Security of Distance-Bounding: A Survey

GILDAS AVOINE, INSA Rennes, IRISA, IUF, France
MUHAMMED ALI BINGÖL, TÜBİTAK BİLGEM, and Sabanci University, Turkey
IOANA BOUREANU, Imperial College London, United Kingdom
SRDJAN ČAPKUN, ETH Zurich, Switzerland
GERHARD HANCKE, City University of Hong Kong, Hong Kong SAR
SÜLEYMAN KARDAŞ, Batman University, Batman, Turkey
CHONG HEE KIM, Université catholique de Louvain, Louvain-la-Neuve, Belgium
CÉDRIC LAURADOUX, INRIA, France
BENJAMIN MARTIN, Université catholique de Louvain, Louvain-la-Neuve, Belgium
JORGE MUNILLA, University of Málaga, Spain
ALBERTO PEINADO, University of Málaga, Spain
KASPER BONNE RASMUSSEN, University of Oxford, United Kingdom
DAVE SINGELÉE, Katholieke Universiteit Leuven and iMinds, Belgium
ASLAN TCHAMKERTEN, Telecom ParisTech, Paris, France
ROLANDO TRUJILLO-RASUA, University of Luxembourg, SnT, Luxembourg
SERGE VAUDENAY, EPFL, Lausanne, Switzerland

Distance bounding protocols allow a verifier to both authenticate a prover and evaluate whether the latter is located in his vicinity. These protocols are of particular interest in contactless systems, e.g., electronic payment or access control systems, which are vulnerable to distance-based frauds. This survey analyzes and compares in a unified manner many existing distance bounding protocols with respect to several key security and complexity features.

CCS Concepts: • **Security and privacy** → **Cryptography**; **Authentication**; **Hardware-based security protocols**; *Security protocols*; • **Computer systems organization** → **Embedded and cyber-physical systems**; • **Hardware** → *Radio frequency and wireless interconnect*;

General Terms: Security, Design, Algorithms, Performance

The work of Dave Singelée is supported in part by the IAP Programme P6/26 BCRYPT of the Belgian State, by the European Commission under contract number ICT-2007-216676 ECRYPT NoE phase II, by the Flemish IWT SBO project MobCom, and by the Research Council K.U.Leuven: GOA TENSE. The work of Rolando Trujillo-Rasua has been supported by the Luxembourg National Research Fund (FNR) under grant C15/IS/10428112 (DIST). Finally, the work has been partly supported by the COST Action IC1403 (Cryptacus).

DOI: 10.1145/nnnnnnn.nnnnnnn

Additional Key Words and Phrases: Information Security, Cryptography, Contactless, Relay attacks, Distance fraud, Mafia fraud, Terrorist fraud, Distance bounding, Proximity check

ACM Reference format:
Gildas Avoine, Muhammed Ali Bingöl, Ioana Boureanu, Srdjan Čapkun, Gerhard Hancke, Süleyman Kardaş, Chong Hee Kim, Cédric Lauradoux, Benjamin Martin, Jorge Munilla, Alberto Peinado, Kasper Bonne Rasmussen, Dave Singelée, Aslan Tchamkerten, Rolando Trujillo-Rasua, and Serge Vaudenay. 2017. Security of Distance-Bounding: A Survey. *ACM Comput. Surv.* 1, 1, Article 1 (April 2017), 71 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

---

1 INTRODUCTION AND STATE-OF-THE-ART

1.1 From Relay Attacks to Evolved Distance-based Frauds

The basic concept of a relay attack was first described by Conway [25] in 1976, in a scenario referred to as the *Chess Grandmaster Problem*. In this scenario, any player could play against two Grandmasters by challenging both of them to a game of chess by post. The player would then simply forward the move received from one Grandmaster to the other, effectively making them playing against each other. This results in the player either winning one match, or earning a draw in both matches. Desmedt, Goutier, and Bengio [28] extended this concept to security protocols in 1987, with an attack on the Fiat-Shamir protocol [31, 32] they named *mafia fraud*. In general, a protocol is seen to be executed between a party making a claim, the prover, and a party verifying this claim, the verifier. Mafia fraud involves a malicious third party who aims to convince the verifier that he is the legitimate prover. To start, the third-party simply takes all the messages sent by the verifier and forwards these to the prover. As the messages are legitimate, the prover believes he is communicating with the legitimate verifier. The prover then generates a valid response which the third party forwards to the verifier. Upon receiving this response, the verifier is convinced that he is communicating with the legitimate prover and the attack succeeds. A variant of mafia fraud, denoted by *terrorist fraud*, is an attack in which the prover colludes with the adversary to deceive the verifier, and was subsequently proposed by Bengio et al. [10]. In practice, this involves a prover sharing protocol information, other than key material, with a third-party in such a way that he allows this third-party to convince the verifier that he is the legitimate prover without having to relay all the verifier's messages.

Even though mafia fraud could be classified as a special type of man-in-the-middle attack, there are fundamental differences between these attacks. In man-in-the-middle attacks, the third party actively modifies messages between the verifier and the prover, and in general the attack is made possible through a security vulnerability in the protocol. In other words, man-in-the-middle attacks can be mitigated with conventional security mechanisms. In mafia fraud, the third party is passive and simply relays messages. He does not need to perform any further logical attack on the messages or the protocol sequence and in fact the third party does not even need to know what he is relaying. The protocol

and security mechanisms are irrelevant as the attacker just relays the entire message generated by the legitimate parties, regardless of their content, thereby ensuring that both the verifier and the prover always receive a valid message. Conventional security mechanisms are therefore not an effective countermeasure.

Brands and Chaum early proposed the idea of using so-called distance-bounding protocols [17] to mitigate mafia fraud. In addition to mafia fraud, Brands and Chaum also considered the possibility of *distance fraud*. Distance fraud involves a fraudulent prover that wants to convince the verifier that he is closer than he really is. Most recently, a new fraud termed *distance hijacking* was proposed [26]. In this case, a fraudulent prover takes advantage of a protocol executed between an honest prover and the verifier. The fraudulent prover selectively uses parts of this protocol instance to convince the verifier that he is at a distance, at which some other honest prover resides, which differs from the actual distance of the dishonest prover to the verifier.

1.2 Practical Attacks

The frauds discussed above are of practical significance when considering real-world system security. For example, mafia fraud is especially relevant in access control and payment systems. An RFID door access reader might authenticate an access token by transmitting a challenge, e.g., a nonce, and then checking whether the cryptographic response, constructed with the token's key, is valid. In such a case, an attacker can present a proxy-token, a device under the attacker's control that emulates a token, to the door reader. At the same time his accomplice has a proxy-reader, a reader under the attacker's control, which is used to communicate with a legitimate token. This can be done in a covert manner, e.g., holding the reader against the token holder's pocket while he is outside the premises. The attacker's proxy-token gets the challenge from the door reader and transmits it to the accomplice's proxy-reader. The latter sends the challenge to the legitimate token. The proxy-reader thus obtains the valid response, which is transmitted to the proxy-token and then sent to the door reader. The door reader is now convinced that the token it is communicating with is the legitimate token and opens the door. A practical mafia fraud of this nature was first demonstrated by Hancke [40], using a built-for-purpose proxy-token and relaying radio channel with an effective range of 50 meters, alongside a modified off-the-shelf reader for the purpose of proxy-reader. Francillon, Danev, and Čapkun have also practically demonstrated the feasibility of mafia fraud against remote keyless entry systems in vehicles [35].

Similarly, payment systems are also vulnerable to mafