langle s | \rho_k^Q | s \rangle \cdot \frac{1}{\hat{Q}_{\mathsf{H}}} \ . && (40) \end{aligned}$$

Compare with $\hat{a}_{\mathcal{S}}$, which is bounded by via Jensen's inequality by

$$\begin{aligned} \hat{a}_{\mathcal{S}} = \sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \sqrt{\sum_{s \in \mathcal{S}} \langle s | \rho_k^Q | s \rangle} &= \hat{Q}_{\mathsf{H}} \sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \frac{1}{\hat{Q}_{\mathsf{H}}} \sqrt{\sum_{s \in \mathcal{S}} \langle s | \rho_k^Q | s \rangle} && (41) \\ &\leq \hat{Q}_{\mathsf{H}} \sqrt{\sum_{k=0}^{\hat{Q}_{\mathsf{H}}-1} \frac{1}{\hat{Q}_{\mathsf{H}}} \sum_{s \in \mathcal{S}} \langle s | \rho_k^Q | s \rangle} \ . && (42) \end{aligned}$$

Plugging in Eqn. 40 and Lemma 3 yields the theorem statement. □

We draw attention to some differences with respect to Unruh's one-way to hiding lemma [64]. First, our lemma works with an arbitrary potential query set $\mathcal{S}$, whereas Unruh's lemma works only for a single query. Second, our lemma does not assume H_0 and H_1 are random functions per se, but only that they are black boxes accessed as oracles. Third, in Unruh's lemma the adversary A

has access to only one random oracle H and his input is the query-response pair (x, z), where either $z = \mathsf{H}(x)$ or $z = y \neq \mathsf{H}(x)$, and his task is to decide which is the case. In our lemma the distinguisher D is tasked with distinguishing which of two different oracles he has access to. This difference is immaterial, however, since one can used to derive the other. In fact, Unruh's original proof starts by translating the problem into distinguishing two oracles that differ only on x.

5.2 Security Reduction

The security bound involves two parameters determined by the NKA protocol: ϵ and ϕ. The first is the failure probability. The second warrants some explanation. In the NKD game when $b = 0$, S is sampled uniformly at random. However, there is a small probability that this uniform S happens to lie in the radius-t sphere centered at S_A, and in this case the adversary might decide that the ciphertext is correctly formed or decapsulate it outright and indicate incorrectly that $b = 1$. We therefore capture this probability explicitly: $\phi = \left(\sum_{k=0}^{t} \binom{\ell}{k}\right) / 2^{\ell}$.

The construction involves two hash functions, H_1 and H_2, and one variable output function, H_3. In the security argument these are modeled as random oracles.

Theorem 2 (IND-CCA security if NKD Assumption holds). *Let* A *be a quantum adversary in the* $\mathsf{IND\text{-}CCA}$ *game against* $\mathcal{K} = \mathsf{SNOTP}(\Pi, \mathcal{C}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3)$. *Let* $Q_d, \hat{Q}_{\mathsf{H}_1}, \hat{Q}_{\mathsf{H}_2}, \hat{Q}_{\mathsf{H}_3}$ *be its number of queries to the decapsulation oracle,* H_1, H_2 *and* H_3*, respectively. Let* ℓ*,* t *and* ϵ *be the session key length, noise threshold, and failure probability of the NKA protocol* Π*. Then the advantage* $\mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A})$ *is upper bounded by*

$$\mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A}) \leq \frac{2\epsilon + \phi - \epsilon\phi}{2(1-\phi)(1-\epsilon)} + \frac{3 - 2\epsilon - 2\phi + 2\epsilon\phi}{(1-\epsilon)(1-\phi)} \mathsf{Adv}_{\Pi}^{\mathsf{NKD}} + C(\hat{Q}_{\mathsf{H}_2} + 1)^3 2^{-\lambda}$$
$$+ 2\hat{Q}_{\mathsf{H}_3}\sqrt{2(\hat{Q}_{\mathsf{H}_1} + 1)\sqrt{2^{-\lambda}}} + 2\hat{Q}_{\mathsf{H}_3}\sqrt{2(\hat{Q}_{\mathsf{H}_2} + 1)\sqrt{2^{-\lambda}}} + 4\hat{Q}_{\mathsf{H}_3}\sqrt{\mathsf{Adv}_{\Pi}^{\mathsf{NKD}}} \quad (43)$$

in the quantum random oracle model, where C *is the constant of collision resistance insecurity.*

Proof. The proof follows from a sequence of games arguments. At each iteration, a simulator is simulating the previous game and the previous game's adversary in order to win the next game.

- $\mathsf{Game\ 1}$ is identical to the $\mathsf{IND\text{-}CCA}$ for KEMs game against $\mathcal{K}$. So by definition,
$$\Pr[\mathsf{Game\,1}^{\mathsf{A}^{\mathsf{D}(\cdot)}}(1^{\lambda}) \Rightarrow 1] = \mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A}) \ . \quad (44)$$
- $\mathsf{Game\ 2}$ is the $\mathsf{IND\text{-}CPA}$ game against a variant of the KEM that drops derandomization. In particular, there are three modifications: a) the modified algorithm $\mathsf{DetEncaps}'$ is identical to $\mathsf{DetEncaps}$ except for line 1, which becomes

```
1: $B\_state$, $B\_contr \leftarrow \mathit{\Pi}.\mathsf{BContr}(iparams)$ ;
```

b) Decaps' is identical to Decaps except with lines 6–9 replaced by

```
6: $k \leftarrow \mathsf{H}_1(pk \| s)$ ;
```

c) the hash value h is dropped from the ciphertext and line 3 of Decaps becomes

```
3: if $s = \perp$ then: .
```

The adversary B of $\mathsf{Game\ 2}$ simulates A and is therefore responsible for making A's view of events as close to an authentic run of $\mathsf{Game\ 1}$ as possible. In particular, B forwards all queries to the oracles to its oracles $\mathsf{H}_1, \mathsf{H}_3$ and forwards all responses back. However, B presents A with a backdoored random oracle H_2 which is really a random polynomial of degree at most $2\hat{Q}_{\mathsf{H}_2} - 1$. The purpose of this switch is to be able to answer decapsulation queries as follows.

```
define $\mathsf{D}(q)$ as:
    $B\_contr, E, h \leftarrow q$
    $factors \leftarrow \mathsf{factorize}(\mathsf{H}_2(x) - h)$
    for $x \in factors$ do:
        $k', c' \leftarrow \mathsf{DetEncaps}(pk, x)$
        if $c' = q$ then return $k'$
    return $\perp$
```

Since H_2 is a $2\hat{Q}_{\mathsf{H}_2}$-wise independent function, it is perfectly indistinguishable from a true random oracle as long as at most $\hat{Q}_{\mathsf{H}_2}$ queries are made to it. Consequently, this simulated random oracle does not affect security or winning probability. The simulator's running time does increase as a result of this inversion strategy. For every query, he has to factorize a degree at most $2\hat{Q}_{\mathsf{H}_2} - 1$ polynomial and then for each of the at most $2\hat{Q}_{\mathsf{H}_2} - 1$ factors run the deterministic encapsulation procedure followed by some testing. Nevertheless, this operational cost is still linear in Q_d and polynomial in $\hat{Q}_{\mathsf{H}_2}$.

At some point, the simulator B receives the challenge ciphertext-key pair (c, k), where c is lacking a hash-of-seed h. The simulator appends a random value $h^* \xleftarrow{\$} \{0,1\}^\lambda$ to the ciphertext before forwarding it, along with the challenge key, to the adversary A. The simulator B outputs whatever the adversary A outputs.

The difference in input distribution of A when it is playing $\mathsf{Game\ 1}$ versus when it is being simulated by B is characterized by the fact that no $s' \in \{0,1\}^\lambda$ satisfies $\mathsf{DetEncaps}'(pk, \mathsf{H}_3(s')) = (c, \cdot)$ in the latter case. Therefore, provided that A fails to query H_3 on likely candidates for s, the difference in winning probability of A and B in their proper games is negligible. To formalize this argument, consider the adversary's aggregate quantum query amplitude $\hat{a}_{\mathcal{S}}$ on H_3 for the set $\mathcal{S}$ whose members s' satisfy:

- $\mathsf{H}_1(pk\|s') = k$, or
- $\mathsf{H}_2(s') = h^*$, or
- $\mathit{\Pi}.\mathsf{BContr}(\mathit{iparams}; \mathsf{H}_3(s')) = (B\_contr, \cdot)$, or
- $\mathit{\Pi}.\mathsf{BContr}(\mathit{iparams}; \mathsf{H}_3(s')) = (\cdot, B\_state)$ and $\mathcal{C}.\mathsf{decode}(\mathit{\Pi}.\mathsf{BConv}(A\_contr, B\_state) \oplus E) = s'$, or
- $s' = s$.

This list is exhaustive because any s' that does not satisfy any of these conditions is independent of the provided ciphertext and key. The first bullet point represents $\mathsf{H}_1(pk, \cdot)^{-1}$, the set of preimages of k under $\mathsf{H}_1(pk, \cdot)$. The second bullet point represents $\mathsf{H}_2^{-1}(k)$, the set of preimages of h^* under H_2. The next two bullet points represent the set $\mathcal{S}_{\mathsf{H}_3}$, the set of preimages under H_3 to bitstrings that, when fed as random tape to $\mathit{\Pi}.\mathsf{BContr}$, generate a state B_state or contribution B_contr with which the NKD game is won. The last bullet point indicates that the adversary is querying the payload s, which he obtained from solving the NKS problem to find S and then decoding $S \oplus S_B \oplus \mathcal{C}.\mathsf{encode}(s)$.

By separating the aggregate amplitude along these lines we obtain using lemma 2

$$\hat{a}_{\mathcal{S}} \leq \hat{a}_{\mathsf{H}_1(pk,\cdot)^{-1}(k)} + \hat{a}_{\mathsf{H}_2^{-1}(h^*)} + \hat{a}_{\mathcal{S}_{\mathsf{H}_3}} + \hat{a}_s \ . \tag{45}$$

The first two terms in this expression can be bounded by the an extractor's success probability at winning a One-Wayness game using lemma 4. Specifically,

$$\left(\frac{\hat{a}_{\mathsf{H}_1(pk,\cdot)^{-1}(k)}}{\hat{Q}_{\mathsf{H}_3}}\right)^2 \leq \Pr[\mathsf{E}^{\mathsf{A}}() \Rightarrow s \in \mathsf{H}_1(pk, \cdot)^{-1}(k)] \tag{46}$$

$$\leq 2(\hat{Q}_{\mathsf{H}_1} + 1)\sqrt{2^{-\lambda}} \ , \tag{47}$$

and similarly, $\hat{a}^2_{\mathsf{H}_2^{-1}(h^*)} \leq 2\hat{Q}^2_{\mathsf{H}_3}(\hat{Q}_{\mathsf{H}_2} + 1)\sqrt{2^{-\lambda}}$. With respect to the third term, observe that this gives rise to an extractor machine that solves NKD, so $\hat{a}_{\mathcal{S}_{\mathsf{H}_3}} \leq \hat{Q}_{\mathsf{H}_3}\sqrt{\mathsf{Adv}^{\mathsf{NKD}}_{\mathit{\Pi}}}$. The same is true for the fourth term but in a roundabout manner. Define this fourth extractor machine as follows: E_4 takes an NKD instance $(\mathit{iparams}, A\_contr, B\_contr, S)$ and embeds this instance into a public key and ciphertext in order to simulate the adversary. In particular, the public key is $(\mathit{iparams}, A\_contr)$ and the ciphertext is $(B\_contr, \mathcal{C}.\mathsf{encode}(s) \oplus S, h)$ for randomly chosen s, h. Next, E_4 measures a random query to H_3 in the computational basis and outputs 1 if this measurement yields s and 0 otherwise. If the adversary solves NKS and queries s, then E_4 has a $1/\hat{Q}_{\mathsf{H}_3}$ chance of winning the NKD game. So

$$\hat{a}_s \leq \hat{Q}_{\mathsf{H}_3}\sqrt{\Pr[\mathsf{E}_4 \ \mathit{wins} \ \mathsf{NKD}]} \leq \hat{Q}_{\mathsf{H}_3}\sqrt{\mathsf{Adv}^{\mathsf{NKD}}_{\mathit{\Pi}}} \ . \tag{48}$$

Putting these terms together we obtain

$$\hat{a}_{\mathcal{S}} \leq \hat{Q}_{\mathsf{H}_3}\left(\sqrt{2(\hat{Q}_{\mathsf{H}_1} + 1)\sqrt{2^{-\lambda}}} + \sqrt{2(\hat{Q}_{\mathsf{H}_2} + 1)\sqrt{2^{-\lambda}}} + 2\sqrt{\mathsf{Adv}^{\mathsf{NKD}}_{\mathit{\Pi}}}\right) . \tag{49}$$

Without loss of generality, the behavior of H_1, H_2 and H_3 with respect to inputs $s' \notin \mathcal{S}$ is identical across games 1 and 2; in other words, these functions are only different on members of $\mathcal{S}$. However, the adversary has access to another oracle whose responses can potentially help it distinguish. In particular, if the adversary manages to find one element of a pair (s_1, s_2) such that $(B\_contr', E', h') = \mathsf{DetEncaps}(pk, s_1) = \mathsf{DetEncaps}(pk, s_2)$, then the decapsulation oracle might produce different outputs. In $\mathsf{Game\ 1}$ the oracle will decapsulate using $\mathcal{C}.\mathsf{decode}(\mathit{\Pi}.\mathsf{AConv}(A\_state, B\_contr') \oplus E')$ and obtain s_1 or s_2 via $\mathcal{C}.\mathsf{decode}(S_A \oplus E')$, but the decapsulation oracle from the simulation of B will decapsulate using whichever factor of the polynomial $\mathsf{H}_2(x) - h'$ happens to be the first member of this list to pass the re-encapsulation test. When there is a colliding pair (s_1, s_2) for the query ciphertext, this first factor might be the wrong one.

Nevertheless, it is possible to bound the probability of such a collision. The third component is $h' = \mathsf{H}_2(s_1) = \mathsf{H}_2(s_2)$. So it is possible to turn B into a collision-finder for H_2 by modifying its decapsulation oracle $\mathsf{D}(q)$. Instead of returning the first ciphertext that passes the re-encapsulation test of line 6, it runs through all iterations of the loop first. If there are two (or more) factors that pass this test, all are outputted. If there is only one, then k' is outputted, and otherwise $\perp$.

Consequently, an adversary A that distinguishes $\mathsf{Game\ 1}$ from the simulation of B leads to either a collision for H_2, or to an extractor producing a member of $\mathcal{S}$. This means that the distinguishing advantage of any adversary A across game 1 and game 2 (where it is being simulated by B) can be bounded using lemma 3 and the collision resistance insecurity:

$$|\Pr[\mathsf{Game\,1}^{\mathsf{A}}(1^\lambda) \Rightarrow 1] - \Pr[\mathsf{Game\,2}^{\mathsf{B}^{\mathsf{A}}}(1^\lambda) \Rightarrow 1]| \leq 2\hat{a}_{\mathcal{S}} + C(\hat{Q}_{\mathsf{H}_2} + 1)^3 2^{-\lambda} \tag{50}$$

$$\leq 2\hat{Q}_{\mathsf{H}_3} \left(\sqrt{2(\hat{Q}_{\mathsf{H}_1} + 1)\sqrt{2^{-\lambda}}} + \sqrt{2(\hat{Q}_{\mathsf{H}_2} + 1)\sqrt{2^{-\lambda}}} + 2\sqrt{\mathsf{Adv}_{\mathit{\Pi}}^{\mathsf{NKD}}} \right) + C(\hat{Q}_{\mathsf{H}_2} + 1)^3 2^{-\lambda} \ . \tag{51}$$

- $\mathsf{Game\ 3}$ is the NKD game. The adversary C in this game simulates B and is thus responsible for making B's view of events as close as possible to an authentic execution of $\mathsf{Game\ 2}$. In particular, C uses its input as well as the challenge session key to generate the public key and a challenge ciphertext that transmits a random seed $s \xleftarrow{\$} \{0,1\}^\lambda$. He presents the simulated algorithm B with a random oracle H_1 that is programmed to output $k = \mathsf{H}_1(pk \| s)$ for some randomly chosen $k \xleftarrow{\$} \{0,1\}^\lambda$. At some point the simulated adversary B outputs a bit $\hat{b}$ and the simulator C outputs this same bit.

  If an NKA failure event F occurs, then the simulator C "wins" regardless of the behavior of the adversary B — because its output $\perp$ contributes to the adversary's advantage just as much as the output 1.

  If the adversary B wins with output $\hat{b} = 0$, then the ciphertext $c = (B\_contr, S \oplus \mathcal{C}.\mathsf{encode}(s))$ is not an encapsulation of k and consequently $\mathcal{C}.\mathsf{decode}(S_A \oplus$

$S \oplus \mathcal{C}.\mathsf{encode}(s)) \neq s$. This implies that S_A and S are more than t bits apart, implying that S was chosen randomly because $b = 0$. So the simulator C wins by outputting $\hat{b}$.
If the adversary B wins with output $\hat{b} = 1$, then the ciphertext $c = (B\_contr$, $S \oplus \mathcal{C}.\mathsf{encode}(s))$ is an encapsulation of $k = \mathsf{H}_1(pk\|s)$, meaning that $\mathcal{C}.\mathsf{decode}($ $S_A \oplus S \oplus \mathcal{C}.\mathsf{encode}(s)) = s$. This implies that S_A is t or fewer bits apart from S. This in turn implies one of two things; either that S was chosen from the intersection of spheres centered at S_A or S_B because $b = 1$; or else that S was drawn uniformly at random and happens to lie close to S_A. In the former case, the simulator who outputs $\hat{b} = 1$ wins as well. If $b = 0$, the latter case occurs with a probability $\phi = \left(\sum_{k=0}^{t} \binom{\ell}{k}\right)/2^{\ell}$, *i.e.* the probability of a uniformly random string $S \xleftarrow{\$} \{0,1\}^{\ell}$ having Hamming distance at most t from a given $S_A \in \{0,1\}^{\ell}$.

$$\begin{aligned} &\Pr[\mathsf{Game}_{\mathsf{NKD}}^{\mathsf{C}^{\mathsf{B}}}(1^{\lambda}) \not\Rightarrow 0] = \Pr[\mathsf{Game\,3}^{\mathsf{C}^{\mathsf{B}}}(1^{\lambda}) \not\Rightarrow 0] \\ &= \Pr[F \vee (\neg F \wedge \mathsf{C}^{\mathsf{B}}(1^{\lambda}) \Rightarrow b)] = \Pr[F] + \Pr[\neg F \wedge \mathsf{C}^{\mathsf{B}}(1^{\lambda}) \Rightarrow b] \qquad (52) \\ &\geq \Pr[\neg F] \Pr[\mathsf{C}^{\mathsf{B}}(1^{\lambda}) \Rightarrow b \mid \neg F] \qquad (53) \end{aligned}$$

$$\begin{aligned} &= (1-\epsilon)\left(\Pr[\mathsf{C}^{\mathsf{B}}(1^{\lambda}) \Rightarrow b = 0 \mid \neg F] + \Pr[\mathsf{C}^{\mathsf{B}}(1^{\lambda}) \Rightarrow b = 1 \mid \neg F]\right) \qquad (54) \\ &= \frac{1-\epsilon}{2}\left(\Pr[\mathsf{C}^{\mathsf{B}}(1^{\lambda}) \Rightarrow 0 \mid b = 0 \wedge \neg F] + \Pr[\mathsf{C}^{\mathsf{B}}(1^{\lambda}) \Rightarrow 1 \mid b = 1 \wedge \neg F]\right) \qquad (55) \\ &= \tfrac{1-\epsilon}{2}\left(\Pr[\mathsf{B}(1^{\lambda}) \Rightarrow 0 \wedge \textsc{hw}(S \oplus S_A) > t \mid b = 0 \wedge \neg F] + \Pr[\mathsf{B}(1^{\lambda}) \Rightarrow 1 \mid b = 1 \wedge \neg F]\right) \\ &\qquad (56) \\ &\geq \tfrac{1-\epsilon}{2}\left(\Pr[\mathsf{B}(1^{\lambda}) \Rightarrow 0 \mid b = 0 \wedge \neg F] \cdot (1-\phi) + \Pr[\mathsf{B}(1^{\lambda}) \Rightarrow 1 \mid b = 1 \wedge \neg F]\right) \\ &\qquad (57) \\ &\geq \tfrac{1-\epsilon}{2}\left(\Pr[\mathsf{B}(1^{\lambda}) \Rightarrow 0 \mid b = 0 \wedge \neg F] + \Pr[\mathsf{B}(1^{\lambda}) \Rightarrow 1 \mid b = 1 \wedge \neg F]\right) \cdot (1-\phi) \\ &\qquad (58) \\ &= (1-\phi) \cdot \Pr[\neg F] \cdot \Pr[\mathsf{B}(1^{\lambda}) \Rightarrow b] = (1-\epsilon-\phi+\epsilon\phi) \cdot \Pr[\mathsf{Game\,2}^{\mathsf{B}}(1^{\lambda}) \Rightarrow 1] \\ &\qquad (59) \end{aligned}$$

Now describe $\mathsf{Adv}_{\Pi}^{\mathsf{NKD}}(\mathsf{C}) = \left|\Pr[\mathsf{NKD}_{\Pi}^{\mathsf{C}^{\mathsf{B}}}(1^{\lambda}) \not\Rightarrow 0] - \frac{1+\epsilon}{2}\right|$ in terms of $\mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A})$. Then we get:

$$\begin{aligned} &\mathsf{Adv}_{\Pi}^{\mathsf{NKD}}(\mathsf{C}) = \Pr[\mathsf{Game\,3}^{\mathsf{C}}(1^{\lambda}) \not\Rightarrow 0] - \frac{1+\epsilon}{2} \qquad (60) \\ &\geq (1-\epsilon-\phi+\epsilon\phi) \cdot \Pr[\mathsf{Game\,2}^{\mathsf{B}}(1^{\lambda}) \Rightarrow 1] - \frac{1+\epsilon}{2} \qquad (61) \\ &\geq (1-\epsilon-\phi+\epsilon\phi) \cdot \Bigg(\Pr[\mathsf{Game\,1}^{\mathsf{A}}(1^{\lambda}) \Rightarrow 1] - 2\hat{Q}_{\mathsf{H}_3}\Big(\sqrt{2(\hat{Q}_{\mathsf{H}_1}+1)\sqrt{2^{-\lambda}}} \\ &\quad + \sqrt{2(\hat{Q}_{\mathsf{H}_2}+1)\sqrt{2^{-\lambda}}} + 2\sqrt{\mathsf{Adv}_{\Pi}^{\mathsf{NKD}}}\Big) - C(\hat{Q}_{\mathsf{H}_2}+1)^3 2^{-\lambda}\Bigg) - \frac{1+\epsilon}{2}\ . \qquad (62) \end{aligned}$$

Isolate the term $\Pr[\mathsf{Game}\,1^{\mathsf{A}}(1^\lambda) \Rightarrow 1] = \mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A})$ and use $\mathsf{Adv}_{\Pi}^{\mathsf{NKD}}(\mathsf{C}) \leq \mathsf{Adv}_{\Pi}^{\mathsf{NKD}}$. This yields the theorem statement:

$$\begin{aligned} \mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A}) \leq & \frac{\mathsf{Adv}_{\Pi}^{\mathsf{NKD}} + \frac{1+\epsilon}{2}}{(1-\epsilon)(1-\phi)} - \frac{1}{2} + 2\hat{Q}_{\mathsf{H}_3}\Big(\sqrt{2(\hat{Q}_{\mathsf{H}_1}+1)\sqrt{2^{-\lambda}}} \\ & + \sqrt{2(\hat{Q}_{\mathsf{H}_2}+1)\sqrt{2^{-\lambda}}} + 2\sqrt{\mathsf{Adv}_{\Pi}^{\mathsf{NKD}}}\Big) + C(\hat{Q}_{\mathsf{H}_2}+1)^3 2^{-\lambda} \ . \qquad \square \end{aligned} \tag{63}$$

6 Conclusion

This paper introduces the noisy key agreement (NKA) protocol as a standalone concept, and an appropriate security definition in the form of the NKD game. Furthermore, it presents a transformation turning an NKA protocol into a key encapsulation mechanism (KEM) secure in the quantum random oracle model. The security proof relies on modeling the derandomization function H_3 as a variable output length random oracle, along with new techniques for refined reasoning about the queries made by a quantum adversary and uses the NKA protocol as a starting point.

The bound's reliance on the error probabiliy ϵ is to be expected because the occurrence of a protocol failure is equated to a complete loss of security. However, there is also a term involving ϕ, the probability of a uniformly random bitstring being less than t bits apart from a given one. The presence of this parameter is an artifact of the NKD formalism as $(1-\epsilon)(1-\phi)/2$ upper bounds any adversary's advantage in that game. In practice, both ϵ and ϕ should be made negligible in the security parameter.

Provided that this constraint is satisfied, our bound is much tighter than the those of Targhi-Unruh and Hofheinz *et al.* [61,35]. In particular, the term $\mathsf{Adv}_{\Pi}^{\mathsf{NKD}}$, which captures the insecurity of the underlying primitive, is degraded only by a square root, similar to the bound of Jiang *et al.* [39]. In contrast, the insecurity of the underlying primitive degrades with a quartic root in Targhi-Unruh and Hofheinz *et al.* All roots are the result either of the One-Way to Hiding Lemma or else of the One-Wayness game.

With respect to the concrete security of the Ramstake proposal, a couple of remarks are in order. First, the security bound explicitly features the error probability ϵ which in the case of Ramstake is rather high — roughly 2^{-64} for a security level of 128 bits against quantum computers. The bound therefore establishes less security than the claimed 128 bits. Nevertheless, when conditioning for the absence of decapsulation failures, the bottleneck becomes preimage search in a random function, and after that the NKD advantage. Moreover, it is by no means clear how much and even whether security is lost in the event of a decapsulation failure, although answering this question is a task for cryptanalysis rather than provable security.

Second, length of the hashes and seed is twice the claimed security level, in accordance with a speedup due to Grover's algorithm. However, the security degradation in the present bound resulting from these hash functions is a *fourth* *root*, much better than Grover's algorithm from the attacker's point of view. It remains an open question to determine whether this fourth root degradation is tight, *i.e.*, whether it can be matched by an attack. We note that Hülsing *et al.* [37] have a root-free insecurity function for preimage search applying specifically in the context of compressing hash functions. While their result does not apply in the present context, it is an uplifting indication that maybe the fourth root degradation is not a necessary quality of a security bound.

Acknowledgements

The authors would like to thank Aysajan Abidin, Chris Peikert, Mike Hamburg, Keita Xagawa, and others for useful comments and feedback, missing references, and pointing out flaws in proofs.

References

1. Aggarwal, D., Joux, A., Prakash, A., Santha, M.: A New Public-Key Cryptosystem via Mersenne Numbers. IACR Cryptology ePrint Archive 2017, 481, version 20170530:072202
2. Aguilar, C., Gaborit, P., Lacharme, P., Schrek, J., Zémor, G.: Noisy Diffie-Hellman Protocols (2010), `https://pqc2010.cased.de/rr/03.pdf`, PQCrypto 2010 (recent results session)
3. Aguilar, C., Gaborit, P., Lacharme, P., Schrek, J., Zémor, G.: Noisy Diffie-Hellman protocols or code-based key exchanged and encryption without masking (2010), `https://rump2010.cr.yp.to/fae8cd8265978675893352329786cea2.pdf`, CRYPTO 2010 (rump session)
4. Albrecht, M.R., Orsini, E., Paterson, K.G., Peer, G., Smart, N.P.: Tightly Secure Ring-LWE Based Key Encapsulation with Short Ciphertexts. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) ESORICS 2017, Part I. LNCS, vol. 10492, pp. 29–46. Springer (2017)
5. Alekhnovich, M.: More on average case vs approximation complexity. In: FOCS 2003. pp. 298–307. IEEE Computer Society (2003)
6. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: NewHope without reconciliation. IACR Cryptology ePrint Archive 2016, 1157
7. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum Key Exchange - A New Hope. In: Holz, T., Savage, S. (eds.) USENIX 2016. pp. 327–343. USENIX Association (2016)
8. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum Attacks on Classical Proof Systems: The Hardness of Quantum Rewinding. In: IEEE FOCS 2014. pp. 474–483. IEEE Computer Society (2014)
9. Anada, H., Arita, S.: Identification Schemes from Key Encapsulation Mechanisms. IEICE Transactions 95-A(7), 1136–1155 (2012)
10. Barreto, P.S.L.M., Gueron, S., Gueneysu, T., Misoczki, R., Persichetti, E., Sendrier, N., Tillich, J.: CAKE: Code-based Algorithm for Key Encapsulation. IACR Cryptology ePrint Archive 2017, 757
11. Bellare, M., Hofheinz, D., Kiltz, E.: Subtleties in the Definition of IND-CCA: When and How Should Challenge Decryption Be Disallowed? J. Cryptology 28(1), 29–48 (2015)

12. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: CRYPTO '93. LNCS, vol. 773, pp. 232–249. Springer (1993)
13. Bellare, M., Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS '93. pp. 62–73. ACM (1993)
14. Bernstein, D.J., Chou, T., Schwabe, P.: McBits: Fast Constant-Time Code-Based Cryptography. In: Bertoni, G., Coron, J. (eds.) CHES 2013. LNCS, vol. 8086, pp. 250–272. Springer (2013)
15. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random Oracles in a Quantum World. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer (2011)
16. Bos, J.W., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 2016. pp. 1006–1018. ACM (2016)
17. Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem. In: IEEE S&P 2015. pp. 553–570. IEEE Computer Society (2015)
18. Bos, J.W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Stehlé, D.: CRYSTALS - Kyber: a CCA-secure module-lattice-based KEM. IACR Cryptology ePrint Archive 2017, 634
19. Boyd, C., Cliff, Y., Nieto, J.M.G., Paterson, K.G.: One-round key exchange in the standard model. IJACT 1(3), 181–199 (2009)
20. Canetti, R., Krawczyk, H.: Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 453–474. Springer (2001)
21. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. IACR Cryptology ePrint Archive 2018, 383 (2018)
22. Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In: CRYPTO '98
23. Cramer, R., Shoup, V.: Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. SIAM J. Comput. 33(1), 167–226 (2003)
24. Cremers, C.J.F., Feltz, M.: Beyond eCK: Perfect Forward Secrecy under Actor Compromise and Ephemeral-Key Reveal. Des. Codes Cryptography 74(1), 183–218 (2015)
25. Dagdelen, Ö., Fischlin, M., Gagliardoni, T.: The Fiat-Shamir Transformation in a Quantum World. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 62–81. Springer (2013)
26. Deneuville, J., Gaborit, P., Zémor, G.: Ouroboros: A Simple, Secure and Efficient Key Exchange Protocol Based on Coding Theory. In: Lange, T., Takagi, T. (eds.) PQCrypto 2017. LNCS, vol. 10346, pp. 18–34. Springer (2017)
27. Dent, A.W.: A Designer's Guide to KEMs. In: Paterson, K.G. (ed.) IMA 9th Conf. Cryptography and Coding. LNCS, vol. 2898, pp. 133–151. Springer (2003)
28. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Information Theory 22(6), 644–654 (1976)
29. Ding, J., Lie, X., Lin, X.: A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem. IACR Cryptology ePrint Archive 2012, 688 (2012)

30. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO '86. LNCS, vol. 263, pp. 186–194. Springer (1986)
31. Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Strongly secure authenticated key exchange from factoring, codes, and lattices. Des. Codes Cryptography 76(3), 469–504 (2015)
32. Fujisaki, E., Okamoto, T.: How to Enhance the Security of Public-Key Encryption at Minimum Cost. In: Imai, H., Zheng, Y. (eds.) PKC '99. LNCS, vol. 1560, pp. 53–68. Springer (1999)
33. Grover, L.K.: A Fast Quantum Mechanical Algorithm for Database Search. In: Miller, G.L. (ed.) ACM STOC 1996. pp. 212–219. ACM (1996)
34. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A Ring-Based Public Key Cryptosystem. In: Buhler, J. (ed.) ANTS-III, 1998. LNCS, vol. 1423, pp. 267–288. Springer (1998)
35. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A Modular Analysis of the Fujisaki-Okamoto Transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 341–371. Springer (2017)
36. Hülsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P.: High-Speed Key Encapsulation from NTRU. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 232–252. Springer (2017)
37. Hülsing, A., Rijneveld, J., Song, F.: Mitigating Multi-target Attacks in Hash-Based Signatures. In: Cheng, C., Chung, K., Persiano, G., Yang, B. (eds.) PKC 2016, Part I. LNCS, vol. 9614, pp. 387–416. Springer (2016)
38. Jao, D., Feo, L.D.: Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In: Yang, B. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 19–34. Springer (2011)
39. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: Post-quantum IND-CCA-secure KEM without additional hash. IACR Cryptology ePrint Archive 2017, 1096 (2017)
40. Jin, Z., Zhao, Y.: Optimal Key Consensus in Presence of Noise. CoRR abs/1611.06150 (2016)
41. Krawczyk, H.: SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 400–425. Springer (2003)
42. LaMacchia, B.A., Lauter, K.E., Mityagin, A.: Stronger Security of Authenticated Key Exchange. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 1–16. Springer (2007)
43. Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 319–339. Springer (2011)
44. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. J. ACM 60(6), 43:1–43:35 (2013)
45. Matsumoto, T., Imai, H.: Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. In: Günther, C.G. (ed.) EUROCRYPT '88. LNCS, vol. 330, pp. 419–453. Springer (1988)
46. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. DNS Progress Report 4244, 114–116 (1978)
47. Melchor, C.A., Blazy, O., Deneuville, J., Gaborit, P., Zémor, G.: Efficient Encryption from Random Quasi-Cyclic Codes. CoRR abs/1612.05572 (2016)
48. National Institute for Standards and Technology (NIST): FIPS PUB 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (2015)

49. National Institute for Standards and Technology (NIST): Post-quantum crypto standardization (2018), http://csrc.nist.gov/groups/ST/post-quantum-crypto/
50. National Institute for Standards and Technology (NIST): Submission to the NIST call for PQC proposals. (2018), https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
51. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory. Problemy Upravlenija i Teorii Informacii. 15, 159–166 (1986)
52. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. Cambridge university press (2010)
53. Peikert, C.: Some recent progress in lattice-based cryptography. In: Reingold, O. (ed.) TCC 2009. Lecture Notes in Computer Science, vol. 5444. Springer (2009), invited talk.
54. Peikert, C.: Lattice Cryptography for the Internet. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 197–219. Springer (2014)
55. Porras, J., Baena, J., Ding, J.: Zhfe, a new multivariate public key encryption scheme. In: Mosca, M. (ed.) PQCrypto 2014. Lecture Notes in Computer Science, vol. 8772, pp. 229–245. Springer (2014)
56. Rackoff, C., Simon, D.R.: Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In: Feigenbaum, J. (ed.) CRYPTO '91. LNCS, vol. 576, pp. 433–444. Springer (1991)
57. Shor, P.W.: Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In: FOCS 35. pp. 124–134. IEEE Computer Society (1994)
58. Song, F.: A Note on Quantum Security for Post-Quantum Cryptography. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 246–265. Springer (2014)
59. Szepieniec, A., Ding, J., Preneel, B.: Extension Field Cancellation: A New Central Trapdoor for Multivariate Quadratic Systems. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 182–196. Springer (2016)
60. Tao, C., Diene, A., Tang, S., Ding, J.: Simple Matrix Scheme for Encryption. In: PQCrypto 2013
61. Targhi, E.E., Unruh, D.: Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 192–216 (2016)
62. Tolhuizen, L., Rietman, R., García-Morchón, Ó.: Improved key-reconciliation method. IACR Cryptology ePrint Archive 2017, 295
63. Unruh, D.: Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 755–784. Springer (2015)
64. Unruh, D.: Revocable quantum timed-release encryption. J. ACM 62(6), 49:1–49:76 (2015)
65. Unruh, D.: Computationally Binding Quantum Commitments. In: Fischlin, M., Coron, J. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 497–527. Springer (2016)
66. Yoneyama, K.: Compact Authenticated Key Exchange from Bounded CCA-Secure KEM. IEICE Transactions 98-A(1), 132–143 (2015)
67. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 758–775. Springer (2012)

68. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Information & Computation 15(7&8), 557–567 (2015), http://www.rintonpress.com/xxqic15/qic-15-78/0557-0567.pdf

A Concrete Instantiations of NKA Protocols

We now consider several concrete instantiations of noisy key agreement that are used in the literature to generate key encapsulation mechanisms or public key encryption schemes. In all cases, the participants of the protocol converge to mathematical objects whose distance is small in some sense. We make abstraction of this notion of smallness and represent the mathematical objects as bitstrings (denoted by $\llcorner \cdot \lrcorner$) at which point the Hamming weight metric can be used.

It is worth emphasizing that the concrete problems we identify must be assumed to be hard, even in the context of quantum computers, in order for the protocol and KEM or PKE to be secure. Nevertheless, the NKD Assumption is the only requirement; the other problems are hard on average if the NKD Assumption is true.

NewHope [7]. NewHope defines a ring $\mathcal{R}_q \cong \mathbb{Z}[X]/\langle q, X^n + 1\rangle$ and a centered binomial distrubution Ψ_{16}^n over $\mathcal{R}_q$. Elements that are sampled according to Ψ_{16}^n are considered small. The protocol functionalities and noisy key views are as follows.

Init: generate $\mathbf{a} \in \mathcal{R}_q$ from *seed*
AContr: sample $\mathbf{s}, \mathbf{e} \sim \Psi_{16}^n$ and transmit $\mathbf{b} = \mathbf{as} + \mathbf{e}$
BContr: sample $\mathbf{s}', \mathbf{e}' \sim \Psi_{16}^n$ and transmit $\mathbf{u} = \mathbf{as}' + \mathbf{e}'$
AConv: compute $\mathbf{v} = \mathbf{us}$
BConv: compute $\mathbf{v}' = \mathbf{bs}'$
S_A: $\llcorner \mathbf{v} \lrcorner$
S_B: $\llcorner \mathbf{v}' \lrcorner$

This description gives rise to the following hard problems. The state recovery problems are instances of Ring-LWE.

A State Recovery (ASR).
Input: $\mathbf{a}, \mathbf{b} \in \mathcal{R}_q$ s.t. $\mathbf{b} = \mathbf{as} + \mathbf{e}$ for some $\mathbf{e}, \mathbf{s} \sim \Psi_{16}^n$
Task: find $\mathbf{s}, \mathbf{e} \sim \Psi_{16}^n$ s.t. $\mathbf{b} = \mathbf{as} + \mathbf{e}$

B State Recovery (BSR).
Input: $\mathbf{a}, \mathbf{u} \in \mathcal{R}_q$ s.t. $\mathbf{u} = \mathbf{as}' + \mathbf{e}'$ for some $\mathbf{e}', \mathbf{s}' \sim \Psi_{16}^n$
Task: find $\mathbf{s}, \mathbf{e} \sim \Psi_{16}^n$ s.t. $\mathbf{u} = \mathbf{as}' + \mathbf{e}'$

Noisy Key Search (NKS).
Input: $\mathbf{a}, \mathbf{b}, \mathbf{u} \in \mathcal{R}_q$ such that $\mathbf{b} = \mathbf{as} + \mathbf{e}$ and $\mathbf{u} = \mathbf{as}' + \mathbf{e}'$ for some $\mathbf{s}, \mathbf{s}', \mathbf{e}, \mathbf{e}' \sim \Psi_{16}^n$
Task: find $S \in \{0,1\}^\ell$ such that $\textsc{hw}(S \oplus \llcorner \mathbf{v} \lrcorner) \leq t$ and $\textsc{hw}(S \oplus \llcorner \mathbf{v}' \lrcorner) \leq t$, where $\mathbf{v} = \mathbf{us}$ and $\mathbf{v}' = \mathbf{bs}'$.

Noisy Key Distinguishing (NKD).
Input: $\mathbf{a}, \mathbf{b}, \mathbf{u} \in \mathcal{R}_q$ and $S \in \{0,1\}^\ell$ such that $\mathbf{b} = \mathbf{as} + \mathbf{e}$ and $\mathbf{u} = \mathbf{as}' + \mathbf{e}'$ for some $\mathbf{s}, \mathbf{s}', \mathbf{e}, \mathbf{e}' \sim \Psi_{16}^n$
Task: output 1 if $\textsc{hw}(S \oplus \llcorner \mathbf{v} \lrcorner) \leq t$ and $\textsc{hw}(S \oplus \llcorner \mathbf{v}' \lrcorner) \leq t$, where $\mathbf{v} = \mathbf{us}$ and $\mathbf{v}' = \mathbf{bs}'$; and 0 otherwise.

Ramstake [50]. Ramstake operates on integers modulo a large Mersenne prime p, the set of which we denote by $\mathbb{Z}_p$. Smallness is associated with having a bit expansion of low Hamming weight. We denote this set of sparse integers by $\mathcal{S}$. The functionalities and noisy key views are as follows.

Init: sample $A \in \mathbb{Z}_p$
AContr: sample $b, c \xleftarrow{\$} \mathcal{S}$ and transmit $D = Ab + c \,\mathsf{mod}\, p$
BContr: sample $b', c' \xleftarrow{\$} \mathcal{S}$ and transmit $D' = Ab' + c' \,\mathsf{mod}\, p$
AConv: compute $E = D'a \,\mathsf{mod}\, p$
BConv: compute $E' = Da' \,\mathsf{mod}\, p$
S_A: $\llcorner E \lrcorner$
S_B: $\llcorner E' \lrcorner$

The problems of recovering either participant's state is in fact an affine variant of the low-Hamming weight ratio problem introduced by Aggarwal *et al.* [1]. Paraphrased but without loss of generality, this problem asks to find low-Hamming-weight integers f and g such that the given integer H satisfies $f \times (-H) + g = 0 \,\mathsf{mod}\, p$.

A State Recovery (ASR).
Input: $A, D \in \mathbb{Z}_p$ s.t. $D = Ab + c$ for some $b, c \in \mathcal{S}$
Task: find $b, c \in \mathcal{S}$ s.t. $D = Ab + c$

B State Recovery (BSR).
Input: $A, D' \in \mathbb{Z}_p$ s.t. $D' = Ab' + c'$ for some $b', c' \in \mathcal{S}$
Task: find $b', c' \in \mathcal{S}$ s.t. $D' = Ab' + c'$

Noisy Key Search (NKS).
Input: $A, D, D' \in \mathbb{Z}_p$ such that $D = Ab + c$ and $D' = Ab' + c'$ for some $b, c, b', c' \in \mathcal{S}$
Task: find $S \in \{0,1\}^\ell$ such that $\textsc{hw}(S \oplus \llcorner E \lrcorner) \leq t$ and $\textsc{hw}(S \oplus \llcorner E' \lrcorner) \leq t$, where $E = D'a$ and $E' = Da'$.

Noisy Key Distinguishing (NKD).
Input: $A, D, D' \in \mathbb{Z}_p, S \in \{0,1\}^\ell$ such that $D = Ab + c$ and $D' = Ab' + c'$ for some $b, c, b', c' \in \mathcal{S}$
Task: output 1 if $\textsc{hw}(S \oplus \llcorner E \lrcorner) \leq t$ and $\textsc{hw}(S \oplus \llcorner E' \lrcorner) \leq t$, where $E = D'a$ and $E' = Da'$; and 0 otherwise.

Ouroboros [26]. Ouroboros uses the ring $\mathcal{R} = \mathbb{F}_2[X]/\langle X^n - 1\rangle$, in which elements are considered small if their Hamming weight is less than a given bound. Let $\mathcal{S}_w^n \subset \mathcal{R}$ denote the subset of ring elements whose Hamming weight is w. The functionalities and noisy key views are as follows.

Init: generate $\mathbf{h} \in \mathcal{R}$ from *seed*
AContr: sample $\mathbf{x}, \mathbf{y} \xleftarrow{\$} \mathcal{S}_w^n$ and transmit $\mathbf{s} = \mathbf{x}\mathbf{h} + \mathbf{y}$
BContr: sample $\mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n$ and transmit $\mathbf{s}_r = \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2$
AConv: compute $S_A = \mathbf{y}\mathbf{s}_r$
BConv: compute $S_B = \mathbf{s}\mathbf{r}_2$
S_A: $\llcorner S_A \lrcorner$
S_B: $\llcorner S_B \lrcorner$

While the values S_A and S_B are computed, both are instantly added to other values. Bob obtains $\mathbf{s}_\epsilon = S_B + \mathbf{e}_r + \boldsymbol{\epsilon}$ for specific values of $\mathbf{e}_r$ and $\boldsymbol{\epsilon}$, and transmits this value alongside $\mathbf{s}_r$. Alice obtains $\mathbf{e}_c = \mathbf{s}_\epsilon - S_A$, which is a noisy codeword from which the specialized decoder can recover $\boldsymbol{\epsilon}$. Ouroboros thus uses the transmission-based approach, and makes clever use of the decoder provided by the algebraic structure on which the noisy key agreement protocol is based.

A State Recovery (ASR).
Input: $\mathbf{h}, \mathbf{s} \in \mathcal{R}$ s.t. $\mathbf{s} = \mathbf{xh} + \mathbf{y}$ for some $\mathbf{x}, \mathbf{y} \in \mathcal{S}_w^n$
Task: find $\mathbf{x}, \mathbf{y} \in \mathcal{S}_2^n$ s.t. $\mathbf{s} = \mathbf{hx} + \mathbf{y}$

B State Recovery (BSR).
Input: $\mathbf{h}, \mathbf{s}_r \in \mathcal{R}$ s.t. $\mathbf{s}_r = \mathbf{r}_2\mathbf{h} + \mathbf{r}_2$ for some $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{S}_w^n$
Task: find $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{S}_2^n$ s.t. $\mathbf{s}_2 = \mathbf{hr}_2 + \mathbf{r}_1$

Noisy Key Search (NKS).
Input: $\mathbf{h}, \mathbf{s}, \mathbf{s}_r \in \mathcal{R}$ such that $\mathbf{s} = \mathbf{xh} + \mathbf{y}$ and $\mathbf{s}_r = \mathbf{hr}_2 + \mathbf{r}_1$ for some $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2 \in \mathcal{S}_w^n$
Task: find $S \in \{0,1\}^\ell$ such that $\textsc{hw}(S \oplus \llcorner S_A \lrcorner) \le t$ and $\textsc{hw}(S \oplus \llcorner S_B \lrcorner) \le t$, where $S_A = \mathbf{s}_r\mathbf{x}$ and $S_B = \mathbf{sr}_2$.

Noisy Key Distinguishing (NKD).
Input: $\mathbf{h}, \mathbf{s}, \mathbf{s}_r \in \mathcal{R}, S \in \{0,1\}^\ell$ such that $\mathbf{s} = \mathbf{xh} + \mathbf{y}$ and $\mathbf{s}_r = \mathbf{hr}_2 + \mathbf{r}_1$ for some $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2 \in \mathcal{S}_w^n$
Task: output 1 if $\textsc{hw}(S \oplus \llcorner S_A \lrcorner) \le t$ and $\textsc{hw}(S \oplus \llcorner S_B \lrcorner) \le t$, where $S_A = \mathbf{s}_r\mathbf{x}$ and $S_B = \mathbf{sr}_2$; and 0 otherwise.

SIDH [38]. The supersingular isogeny Diffie-Hellman (SIDH) is the only noise-free key agreement protocol on this list, and as such achieves identical views on the session key for both parties. The protocol relies on the commutativity of random walks in an isogeny graph of supersingular elliptic curves. We use the following standard notation, denoting elliptic curves by E; k-order torsion subgroups by $E[k]$; isogenies by ψ, ϕ; base points by P, Q; j-invariant by $j(\cdot)$. Generally speaking, $\ell_A = 2$ and $\ell_B = 3$ and the exponents e_A and e_B are large, say on the order of several hundreds. $P_A, Q_A \in E[\ell_A^{e_A}]$ are elements of the $\ell_A^{e_A}$-order torsion subgroup of E, and vice versa for B. The protocol's functionalities and session key can be summarized as follows.

Init: select $E_0 \xleftarrow{\$} \mathbb{E}(\mathbb{F}_q)$; $P_A, Q_A \xleftarrow{\$} E_0[\ell_A^{e_A}]$; $P_B, Q_B \xleftarrow{\$} E_0[\ell_B^{e_B}]$

AContr: sample $m_A, n_A \xleftarrow{\$} \mathbb{Z}/{\ell_A}^{e_A}\mathbb{Z}$; compute $R_A = m_A P_A + n_A Q_A$;
find $\phi : \mathbb{E}(\mathbb{F}_q) \to \mathbb{E}(\mathbb{F}_q)$ such that $\mathsf{ker}\,\phi = \langle R_A \rangle$;
transmit $E_A = \phi(E_0), \phi(P_B), \phi(Q_B)$

BContr: sample $m_B, n_B \xleftarrow{\$} \mathbb{Z}/{\ell_B}^{e_B}\mathbb{Z}$; compute $R_B = m_B P_B + n_B Q_B$;
find $\psi : \mathbb{E}(\mathbb{F}_q) \to \mathbb{E}(\mathbb{F}_q)$ such that $\mathsf{ker}\,\psi = \langle R_B \rangle$;
transmit $E_B = \psi(E_0), \psi(P_A), \psi(Q_A)$

AConv: compute $R'_A = n_A \psi(P_A) + m_A \psi(Q_A) \in E_B$;
find $\phi' : \mathbb{E}(\mathbb{F}_q) \to \mathbb{E}(\mathbb{F}_q)$ such that $\mathsf{ker}\,\phi' = \langle R'_A \rangle$;
compute $E_{BA} = \phi'(E_B)$

BConv: compute $R'_B = n_B \phi(P_B) + m_B \phi(Q_B) \in E_A$;
find $\psi' : \mathbb{E}(\mathbb{F}_q) \to \mathbb{E}(\mathbb{F}_q)$ such that $\mathsf{ker}\,\psi' = \langle R'_B \rangle$;
compute $E_{AB} = \psi'(E_A)$

S_A: $\llcorner j(E_{BA}) \lrcorner$

S_B: $\llcorner j(E_{AB}) \lrcorner$

The original SIDH paper already explicitly considers the hard problems associated with the protocol. They are called the Computational Supersingular Isogeny (CSSI) problem for ASR or BSR; Supersingular Computational Diffie-Hellman (SSCDH) problem for NKS; and Supersingular Decisional Diffie-Hellman (SSDDH) problem for NKD. We adopt this nomenclature.

Computational Supersingular Isogeny Problem (CSSI).
Input: $E_0, E_A = \phi(E_0); P_B, Q_B, P_A, Q_A \in E_0; \phi(P_B), \phi(Q_B) \in E_A$ for some isogeny $\phi : E_0 \to E_A$ with $\mathsf{ker}\,\phi = \langle n_A P_A + m_A Q_A \rangle$
Task: find a generator for $\langle R \rangle = \langle n_A P_A + m_A P_A \rangle = \mathsf{ker}\,\phi$

Supersingular Isogeny Computational Diffie-Hellman (SSCDH) Problem.
Input: $E_0, E_A = \phi(E_0), E_B = \psi(E_0)$; $P_A, Q_A P_B, Q_B \in E_0$; $\phi(P_B), \phi(Q_B) \in E_A; \psi(P_A), \psi(Q_A) \in E_B$ for isogenies $\phi, \psi : E_0 \to E_A$ with $\mathsf{ker}\,\phi = \langle n_A P_A + m_A Q_A \rangle$ and $\mathsf{ker}\,\psi = \langle n_B P_B + m_B Q_B \rangle$
Task: find $j(E_{AB})$ where $E_{AB} \cong E_0/\langle n_A P_A + m_A Q_A + n_B P_B + m_B Q_B \rangle$.

Supersingular Isogeny Decisional Diffie-Hellman (SSDDH) Problem.
Input: $E_0, E_A = \phi(E_0), E_B = \psi(E_0)$; $P_A, Q_A P_B, Q_B \in E_0$; $\phi(P_B), \phi(Q_B) \in E_A; \psi(P_A), \psi(Q_A) \in E_B; j \in \mathbb{F}_q$ for isogenies $\phi, \psi : E_0 \to E_A$ with $\mathsf{ker}\,\phi = \langle n_A P_A + m_A Q_A \rangle$ and $\mathsf{ker}\,\psi = \langle n_B P_B + m_B Q_B \rangle$
Task: output 1 if $j = j(E_0/\langle n_A P_A + m_A Q_A + n_B P_B + m_B Q_B \rangle)$; and 0 otherwise.

B Noisy Key Security

In the previous we have defined, with some justification, the security of NKA protocols in terms of the NKD game. Here we extend this justification by considering the most general possible security definition, *i.e.*, an adaptation of the Canetti-Krawczyk session key security (SK-security) notion in the authenticated links model [20], which we call *noisy key security (NK-security)*. It turns out that NK-security is equivalent to the hardness of NKD, up to a polynomial factor.

B.1 NK-Security

Adapting SK-security to the noisy case presents two difficulties.

First, Alice and Bob do not agree on the same key but on two different views S_A and S_B which are close under the Hamming metric. The adversary is deemed successful if he can distinguish between a uniformly random key and one drawn at random from the intersection of radius-t spheres centered at S_A and S_B. This extension captures the special case of noise-free key agreement of the Canetti-Krawczyk model, in which this intersection collapses to a single point $S_A = S_B$.

Second, there is a small but nonzero probability of failure even when the adversary does not interfere and it is conceivable that approximating either Alice's view or Bob's view of the session key is easier in this case. To deal with this issue, the security game aborts when the adversary picks a failing game. This choice is the same for the NKD game.

Like Canetti-Krawczyk's definition, ours considers an adversary A and any number of parties P_i each pair of which can run any number of *sessions*. The adversary can

- see, block, resend all messages passed between parties (but not modify them);
- schedule events, *i.e.*, instruct parties to start sessions or proceed with the next step;
- expire sessions, *i.e.*, instruct parties to forget the agreed-upon session key or associated state;
- *expose* sessions, either though
  - *session-state reveal*, which reveals a party's session state; or
  - *session-key query*, which reveals one party's view of the session key; or
  - *corruption*, in which case the adversary learns the entire working memory of a targeted party whose subsequent actions are all directed by the adversary.

The adversary chooses among all the unexposed sessions one *test session*, and if this test session is unsuccessful ($\textsc{hw}(S_A \oplus S_B) > t$) the game aborts. Otherwise the adversary receives a string S which is, depending on a coin flip $b \xleftarrow{\$} \{0, 1\}$, either either drawn from the intersection of radius-t spheres centered at S_A and S_B, or uniformly at random from the set of all bit strings of the same length. The adversary outputs a bit $\hat{b}$ guessing at the distribution from which S was drawn; he wins if he guesses correctly. The protocol is *noisy key secure* in the authenticated links model if no polynomial-time quantum adversary has more than a negligible distinguishing advantage. This notion is captured in words by Definition 4. Pseudocode for the oracles' behavior and the game mechanics is given in Appendix B.2.

Definition 4 (noisy key security). *Let* $\varPi = (\mathsf{Init}, \mathsf{AContr}, \mathsf{BContr}, \mathsf{AConv}, \mathsf{BConv})$ *be a noisy key agreement protocol between parties* P_A *and* P_B*, with correctness error* ϵ*. The game* NK *defines an adversary* $\mathsf{A}^{\cdots} = (\mathsf{A}_1^{\cdots}, \mathsf{A}_2^{\cdots})$ *with oracle access to the following functions:*

- $\mathsf{start}(P_A, P_B)$ *instructs parties* P_A *and* P_B *to start a new session with a fresh session id;*
- $\mathsf{deliver}(\mathit{receiver}, \mathit{sender}, \mathit{session\_id}, \mathit{contribution})$ *delivers the unaltered contribution message from receiver to sender if both are involved in session id session_id;*
- $\mathsf{contribute}(\mathit{party}, \mathit{session\_id})$ *instructs participant party to generate a contribution message for session session_id;*
- $\mathsf{converge}(\mathit{party}, \mathit{session\_id}, \mathit{contribution})$ *instructs participant party to converge, and thus obtain their view of the shared noisy session key;*
- $\mathsf{expire}(\mathit{party}, \mathit{session\_id})$ *instructs participant party to consider session session_id expired, that is to say inactive for all intents and purposes;*
- $\mathsf{reveal\_state}(\mathit{party}, \mathit{session})$ *reveals the secret state of participant party for session session_id, but as a result the session becomes exposed;*
- $\mathsf{query\_key}(\mathit{party}, \mathit{session\_id})$ *reveals party's view of the shared noisy session key from session session_id, but as a result the session becomes exposed;*
- $\mathsf{corrupt}(\mathit{party}, \mathit{code})$ *instructs participant party to execute code with access to party's state and with capability to send authentic-looking messages on behalf of party, but as a result all of party's sessions become exposed.*

The NK *game proceeds in two phases: in phase 1, the adversary* $\mathsf{A}_1^{\cdots}$ *runs with access to all the above oracles for a polynomially bounded number of time steps and as a result outputs a secret state and a test session test_session_id. If session test_session_id fails* ($\textsc{hw}(S_A \oplus S_B) > t$) *or if it has been exposed (through an invocation of* $\mathsf{reveal\_state}$*,* $\mathsf{query\_key}$*, or* $\mathsf{corrupt}$*) then the game aborts and outputs* $\perp$*. Phase 2 starts when the challenger flips a coin* b *and if* $b = 0$ *he sets* $S \xleftarrow{\$} \{0,1\}^\ell$ *but if* $b = 1$ *then* $S \xleftarrow{\$} \{x \in \{0,1\}^\ell \,|\, \textsc{hw}(x \oplus S_A) \leq t \wedge \textsc{hw}(x \oplus S_B) \leq t\}$ *where* S_A *and* S_B *are the views of the shared noisy session key of parties* P_A *and* P_B *associated with session test_session_id. Then* $\mathsf{A}_2^{\cdots}$ *is run with access to all oracles on input* (state, S) *for another polynomially bounded number of steps, after which he outputs a guess* $\hat{b}$*. The game outputs 1 if* $b = \hat{b}$ *and 0 otherwise.*

Then the noisy key agreement protocol is noisy key secure *(NK-secure) in the* authenticated links model *if for all polynomial time quantum adversaries* $\mathsf{A}^{\cdots}$ *who starts* k *sessions and corrupts* r *of them, their advantage* $\mathsf{Adv}_{\varPi}^{\mathsf{NK}}(\mathsf{A}^{\cdots})$ *is negligible:*

$$\mathsf{Adv}_{\varPi}^{\mathsf{NK}}(\mathsf{A}^{\cdots}) \triangleq \left| \Pr[\mathsf{Game}_{\mathsf{NK}}^{\mathsf{A}}(1^\lambda) \not\Rightarrow 0] - \frac{1 + \epsilon\frac{k-r}{k} + \frac{r}{k}}{2} \right| \leq \mathsf{negl}(\lambda) \ . \tag{64}$$

This expression for the adversary's advantage is rather complex but certainly valid. The adversary who corrupts every session he starts in order to engineer game abortions, has advantage zero. The same is true for the adversary who does not corrupt any session but flips a coin and guesses accordingly. The adversary's advantage remains zero for any combination of these two extremes. Therefore, the expression captures the adversary's advantage over a naïve strategy.

We stress that a NKA protocol must consist of two independent messages, one in each direction, as formalized in the syntax. While the Canetti-Krawczyk security model does not impose any bounds on the number of messages exchanged or their scheduling, in the case of NKA this restriction on the number of passes is critical; if the parties involved are allowed more then they can agree on an exact key simply by transmitting auxiliary information to correct errors.

Game 9: $\mathsf{NK}^{\mathsf{A}^{\cdots}}(1^{\lambda})$

```
$\mathit{party\_states} \leftarrow [\varnothing$ for all parties$]$
$\mathit{authentic\_messages}, \mathit{global\_sessions} \leftarrow \mathsf{empty\_lists}$
$\mathit{session\_counter} \leftarrow 0$
$\mathit{test\_session\_id}, \mathit{state} \leftarrow \mathsf{A}_1^{\cdots}(1^{\kappa})$
if $\mathit{global\_sessions}[\mathit{test\_session\_id}].\mathit{exposed} = \mathsf{True}$ then:
    return $\perp$
$P_A \leftarrow \mathit{global\_sessions}[\mathit{test\_session\_id}].A$
$P_B \leftarrow \mathit{global\_sessions}[\mathit{test\_session\_id}].B$
$S_A \leftarrow \mathit{party\_states}[P_A].\mathit{sessions}[\mathit{test\_session\_id}].S$
$S_B \leftarrow \mathit{party\_states}[P_B].\mathit{sessions}[\mathit{test\_session\_id}].S$
if $\textsc{hw}(S_A \oplus S_B) > t$ then:
    return $\perp$
$b \xleftarrow{\$} \{0,1\}$
if $b = 1$ then:
    $S \xleftarrow{\$} \{x \in \{0,1\}^{\ell} \mid \textsc{hw}(x \oplus S_A) \wedge \textsc{hw}(x \oplus S_B)\}$
else:
    $S \xleftarrow{\$} \{0,1\}^{\ell}$
$\hat{b} \leftarrow \mathsf{A}_2^{\cdots}(\mathit{state}, S)$
return $[\![b = \hat{b}]\!]$
```

B.2 Pseudocode for Oracle Behavior

Oracle 10: $\mathsf{deliver}(\cdot)$

```
define deliver($receiver$, $sender$, $session\_id$, $contribution$) as:
    if ($receiver$, $sender$, $session\_id$, $contribution$) $\notin$ $authentic\_messages$ then:
        return $\perp$
    if $session\_id \notin party\_states[receiver].sessions.\mathsf{keys}()$ then:
        return $\perp$
    if $session\_id \notin party\_states[sender].sessions.\mathsf{keys}()$ then:
        return $\perp$
    $party\_states[receiver].sessions[session\_id].contribution = contribution$
```

Oracle 11: $\mathsf{start}(\cdot)$

```
define start($P_A, P_B$) as:
    $global\_sessions.append$(global_session(
        $A = P_A$,
        $B = P_B$,
        $exposed$ = False))
    $iparams \leftarrow \Pi.\mathsf{Init}(1^\kappa)$
    $party\_states[P_A].sessions$.append(session(
        $key = session\_counter$,
        $A = P_A$,
        $B = P_B$,
        $params = iparams$,
        $state = \varnothing$,
        $contribution = \varnothing$,
        $S = 0^\ell$))
    $party\_states[P_B].sessions$.append(session(
        $key = session\_counter$,
        $A = P_A$,
        $B = P_B$,
        $params = iparams$,
        $state = \varnothing$,
        $contribution = \varnothing$,
        $S = 0^\ell$))
    $session\_counter \leftarrow session\_counter + 1$
```

Oracle 12: contribute(·)

```
define contribute($party$, $session\_id$) as:
    if $session\_id \notin party\_states[party].sessions$.keys() then:
        return $\perp$
    $session \leftarrow party\_states[party].sessions[session\_id]$
    if $party = session.A$ then:
        $session.state, session.contribution \leftarrow \mathit{\Pi}$.AContr($session.params$)
    else:
        $session.state, session.contribution \leftarrow \mathit{\Pi}$.BContr($session.params$)
    $party\_states[party].sessions[session\_id] \leftarrow session$
    $msg \leftarrow$ message(
        $sender = party$,
        $receiver = \{session.A, session.B\} \backslash party$,
        $session\_id = session\_id$,
        $contribution = session.contribution$)
    $authentic\_messages$.append($msg$)
    return $msg$    ▷ allow adversary to block
```

Oracle 13: converge(·)

```
define converge($party$, $session\_id$, $contribution$) as:
    if $session\_id \notin party\_states[party].sessions$.keys() then:
        return $\perp$
    $session \leftarrow party\_states[party].sessions[session\_id]$
    $other \leftarrow \{session.A, session.B\} \backslash party$
    if $(other, party, session\_id, contribution) \notin authentic\_messages$ then:
        return $\perp$
    if $party = session.\mathrm{A}$ then:
        $session.S \leftarrow \mathit{\Pi}$.AConv($session.state, contribution$)
    else:
        $session.S \leftarrow \mathit{\Pi}$.BConv($session.state, contribution$)
    $session.state = \varnothing$
    $party\_states[party].sessions[session\_id] \leftarrow session$
```

Oracle 14: expire(·)

```
define expire($party$, $session\_id$) as:
    if $session\_id \notin party\_states[party].sessions$.keys then:
        return $\perp$
    $party\_states[party].sessions[session\_id].S = \varnothing$
```

Oracle 15: reveal_state($\cdot$)

```
define reveal_state($party$, $session\_id$) as:
    if $session\_id \notin party\_states[party].sessions.\mathsf{keys}$ then:
        return $\perp$
    $global\_sessions[session\_id].exposed \leftarrow \mathsf{True}$
    return $party\_states[party].sessions[session\_id].state$
```

Oracle 16: query_key($\cdot$)

```
define query_key($party$, $session\_id$) as:
    if $session\_id \notin party\_states[party].sessions.\mathsf{keys}$ then:
        return $\perp$
    $global\_sessions[session\_id].exposed \leftarrow \mathsf{True}$
    return $party\_states[party].sessions[session\_id].S$
```

Oracle 17: corrupt($\cdot$)

```
define corrupt($party$, $code$) as:
    for all $session \in party\_states[party].sessions$ do:
        if $session.state \neq \varnothing$ or $session.S \neq \varnothing$ then:
            $global\_sessions[session.session\_id].exposed \leftarrow \mathsf{True}$
    execute($code$) with access to:
        • $authentic\_messages.\mathsf{append}(sender = party, \cdot, \cdot, \cdot)$
        • $party\_states[party]$
```

Some explanation about the variables' purpose and usage is in order. In the following enumeration we mix descriptions of variables and their types.

- session_id, party_id : integer. These identifiers are just integers.
- *party_state* : list of dict mapping session_id to session. This variable is a list containing for each party i a dict called *sessions*, which is a dictionary mapping session_ids to session objects.
- session. This type is a tuple containing the following objects:
  - *key* : session_id. Integer uniquely identifying the session and counterpart-session pair. (In other words, the other party involved in this session has a matching session and it has the same *key*.)
  - A : party_id. This party id indicates the party who is taking on the role of A in the NKA session.
  - B : party_id. This party id indicates the party who is taking on the role of B in the NKA session.
  - *params* : ParSp. This object takes on the value *iparams* as generated by the Init function of the NKA protocol.
  - *state* : StateSp. This variable takes on the value of *A_state* or *B_state* in the NKA protocol.
  - *contribution* : ContrSp. This variable takes on the value of this party's contribution in the NKA protocol.

  - $S : \{0,1\}^\ell$. This is the view of the shared noisy key as held by the party in question.
- *global_sessions* : list of global_session. This list of global_session objects contains big picture information on superficial session attributes like the parties involved and whether or not the session has been exposed.
- global_session. This object consists of the following variables:
  - A : party_id. This variable is the party id of the party who assumes the role of A in the NKA session pair.
  - B : party_id. This variable is the party id of the party who assumes the role of B in the NKA session pair.
  - *exposed* : $\{\mathsf{True}, \mathsf{False}\}$. Boolean variable indicating whether the session has been exposed or not.
- *authentic_messages* : list of message objects, representing all information transmitted between parties.
- message. This object consists of the following variables:
  - *sender* : party_id. This variable identifies the originator of the message.
  - *receiver* : party_id. This variable identifies the intended receiver of the message.
  - *session_id* : session_id. This variable identifies the session pair to which this protocol contribution pertains.
  - *contribution* : ContrSp. The actual content of the message: an NKA protocol contribution.
- *test_session_id* : session_id. The identifier of the test session as output by the adversary at the end of the first phase.
- *state* : $\{0,1\}^*$. The adversary's state at the end of the first phase; recording this state allows the adversary to pick up where it left off.
- P_A, P_B : party_id. These identifiers determine the parties involved in the session pair that was chosen as test session by the adversary.
- $S_A, S_B : \{0,1\}^\ell$. These are the views of the noisy session key associated with the two parties in the session pair that was chosen as test session by the adversary.
- $S : \{0,1\}^\ell$. Challenge key, to be fed to the adversary in the second phase. The adversary wins if he can tell whether S was drawn from a uniform distribution or from the intersection of two radius-t spheres centered at S_A and S_B.
- $b, \hat{b} : \{0,1\}$. Bits, one determining whether to sample S at random or from the intersection of spheres; the other being the adversary's guess.

B.3 NK-security and NKD Assumption

Theorem 3. *The NKD Assumption is necessary and sufficient for NK-security.*

This theorem is an immediate corollary of the following two lemmas, both of which have straightforward proofs.

Lemma 5 (NKD $\Longrightarrow$ NK). *Let* A *be a polynomial time quantum adversary in the* NK *game with respect to an NKA protocol* $\mathit{\Pi}$ *with failure probability* ϵ*, and let* k *and* r *be the number of sessions started and corrupted, respectively, by* A*, and* $\frac{1+\epsilon\frac{k-r}{k}+\frac{r}{k}}{2}+\zeta$ *its winning probability. Then there is a polynomial time quantum algorithm* B *that wins the* NKD *game in polynomial time with probability* $\frac{1+\epsilon}{2}+\frac{\zeta}{k-r}$*.*

Proof. The arguments of B are $(\mathit{iparams}, \mathit{A\_contr}, \mathit{B\_contr}, S)$. B chooses a random session identifier $\mathit{id} \xleftarrow{\$} \{0, \dots, k-1\}$ and simulates the NK game. The oracles are defined in accordance with Definition 4 except where the session with $\mathit{session\_id} = \mathit{id}$ is concerned. For this session, the instance parameters and both parties' contributions are set to *iparams*, *A_contr*, and *B_contr*. The views of the session keys are set to the same random bitstring of length ℓ.

The adversary $\mathsf{A}_1^{\cdots}(1^\lambda)$ is run and if its output $\mathit{test\_session\_id} \neq k$ then B flips a coin $\hat{b} \xleftarrow{\$} \{0,1\}$ and returns that. Otherwise $\mathsf{A}_2^{\cdots}(\mathit{state}, S)$ is run, where S is B's fourth argument. If session *id* is exposed, B returns a random coin flip $\hat{b} \xleftarrow{\$} \{0,1\}$ and otherwise B returns the output of $\hat{b} \leftarrow \mathsf{A}_2^{\cdots}$. The exact behavior of B and the modified oracle interface it provides the simulated adversary A with, are presented in Algorithm 18 and oracle $\mathsf{contribute}'$, with the other oracles being identically defined to those in Definition 4.

The tuple (*iparams*, *A_contr*, *B_contr*) associated with each session is identically distributed, including session *id*. Therefore the probability that $\mathsf{A}_1^{\cdots}$'s output $\mathit{test\_session\_id} = \mathit{id}$ is exactly $1/k$. Let z be shorthand for the output of the NKD game, *i.e.*, $z \leftarrow \mathsf{NKD}^{\mathsf{B}^{\mathsf{A}^{\cdots}}}(1^\lambda)$.

Algorithm 18: $\mathsf{B}^{\mathsf{A}^{\cdots}}(iparams, A\_contr, B\_contr, S)$

```
$party\_states \leftarrow [\varnothing$ for all parties$]$
$authentic\_messages, global\_sessions \leftarrow$ empty_lists
$session\_counter \leftarrow 0$
$id \xleftarrow{\$} \{0, \dots, k-1\}$
$test\_session\_id, state \leftarrow \mathsf{A}_1^{\cdots}(1^\kappa)$
if $test\_session\_id \neq id$ then:
    return $\perp$
$\hat{b} \leftarrow \mathsf{A}_2^{\cdots}(state, S)$
if $global\_sessions[id].exposed =$ True then:
    $\hat{b} \xleftarrow{\$} \{0, 1\}$
return $\hat{b}$
```

Oracle 19: contribute′

```
define contribute($party$, $session\_id$) as:
    if $session\_id \notin party\_states[party].sessions.$keys() then:
        return $\perp$
    $session \leftarrow party\_states[party].sessions[session\_id]$
    if $party = session.A$ then:
        $session.state, session.contribution \leftarrow \Pi.$AContr$(session.params)$
        if $session\_id = id$ then:
            $session.contribution \leftarrow A\_contr$
    else:
        $session.state, session.contribution \leftarrow \Pi.$BContr$(session.params)$
        it $session\_id = id$ then:
            $session.contribution \leftarrow B\_contr$
    $party\_states[party].sessions[session\_id] \leftarrow session$
    $msg \leftarrow ($
        $sender = party,$
        $receiver = \{session.A, session.B\} \backslash party,$
        $session\_id = session\_id,$
        $contribution = session.contribution)$
    $authentic\_messages.$append$(msg)$
    return $msg$    ▷ allow adversary to block
```

Then we have:

$$
\begin{align}
&\Pr[\mathsf{NKD}^{\mathsf{B}^{\mathsf{A}^{\cdots}}}(1^\lambda) \not\Rightarrow 0] \triangleq \Pr[z \neq 0] \tag{65}\\
&= \Pr[z \neq 0 \,|\, z \neq \perp] \cdot \Pr[z \neq \perp] + \Pr[z \neq 0 \,|\, z = \perp] \cdot \Pr[z = \perp] \tag{66}\\
&= \epsilon + \Pr[z \neq 0 \,|\, z \neq \perp \wedge \mathit{test\_session\_id} = \mathit{id}] \cdot \Pr[\mathit{test\_session\_id} = \mathit{id}] \cdot (1-\epsilon) \nonumber\\
&\quad + \Pr[z \neq 0 \,|\, z \neq \perp \wedge \mathit{test\_session\_id} \neq \mathit{id}] \cdot \Pr[\mathit{test\_session\_id} \neq \mathit{id}] \cdot (1-\epsilon) \tag{67}\\
&= \Pr[\mathsf{NK}^{\mathsf{A}^{\cdots}}(1^\lambda) \not\Rightarrow 0 \,|\, \not\perp] \cdot \frac{1}{k} \cdot (1-\epsilon) + \epsilon + \frac{1}{2} \cdot \frac{k-1}{k} \cdot (1-\epsilon) \tag{68}\\
&= \Big(\Pr[\mathsf{NK}^{\mathsf{A}^{\cdots}}(1^\lambda) \not\Rightarrow 0 \,|\, \not\perp] \cdot \Pr[\not\perp] \nonumber\\
&\qquad + \Pr[\mathsf{NK}^{\mathsf{A}^{\cdots}}(1^\lambda) \not\Rightarrow 0 \,|\, \perp] \cdot \Pr[\perp] \nonumber\\
&\qquad - \Pr[\mathsf{NK}^{\mathsf{A}^{\cdots}}(1^\lambda) \not\Rightarrow 0 \,|\, \perp] \cdot \Pr[\perp]\Big) \cdot (\Pr[\not\perp])^{-1} \cdot \frac{1}{k} \cdot (1-\epsilon) \nonumber\\
&\quad + \epsilon + \frac{1}{2} \cdot \frac{k-1}{k} \cdot (1-\epsilon) \tag{69}\\
&= \Pr[\mathsf{NK}^{\mathsf{A}^{\cdots}}(1^\lambda) \not\Rightarrow 0] \cdot (\Pr[\not\perp])^{-1} \cdot \frac{1}{k} \cdot (1-\epsilon) \nonumber\\
&\quad - \left(\frac{r}{k} + \frac{k-r}{k}\epsilon\right) \cdot (\Pr[\not\perp])^{-1} \cdot \frac{1}{k} \cdot (1-\epsilon) \nonumber\\
&\quad + \epsilon + \frac{1}{2} \cdot \frac{k-1}{k} \cdot (1-\epsilon) \tag{70}\\
&= \Pr[\mathsf{NK}^{\mathsf{A}^{\cdots}}(1^\lambda) \not\Rightarrow 0] \cdot \left(\frac{k}{k-r-\epsilon k+\epsilon r}\right) \cdot \frac{1}{k} \cdot (1-\epsilon) \nonumber\\
&\quad - \left(\frac{r}{k} + \frac{k-r}{k}\epsilon\right) \cdot \left(\frac{k}{k-r-\epsilon k+\epsilon r}\right) \cdot \frac{1}{k} \cdot (1-\epsilon) \nonumber\\
&\quad + \epsilon + \frac{1}{2} \cdot \frac{k-1}{k} \cdot (1-\epsilon) \tag{71}\\
&= \frac{1}{k-r}\Pr[\mathsf{NK}^{\mathsf{A}^{\cdots}}(1^\lambda) \not\Rightarrow 0] - \frac{\epsilon}{k} - \frac{r}{k(k-r)} + \epsilon + \frac{1-\epsilon}{2} \cdot \frac{k-1}{k} \tag{72}\\
&= \frac{1}{k-r}\left(\frac{1 + \frac{k-r}{k}\epsilon + \frac{r}{k}}{2} + \zeta\right) - \frac{\epsilon}{k} - \frac{r}{k(k-r)} + \epsilon + \frac{1-\epsilon}{2} \cdot \frac{k-1}{k} \tag{73}\\
&= \frac{\zeta}{k-r} + \frac{\frac{1}{2}}{k-r} + \frac{\epsilon}{2k} + \frac{\frac{r}{2}}{k(k-r)} - \frac{\epsilon}{k} - \frac{r}{k(k-r)} + \epsilon + \frac{1-\epsilon}{2} \cdot \frac{k-1}{k} \tag{74}\\
&= \frac{1+\epsilon}{2} + \frac{\zeta}{k-r} \tag{75}
\end{align}
$$

□

Lemma 6 (NK $\implies$ NKD). *Let* A *be a polynomial time quantum adversary in the* NKD *game with respect to an NKA protocol* $\mathit{\Pi}$ *with failure probability* ϵ*, whose winning probability is* $\frac{1+\epsilon}{2} + \zeta$*. Then there is a polynomial time quantum* *algorithm* $\mathsf{B}^{\cdots}$ *that wins the* NK *game with respect to* Π *in polynomial time with probability* $\frac{1+\epsilon}{2} + \zeta$.

Proof. The adversary $\mathsf{B}^{\cdots} = (\mathsf{B}_1^{\cdots}, \mathsf{B}_2^{\cdots})$ behaves as follows. In phase 1, $\mathsf{B}_1^{\cdots}$ starts a session between two random parties and instructs both of them to contribute and converge; he thus obtains $\mathit{session\_id}, \mathit{iparams}, \mathit{A\_contr}, \mathit{B\_contr}$. His output is then $(\mathit{test\_session\_id} = \mathit{session\_id}, \mathit{state} = (\mathit{iparams}, \mathit{A\_contr}, \mathit{B\_contr}))$.

In phase 2, $\mathsf{B}_2^{\cdots}$ runs on input $(\mathit{state} = (\mathit{iparams}, \mathit{A\_contr}, \mathit{B\_contr}), S)$. He invokes A as an NKD-oracle, namely by passing it the arguments ($\mathit{iparams}$, $\mathit{A\_contr}$, $\mathit{B\_contr}$, S) and obtaining A's guess $\hat{b}$, which is also $\mathsf{B}_2^{\cdots}$'s output. Whenever A wins, so does $\mathsf{B}^{\cdots}$, so the theorem follows. □

The reduction NKD $\Longrightarrow$ NK loses a security factor $1/(k-r)$, where k is the number of sessions started by the NK adversary and r is the number of sessions corrupted. However, this security loss is a necessary consequence of restricting the number of available sessions to one, as in the NKD game. NK-security and the NKD Assumption remain asymptotically equivalent.

Chapter 8

Standardization Proposals

8.1 Ramstake

Publication data

Alan Szepieniec, "Ramstake" *Submitted to NIST PQC project [75].*

Contributions

Principal submitter.

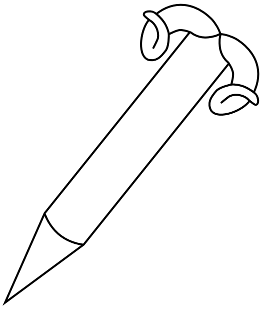
Notes

This cryptosystem was inspired by the NTRU-like cryptosystem by Aggarwal *et al.* [4]. The replacement of the NTRU-like construction with a noisy Diffie-Hellman protocol makes for a simpler cryptosystem. It turns out that they independently came up with essentially the same construction in their own NIST submission "Mersenne-756839" and subsequent ePrint paper [5]. For reference, the original paper was uploaded on the 30th of May 2017, the NIST deadline was 30 November 2017, and the updated paper was 6th of December.

Ramstake

KEM Proposal for NIST PQC Project

September 7, 2018



cryptosystem name	Ramstake
principal submitter	Alan Szepieniec imec-COSIC KU Leuven alan.szepieniec@esat.kuleuven.be tel. +3216321953 Kasteelpark Arenberg 10 bus 2452 3001 Heverlee Belgium
auxiliary submitters	-
inventors / developers	same as principal submitter; relevant prior work is credited as appropriate
owner	same as principal submitter
backup contact info	alan.szepieniec@gmail.com
signature	

Contents

1 Introduction

The long-term security of confidential communication channels relies on their capacity to resist attacks by quantum computers. To this end, NIST envisions a transition away from public key cryptosystems that are known to fail in this scenario, and towards the so-called *post-quantum* cryptosystems. One of the functionalities in need of a post-quantum solution that is essential for securing online communication is *ephemeral key exchange*. This protocol enables two parties to agree on a shared secret key at a cost so insignificant as to allow immediate erasure of all secret key material after execution, as an additional security measure. In the case where the order of the messages need not be interchangeable, this functionality is beautifully captured by the *key encapsulation mechanism* (KEM) formalism of Cramer and Shoup [6]. The same formalism has the added benefit of capturing the syntax and security of the first part of IND-CCA-secure arbitrary-length hybrid encryption schemes, enabling a separation of the public key layer from the symmetric key layer.

The desirable properties of a post-quantum KEM are obvious upon consideration. It should be fast and it should generate short messages, not require too much memory and be implementable on a small area or in a few lines of code. It should inspire confidence by relying on long-standing hard problems or possibly even advertising a proof of security. However, this design document is predicated on the greater importance of a property not included in the previous enumeration: *simplicity.* The requirement for advanced degrees in mathematics on the part of the implementers presents a giant obstacle to mass adoption, whereas no such obstacle exists for mathematically straightforward schemes. More importantly, complexity has the potential to hide flaws and insecurities as they can only be exposed by experts in the field. In contrast, a public key scheme that is accessible to a larger audience is open to scrutiny from that same larger audience, and should therefore engender a greater confidence than a scheme that only a few experts were not able to break.

This document presents Ramstake, a post-quantum key encapsulation mechanism that excels in this category of simplicity. Aside from the well-established tools of hash functions, pseudorandom number generators, and error-correcting codes, Ramstake requires only high school mathematics. Though not optimized for message size and speed, Ramstake is still competitive in these categories with messages of less than one hundred kilobytes generated in a handful of milliseconds on a regular desktop computer at the highest security level. For security, Ramstake relies on a relatively new and under-studied hard problem, which requires several years of attention attention from the larger cryptographic community before it inspires confidence. The flipside of this drawback is the advantage associated with problem diversity: Ramstake is likely to remain immune to attacks that affect other branches of post-quantum cryptography.

Innovation. In a nutshell, this hard problem requires finding *sparse* solutions to linear equations modulo a large Mersenne prime, *i.e.* a prime of the form $p = 2^\pi - 1$. The binary expansions of the solution (x_1, x_2) consist overwhelmingly of zeros. Specifically, these integers can be described as

$$x_i = \sum_{j=1}^{w} 2^{e_j} \quad . \tag{1}$$

We refer to the integer's *Hamming weight* w as the number of ones; their positions e_j are generally chosen uniformly at random from $\{0, \dots \pi - 1\}$. Ramstake's analogue of the discrete logarithm problem requires finding x_1 and x_2 of this form from G and $H = x_1 G + x_2 \operatorname{mod} p$. This is an affine variant of the Low Hamming Weight Ratio problem of the Aggarwal *et al.* Mersenne prime cryptosystem [1], whose task is to obtain f and g of this form (1) from $H = fg^{-1} \operatorname{mod} p$.

Where the Aggarwal *et al* cryptosystem builds on the indistinguishability of low Hamming weight ratios, Ramstake builds on a noisy Diffie-Hellman protocol [2, 3] instead. Alice and Bob agree on a random integer G between 0 and p. Alice chooses sparse integers x_1 and x_2 and sends $H = x_1G + x_2 \bmod p$ to Bob. Bob chooses sparse integers y_1 and y_2 and sends $F = y_1G + y_2 \bmod p$ to Alice. Alice computes $S_a = x_1F \bmod p$ and Bob computes $S_b = y_1G \bmod p$ and both integers approximate $S = x_1y_1G \bmod p$ in the following sense: since p is a Mersenne prime, reduction modulo p does not increase the integer's Hamming weight and as a result the differences $S_a - S = x_1y_2 \bmod p$ and $S_b - S = y_1x_2 \bmod p$ have a sparse binary expansion. Therefore, if x_1, x_2, y_1, y_2 have a sufficiently low Hamming weight, the binary expansions of S_a and S_b agree in most places. Alice and Bob have thus established a shared noisy secret stream of data, or since it will be used as a one-time pad, a *shared noisy one-time pad* (SNOTP, "snow-tipi").

From SNOTP to KEM. There are various constructions in the literature for obtaining KEMs from SNOTPs, each different in its own subtle way. The next couple of paragraphs give a high-level description of a generic transformation targeting IND-CCA security, which is inspired by the "encryption-based approach" of NewHope-Simple [4]. This construction makes abstraction of the underlying sparse integer mathematics.

The encapsulation algorithm is a deterministic algorithm taking a fixed-length random seed s as an explicit argument. If more randomness is needed than is contained in this seed, it is generated from a cryptographically secure pseudorandom number generator (CSPRNG). The algorithm outputs a ciphertext c and a symmetric key k.

The encapsulation algorithm uses an error-correcting code such as Reed-Solomon or BCH to encode the seed s into a larger bitstring. Then the ciphertext c consists of three parts: 1) a contribution to the noisy Diffie-Hellman protocol; 2) the encoding of the seed but one-time-padded with the encapsulator's view of the SNOTP; and 3) the hash of the seed. The decapsulation algorithm computes its own view of the SNOTP using the Diffie-Hellman contribution and undoes the one-time pad to obtain the encoding up to some errors. Under certain conditions, the error-correcting code is capable of retrieving the original seed s from this noisy codeword. At this point, the decapsulation algorithm runs the encapsulation algorithm with the exact same arguments, thus guaranteeing that the produced symmetric key k is the same for both parties. Robust IND-CCA security comes from the fact that the decapsulator can compare bit by bit the received ciphertext against the one that was recreated from the transmitted seed, in addition to verifying the seed's hash against the one that was part of the ciphertext.

2 Specification

2.1 Parameters

The generic description of the scheme refers the following parameters without reference to their value. Concrete values are given in Section 2.4.

- p — the Mersenne prime modulus, satisfies $p = 2^\pi - 1$;
- π — the number of bits in the binary expansion of p;

- w — the Hamming weight, which determine the number of ones in the binary expansion of secret sparse integers;
- ν — the number of codewords to encode the transmitted seed into;
- n — the length of a single codeword (in number of bytes);
- κ — the targeted security level (in log_2 of classical operations);
- λ — the length of seed values (in number of bits);
- χ — the length of the symmetric key (in number of bits).

2.2 Tools

2.2.1 Error-Correcting Codes

Ramstake relies on Reed-Solomon codes over $\mathrm{GF}(2^8)$ with designed distance $\delta = 224$ and dimension $k = 32$. Codewords are $n = 255$ field elements long and if there are 111 or fewer errors they can be corrected. With this choice of finite field, one field element coincides with one byte. The following subroutines are used abstractly:

- encode takes a string of $8k = 256$ bits and outputs a sequence of $8n$ bits that represents the Reed-Solomon encoding of the input.

- decode takes a string of $8n$ bits representing a noisy codeword and tries to decode it. If the codeword is decodable, this routine returns the error symbol $\perp$.

This abstract interface suffices for the description of the KEM. Moreover, any concrete instantiation can be exchanged for any other instantiation that adheres to the same interface, or that modifies the interface slightly to retain compatibility.

2.2.2 CSPRNG

Both key generation and encapsulation require a seed expander. All randomness can be generated up front; there is no need to record state and update it as pseudorandomness is generated. We use $\mathsf{xof}(s, \ell)$ to denote the invocation of the CSPRNG to generate a string of ℓ pseudorandom bytes from the seed s.

This abstract interface suffices for the description of the KEM. In the implementations, xof is instantiated with `SHAKE256`. Like in the case of the Reed-Solomon codec, any concrete instantiation can be exchanged for any other instantiation that adheres to the same interface.

2.3 Description

2.3.1 Serialization of Integers

All big integers represent elements in $\{0, \ldots, p-1\}$ and are therefore fully defined by π bits. Denote by $\mathsf{serialize}(a)$ the array of $\lceil \frac{\pi}{8} \rceil$ bytes satisfying

$$a = \sum_{i=0}^{\lceil \frac{\pi}{8} \rceil - 1} \mathsf{serialize}(a)[i] \times 256^i \ . \tag{2}$$

This serialization puts the least significant byte first and pads the array with zeros to meet the given length if the integer is not large enough. It is essentially Little-Endian padded to length $\lceil \frac{\pi}{8} \rceil$, and corresponds with the GMP function `mpz_export`($\cdot$, NULL, -1, 1, 1, 0, a) regardless of whether the integer a is large enough.

2.3.2 Data Structures

Ramstake uses five data structures: a random seed, a secret key, a public key, a ciphertext, and a symmetric key. Random seeds are bitstrings of length λ, whereas symmetric keys are bitstrings of length χ. The other three data structures are more involved.

Secret key. A secret key consists of the following items:

- `seed` — a random seed which fully determines the rest of the secret key in addition to the public key;
- a, b — sparse integers, represented by π bits each.

Public key. A public key consists of the following items:

- `g_seed` – a random seed which is used to generate the random integer G;
- C — integer between 0 and p which represents a noisy Diffie-Hellman contribution. This value satisfies $C = aG + b \,\mathsf{mod}\, p$.

Ciphertext. A ciphertext consists of the following items:

- D — integer between 0 and p which represents a noisy Diffie-Hellman contribution; this value satisfies $D = a'G + b' \,\mathsf{mod}\, p$ where a', b' are secret sparse integers sampled by the encapsulator;
- `seedenc` — string of $8n\nu$ bits; this value is the bitwise xor of the binary expansion of the first $n\nu$ bytes of $\mathsf{serialize}(S)$ and the sequence of ν times $\mathsf{encode}(s)$, where s is the random seed that is the argument to the encapsulation algorithm, and where S is the encapsulator's view of the SNOTP: $S = a'(aG + b) \,\mathsf{mod}\, p$.
- h — hash of the seed s; the purpose of this value is twofold: 1) to speed up decapsulation by enabling the decoder to recognize correct decodings, and 2) to anticipate a proof technique in which the simulator answers decapsulation queries by finding this value's inverse.

These objects are serialized by appending the serializations of their member items in the order presented above. No length information is necessary as the size of each object is a function of the parameters. We overload *serialize* to denote that operation.

In this notation, the symmetric key $k \in \{0, 1\}^\chi$ satisfies $k = \mathsf{H}(\mathsf{serialize}(pk) \| coins)$, where *pk* is the public key and where *coins* is the byte string of random coins used by the encapsulator. Ramstake instantiates H with SHA3-256 with output truncated to χ bits, but any other secure hash function suffices.

2.3.3 Algorithms

A KEM consists of three algorithms, KeyGen, Encaps, and Decaps. Pseudocode for Ramstake's three algorithms is presented in Algorithms 3, 4, and 5. All three functionalities obtain a pseudorandom integer G from a short seed; this subprocedure is called generate_g and is shown in Algorithm 1. Algorithms KeyGen and Encaps rely on a common subroutine called sample_sparse_integer which deterministically samples a sparse integer given enough random bytes and a target Hamming weight, and which is described in Algorithm 2.

```
algorithm generate_g
input: seed $\in \{0,1\}^\lambda$ — random seed
output: $g \in \{0, \ldots, p-1\}$ — pseudorandom integer

$\mathtt{r} \leftarrow \mathsf{xof}(\mathtt{seed}, \lfloor \frac{\pi}{8} \rfloor + 2)$
$g \leftarrow 0$
for $i$ from $0$ to $\lfloor \frac{\pi}{8} \rfloor + 1\}$ do:
    $g \leftarrow 256 \times g + \mathtt{r}[i]$
end
return $g \bmod p$
```

Algorithm 1: Procedure to sample a random integer from $\{0, \ldots, p-1\}$.

```
algorithm sample_sparse_integer
input: $\mathtt{r} \in \{0, \ldots, 255\}^{4 \times \mathtt{weight}}$ — enough random bytes
       $\mathtt{weight} \in \{0, \ldots, \pi\}$ — number of one bits
output: $a \in \{0, \ldots, p-1\}$ — a sparse integer

$a \leftarrow 0$
for $i$ from $0$ to $\mathtt{weight} - 1$ do:
    $u \leftarrow (\mathtt{r}[4i] \times 256^3 + \mathtt{r}[4i+3] \times 256^2 + \mathtt{r}[4i+2] \times 256 + \mathtt{r}[4i+1]) \bmod \pi$
    $a \leftarrow a + 2^u$
end
return $a$
```

Algorithm 2: Procedure to sample a sparse integer from a CSPRNG.

```
**algorithm** KeyGen
**input**: $\texttt{seed} \in \{0,1\}^\lambda$ — random seed
**output**: $sk$ — secret key
            $pk$ – public key

    ▷ expand randomness
    $\texttt{r} \leftarrow \mathsf{xof}(\texttt{seed}, 4 \times w + 4 \times w + \lambda/8)$
    ▷ grab seed for $G$ and generate $G$
    $\texttt{seed\_g} \leftarrow \texttt{r}[0 : (\lambda/8)]$
    $G \leftarrow \mathsf{generate\_g}(\texttt{seed\_g})$
    ▷ get sparse integers $a$ and $b$
    $a \leftarrow \mathsf{sample\_sparse\_integer}(\texttt{r}[(\lambda/8) : (\lambda/8 + 4 \times w)], w)$
    $b \leftarrow \mathsf{sample\_sparse\_integer}(\texttt{r}[(\lambda/8 + 4 \times w) : (\lambda/8 + 4 \times w + 4 \times w)], w)$
    ▷ compute Diffie-Hellman contribution
    $C \leftarrow aG + b \,\mathsf{mod}\, p$
    **return** $sk = (s, a, b)$, $pk = (\texttt{g\_seed}, C)$
```

Algorithm 3: Generate a secret and public key pair.

```
algorithm Encaps
input: seed $\in \{0,1\}^{\lambda}$ — random seed
       $pk$ — public key
output: $ctxt$ — ciphertext
        $k \in \{0,1\}^{\chi}$ – symmetric key

    ▷ extract randomness and generate $G$ from seed
    r $\leftarrow$ xof(seed, $4 \times w + 4 \times w$)
    $G \leftarrow$ generate_g($pk$.seed_g)
    ▷ sample sparse integers
    $a' \leftarrow$ sample_sparse_integer(r$[0 : (4 \times w)], w$)
    $b' \leftarrow$ sample_sparse_integer(r$[(4 \times w) : (4 \times w + 4 \times w)], w$)
    ▷ compute Diffie-Hellman contribution and SNOTP
    $D \leftarrow a'G + b' \bmod p$
    $S \leftarrow a'\, pk.C \bmod p$
    ▷ encode random seed and apply SNOTP
    seedenc $\leftarrow$ serialize($S$)$[0 : (n\nu)]$
    for $i$ from 0 to $\nu - 1$ do:
        seedenc$[(in) : ((i+1)n)] \leftarrow$ seedenc$[(in) : ((i+1)n)] \oplus$ encode(seed)
    end
    ▷ compute symmetric key
    $k \leftarrow$ H(serialize($pk$)$\|$r)
    ▷ complete ciphertext; and return ciphertext and symmetric key
    $h \leftarrow$ H(seed)
    return $ctxt = (D,$ seedenc$, h), k$
```

Algorithm 4: Encapsulate: generate a ciphertext and a symmetric key.

```
algorithm Decaps
input: $ctxt = (D, \texttt{seedenc}, h)$ — ciphertext
       $sk = (\texttt{seed}, a, b)$ — secret key
output: $k$ — symmetric key on success; otherwise $\perp$

    ▷ recreate public key from secret key seed
    $\texttt{seed\_g} \leftarrow \mathsf{xof}(sk.\texttt{seed}, \lambda/8)$
    $G \leftarrow \mathsf{generate\_g}(\texttt{seed\_g})$
    $C \leftarrow \texttt{sk}.a\, G + \texttt{sk}.b \bmod p$
    ▷ obtain SNOTP and decode seedenc
    $S' \leftarrow \texttt{sk}.a\, \texttt{ctxt}.D \bmod p$
    $\texttt{str} \leftarrow \mathsf{serialize}(S')[0 : (n\nu)] \oplus ctxt.\texttt{seedenc}$
    for $i$ from $0$ to $\nu - 1$ do:
        $s \leftarrow \mathsf{decode}(\texttt{str}[(in) : ((i+1)n)])$
        if $s \neq \perp$ and $\mathsf{H}(s) = ctxt.h$ then:
            break
        end
    end
    if $s = \perp$ then:
        return $\perp$
    end
    ▷ recreate and test ciphertext
    $ctxt', k \leftarrow \mathsf{Enc}(s, \texttt{pk} = (\texttt{g\_seed}, C))$
    if $ctxt \neq ctxt'$ do:
        return $\perp$
    end
    return $k$
```

Algorithm 5: Decapsulate: generate symmetric key and test validity of the given ciphertext.

2.4 Parameter Sets

This document proposes two sets of parameters, called "Ramstake RS 216091", "Ramstake RS 756839". These parameter sets target security levels 128 and 256 in terms of $\log_2$ of required number of operations to mount a successful attack on a classical computer. Both attacks considered in Section 4.3 are fully Groverizable, thus enabling the quantum adversary to divide these target security levels by two. All parameter sets use SHA3-256, SHAKE256, and Reed-Solomon error correction over $\mathbb{F}_{2^8}$ with code length $n = 255$ and design distance $\delta = 224$.

Table 1: Ramstake parameter sets, resulting public key and ciphertext size in kilobytes, and targeted security notion and NIST security level.

π	216091	756839
w	64	128
ν	4	6
λ	256	256
χ	256	256
$\|pk\|$	26.41 kB	92.42 kB
$\|ctxt\|$	27.41 kB	93.91 kB
security	IND-CCA	IND-CCA
NIST level	1	5

3 Performance

3.1 Failure Probability

There is a nonzero probability of decapsulation failure even without malicious activity. This event occurs when the two views of the SNOTP are too different, requiring the correction of too many errors. It is possible to find an exact expression for this probability. However, the following argument opts for a more pragmatic approach.

The Reed-Solomon code used has design distance $\delta = 224$, meaning that it can correct up to $t = \lfloor \frac{\delta-1}{2} \rfloor = 111$ byte errors. Decapsulation fails when all ν codewords contain more than 111 errors. By treating the number of errors e in each codeword as independent normally distributed variables, one can obtain a reasonable estimate of the failure probability.

The Sage script `Scripts/parameters.sage`, which is included in the submission package, computes the mean (μ) and standard deviation (σ) of these distributions empirically. For many different random G and appropriately sparse a, b, a', b', the number of different bytes between $\mathsf{serialize}(aa'G + ba' \bmod p)[0 : 255]$ and $\mathsf{serialize}(aa'G + b'a \bmod p)[0 : 255]$ is computed. From many such trials it computes μ and σ and a recommended number of codewords ν such that the failure probability drops below 2^{-64}. (Indeed, this script is where the values for ν in the parameter sets of Table 2.4 come from.) The statistics are shown in Table 2.

It is possible to push the failure probability even lower by increasing ν. However, this increase results in a larger ciphertext.

Table 2: Mean μ and standard deviation σ of number of errors in a codeword, along with recommended number of codewords ν for a failure probability less than 2^{-64}.

	216091	756839
μ	72.56	81.38
σ	7.89	7.93
ν	4	6
$\left(1 - \Phi(\frac{e-\mu}{\sigma})\right)^{\nu}$	$\leq 2^{-64}$	$\leq 2^{-64}$

3.2 Complexity

3.2.1 Asymptotic

The loops in the pseudocode of Algorithms 1—5 run through a number of iterations determined by the parameters ν, w, π. Of these parameters, ν is independent of the security parameter κ. The relations between w, π and the security parameter κ are more complex. First π must be large enough to spread out roughly $2w^2$ burst-errors so as to guarantee a low enough byte-error-rate and hence non-failure. Second, the slice-and-dice attack of Section 4.3 must be taken into account as well. These parameters are constrained for non-failure by

$$\frac{2w^2}{\pi} \leq c \ , \tag{3}$$

for some constant c roughly around 0.04. For security, the constraint is

$$2w \geq \kappa \ . \tag{4}$$

These equations thus require $\pi \sim \kappa^2$. The size of the public key and ciphertext grows linearly with this number.

While KeyGen, Encaps and Decaps contain only a small fixed number of big field operations, the modulus of this field is p and the field elements involved therefore have an expansion of up to π bits. Nevertheless, there are two available optimizations to ameliorate this cost. (However, none of the provided implementations employ them.)

- Mersenne form. Reduction modulo p does not require costly division as it does for generic moduli. Instead, shifting and adding does the trick. Let $a = a_o \times p + a_r$ with $a_r < p$. Then $a_r + a_o = a \bmod p$.

- Sparsity. In every big field operation, at least one term or factor is sparse. As a result, the sums can be computed through w localized bitflips with carry. The products can be computed through w shifts and as many full additions.

Consequently, the cost of integer arithmetic is linear π and in w. Therefore, the complexity of all three algorithms is $O(\kappa^3)$.

3.2.2 Pratice

The file `perform.c`, which is included in the submission package, runs a number of trials and collects timing and cycle count information. Table 3 presents the information collected from the optimized implementations during 10 000 trials on a Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz machine with 6144 kB of cache on each of its four cores, with 7741 MB of RAM, and running CentOS linux.

Table 3: Implementation statistics — time and cycle count.

	time (ms)	cycles
Ramstake RS 216091		
KeyGen	2.8	9445009
Encaps	5.4	17700978
Decaps	11.1	36706919
Total	19.3	63852906
Ramstake RS 756839		
KeyGen	13.0	43148424
Encaps	24.1	79342014
Decaps	46.9	154721609
Total	84.1	277212047

It is not surprising that Decaps takes the longest, because it runs Encaps as a subprocedure. The striking difference between Encaps and KeyGen is due to the encoding procedure of the error correcting code. Dealing with this error-correcting code is even more costly in Decaps where the errors are corrected.

3.2.3 Memory and Pseudorandomness

It is difficult to estimate the memory requirements of the error-correcting code algebra as well of the big integer arithmetic for two reasons. 1) The current implementation outsources this operation to another library. 2) because this content is highly dynamic: how much memory is needed depends on the value of the mathematical object being represented. By contrast, the memory requirements of the three main functionalities' outputs is easily determined.

The secret key consists of one $\lambda/8$ byte seed and two integers of (after serialization) $\lceil \pi/8 \rceil$ bytes each, although the integers can be generated anew from the seed. The public key contains one seed of $\lambda/8$ bytes and one integer of $\lceil \pi/8 \rceil$ bytes. The ciphertext consists of one integer of $\lceil \pi/8 \rceil$ bytes, a stream of $n\nu$ bytes representing the one-time-padded repetition code, and a hash of $\chi/8$ bytes. Table 4 summarizes these sizes and presents concrete values for the given parameter sets.

All pseudorandomness is generated (*i.e.* extracted from a short seed) in the first line of those functions that need it. So this is $8w + \lambda/8$ for KeyGen, and $8w$ for Encaps. The Decaps function does not require pseudorandomness but it must get the $\lambda/8$-byte seed for G from the secret key seed via the same CSPRNG. Since Decaps invokes Encaps as a subprocedure, it inherits those requirements for extracting and storing pseudorandomness also.

Table 4: Size (in bytes) of output objects.

	secret key	public key	ciphertext
formula	$\lambda/8 + 2\lceil\pi/8\rceil$	$\lambda/8 + \lceil\pi/8\rceil$	$\lceil\pi/8\rceil + n\nu + \chi/8$
Ramstake 216019	54056	27044	28064
Ramstake 756839	189242	94637	96111

4 Security

4.1 Hard Problems

Ramstake relies on the hardness of at least two problems related to finding sparse solutions to affine equations modulo a pseudo-Mersenne prime p. The formal problem statement of the first is as follows.

Low Hamming Combination (LHC) Problem.
Given: Two coefficients $A, B \in \mathbb{F}_p$ in a large Mersenne prime field $\mathbb{F}_p$.
Task: Find two elements $x_1, x_2 \in \mathbb{F}_p$ with binary expansions of Hamming weight at most w_1 and w_2 respectively, such that $B = Ax_1 + x_2 \,\mathsf{mod}\, p$.

The problem was implicitly introduced by Aggarwal *et al.* [1] in the form of an assumption, which states that the distribution $(A, Ax_1 + x_2)$ is indistinguishable from (A, C) when C is drawn uniformly at random and x_1, x_2 uniformly at random subject to having the required Hamming weight. The same paper also introduces the Low Hamming Ratio Search (LHRS) Problem, which asks to find a pair of low Hamming weight integers x_1, x_2 satisfying $x_2/x_1 = H$. The LHRS Problem is equivalent to the subset of the LHC Problem where $B = 0$. (To see this, set $H = -A$. $\square$)

The LHC problem is only the analogue of the discrete logarithm problem in Diffie-Hellman key agreement. The adversary does not need to compute discrete logarithms; he merely needs to break the Diffie-Hellman problem, which comes in search and decisional variants. The analogues of these problems for sparse integers is formally stated below.

Low Hamming Diffie-Hellman Search (LHDHS) Problem.
Given: Three integers (G, H, F) where $H = x_1G + x_2 \,\mathsf{mod}\, p$ and $F = y_1G + y_2 \,\mathsf{mod}\, p$ for some integers x_1, y_1 of Hamming weight w_1 and x_2, y_2 of Hamming weight w_2.
Task: Find an integer S whose Hamming distance with $x_1F \,\mathsf{mod}\, p$ is at most t, and whose Hamming distance with $y_1H \,\mathsf{mod}\, p$ is also at most t.

Low Hamming Diffie-Hellman Decision (LHDHD) Problem.
Given: Four integers (G, H, F, S) where $H = x_1G + x_2 \,\mathsf{mod}\, p$ and $F = y_1G + y_2 \,\mathsf{mod}\, p$ for some integers x_1, y_1 of Hamming weight w_1 and x_2, y_2 of Hamming weight w_2.
Task: Decide whether or not the Hamming distances between S and $x_1F \,\mathsf{mod}\, p$, and between S and $y_1H \,\mathsf{mod}\, p$, are at most t.

Security requires these problems to be hard, meaning that all polynomial-time quantum adversaries decide the LHDHD Problem with a success probability negligibly far from that of a random guess. The assumed hardness of LHDHD implies that LHDHS is hard as well, which in turn implies that LHC is hard also. It is unclear how to solve LHDHD in a way that avoids implicitly solving LHC.

It is clear that breaking LHDHS is enough to break the scheme, as that allows the attacker to unpad the seed encoding and recover the seed from there. It is not clear whether security also relies on the LHDHD problem but we include that problem for the sake of completeness, because many Diffie-Hellman type cryptosystems rely on the proper analogue of the Decisional Diffie-Hellman problem.

4.2 SNOTP-to-KEM Construction

There is a gap between the Low Hamming Diffie-Hellman Decision Problem and the IND-CCA (or even IND-CPA) security of Ramstake, originating from the SNOTP-to-KEM construction. I am working on a proof of security but it is unavailable at this point. The following obstacles make such a proof highly non-trivial.

- Failure events in the noisy Diffie-Hellman protocol affect security, especially in the chosen ciphertext model.
- The search problems may be solved in more than one way.
- Circular encryption: the one-time pad is not independent of the message it hides.
- The hash functions should be modeled as quantum-accessible random oracles. However, many classical proof techniques fail in the quantum random oracle model.

It is conceivable that a security proof can only be made to work conditioned on some changes being made to the construction, for instance by changing the inputs to the hash functions. Nevertheless, I do not expect the proof to recommend big changes, thus leaving the construction's big picture intact:

- generate noisy Diffie-Hellman protocol contributions from a short random seed;
- use the noisy Diffie-Hellman key to one-time-pad the error-correcting encoding of the seed;
- undo the noisy one-time pad and decode the codeword;
- invoke the encapsulation algorithm with identical arguments and test if the generated ciphertext matches the received one exactly.

4.3 Attacks

4.3.1 Slice and Dice

Beunardeau *et al.* present an attack exploiting the sparsity of the solutions to the LHRS Problem [5], but it applies equally to the LHC Problem. The attack proceeds as follows.

For each trial, partition the range $R = \{0, \ldots, \pi - 1\}$ into a number of subranges. This number should not be too large, at most a couple hundred. Do this once for x_1 and once for x_2. This yields

$$R_1^{(0)} \sqcup \cdots \sqcup R_1^{(k-1)} = R_2^{(0)} \sqcup \cdots \sqcup R_2^{(\ell-1)} = R \ . \tag{5}$$

Set each such subrange to active or inactive at random. Ensure that the total cardinality of all inactive subranges is at least π.

Each subrange corresponds to a variable $r_i^{(j)}$ whose binary expansion matches that of x_i but restricted to that subrange. Formulaically, this means

$$x_i = \sum_{j=0}^{k-1} 2^{\mathsf{min}(R_i^{(j)})} r_i^{(j)} \quad \text{and} \quad 0 \leq r_i^{(j)} < 2^{\# R_i^{(j)}} \ . \tag{6}$$

At this point, trim the sums in the left side of Eqn. 6 by dropping the terms that correspond to inactive subranges and replace x_1 and x_2 by their corresponding trimmed sums in the equation $B = Ax_1 + x_2 \mathsf{\ mod\ } p$. Use LLL to find a short solution vector.

A single trial is successful if LLL succeeds in finding the solution that corresponds to the sparse solution. This happens if the guess at inactive subranges is correct, namely if their respective variables are indeed zero (because then their omission does not change the value of the sum).

For the sake of generality, assume x_1 has Hamming weight w_1 and x_2 has Hamming weight w_2. The optimal attacker activates a proportion $\frac{w_1}{w_1+w_2}$ of the range associated to x_1, and a proportion $\frac{w_2}{w_1+w_2}$ of the range associated to x_2. Then the probability of all 1-bits being located inside the active subranges is given by

$$P = \left(\frac{w_1}{w_1 + w_2}\right)^{w_2} \times \left(\frac{w_2}{w_1 + w_2}\right)^{w_1} \ . \tag{7}$$

The formula is a lot simpler when $w_1 = w_2 = w$, and in this case security mandates that

$$2w \geq \kappa \ . \tag{8}$$

This algorithm is fully Groverizable. Therefore, the security level halves when considering quantum adversaries with unlimited circuit depth.

4.3.2 Spray and Pray

Spray and pray is essentially a smart brute force search. Choose a random assignment for x_1 with Hamming weight w_1, compute x_2 from the given information and test if its Hamming weight is at most w_2. Assuming the solution is unique, the success probability of a single trial is one over the size of the search space, or $1/\binom{\pi}{w}$. So κ bits of security requires

$$\mathsf{log}_2 \binom{\pi}{w} \geq \kappa \ . \tag{9}$$

For the parameter sets 216091 and 756839, the left-hand-side of Eqn. 9 is over 838 and 1783, respectively. While the algorithm is fully Groverizable, dividing these numbers by two in order to account for quantum adversaries still results in wildly infeasible complexity.

4.3.3 Stupid Brute Force

Instead of guessing one variable and computing the other from that guess, stupid brute force guesses both at once. A single such guess succeeds with probability $1/\binom{\pi}{w}^2$, *i.e.*, much less likely than the intelligent brute force of the spray-and-pray strategy described above.

Another stupid brute force attack attempts to guess the input of the CSPRNG. By design, these seeds are all 256 bits in length, making for a classical complexity of 2^{256} and 2^{128} quantumly (again assuming unlimited depth).

4.3.4 Lattice Reduction

Aggarwal *et al.* already consider lattice attacks on their cryptosystem and in particular on the LHRS Problem. They observe that it is possible to generate basis vectors for a lattice in which the sought after solution is a short vector. However, that same lattice will contain even shorter vectors that do not correspond to a sparse solution to the original problem. It might be possible to eliminate these parasitical solutions by running lattice reduction with respect to the infinity norm instead of the Euclidean norm, but it is not clear how to do this.

4.3.5 Algebraic System Solving

It is possible in theory to formulate the sparsity constraint algebraically, by constructing polynomials over $\mathbb{F}_p$ that evaluate to zero in all points that satisfy the constraint. At this point a Gröbner basis algorithm can be used to compute a solution. However, the degree of this constraint polynomial is infeasibly large, roughly $\binom{\pi}{w}$. Constructing it requires more work than exhaustively enumerating all potential solutions and testing to see if the linear equations are satisfied.

Another option is to treat the coefficients of the binary expansion of the solutions, as variables in and of themselves. This strategy requires adding polynomials to require that each coefficient lie in $\{0, 1\}$, and that at most w of them are different from zero. The result is a nonlinear system of roughly $4\pi + 2\binom{\pi}{w+1}$ equations in 2π variables with some polynomials having degree $\binom{\pi}{w+1}$. For any practical parameter set, it is infeasible to fully represent this system of equations, let alone to solve it.

4.3.6 Error Triggering

An attacker who can query the decapsulation oracle can obtain feedback on whether the decapsulator was able to decode the transmitted codeword. With enough failures, the attacker can infer the decapsulator's view of the SNOTP. Once the attacker is in possession of this value, he can proceed to decapsulate any ciphertext.

However, in order to exploit this channel of information, the attacker must generate ciphertexts that fail during decapsulation. If his query ciphertext is not the exact output of the encapsulation algorithm upon invocation with the transmitted seed, then the manipulation will trigger a decapsulation failure regardless of whether decoding was successful. In other words, in order to obtain meaningful information about failure events, the attacker must restrict himself to querying only legitimate outputs of Encaps. Worse still, he has no way of knowing beforehand whether or not a ciphertext is more or less likely to cause failure before the first failure response. Since the failure probability is less than 2^{-64}, the attacker has to make on the order 2^{64} honest queries to get this first failure response.

5 Advantages and Limitations

Advantage: Simplicity. Simplicity is the key selling point of Ramstake. Simple schemes are easier to implement, easier to debug, and easier to analyze. While simpler schemes are sometimes also easier to break, a scheme's resilience to attacks should not rely on its complexity.

Advantage: Problem Diversity. Ramstake relies on different hard problems compared other branches of post-quantum cryptography. Consequently, breakthroughs in cryptanalysis or hard problem solving that break or severely harm other schemes may leave Ramstake intact.

Limitation: New Hard Problem. The hard problem on which Ramstake relies is new and understudied. As a result, it does not offer much assurance of security compared to schemes that have existed (and remained unbroken) for much longer.

Limitation: No Proof. Ramstake claims to offer IND-CCA security even though there is no security reduction to the underlying hard problem. It is therefore conceivable that an attack might break the scheme even without solving the hard problem. Nevertheless, simply because something has not been proven secure yet does not mean it is insecure.

Limitation: Bandwidth and Speed. Lattice-based KEMs are likely to be faster and to require less bandwidth. Nevertheless, Ramstake is competitive in comparison to the very first lattice-based and code-based cryptosystems, and it is conceivable that sparse integer cryptosystems will undergo a similar evolution. However, potential future improvements should not be considered for standardization at this point.

Acknowledgments

The author is thankful to Fré Vercauteren, Reza Reyhanitabar and Ward Beullens for useful discussions and insights. Also, the feedback from NIST after the September deadline was highly useful and highly appreciated. The author is being supported by a Ph.D. Fellowship from the Institute for the Promotion of Innovation through Science and Technology in Flanders (VLAIO, formerly IWT).

References

[1] Aggarwal, D., Joux, A., Prakash, A., Santha, M.: A new public-key cryptosystem via mersenne numbers. IACR Cryptology ePrint Archive 2017, 481 (2017), http://eprint.iacr.org/2017/481, version of 30 May 2017.

[2] Aguilar, C., Gaborit, P., Lacharme, P., Schrek, J., Zémor, G.: Noisy diffie-hellman protocols (2010), https://pqc2010.cased.de/rr/03.pdf, PQCrypto 2010 The Third International Workshop on Post-Quantum Cryptography (recent results session)

[3] Aguilar, C., Gaborit, P., Lacharme, P., Schrek, J., Zémor, G.: Noisy diffie-hellman protocols or code-based key exchanged and encryption without masking (2010), https://rump2010.cr.yp.to/fae8cd826597867589335232978 6cea2.pdf, CRYPTO 2010 (rump session)

[4] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Newhope without reconciliation. IACR Cryptology ePrint Archive 2016, 1157 (2016), http://eprint.iacr.org/2016/1157

[5] Beunardeau, M., Connolly, A., Géraud, R., Naccache, D.: On the hardness of the mersenne low hamming ratio assumption. IACR Cryptology ePrint Archive 2017, 522 (2017), http://eprint.iacr.org/2017/522

[6] Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. IACR Cryptology ePrint Archive 2001, 108 (2001), http://eprint.iacr.org/2001/108

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
COSIC
Leuven
B-3001 Leuven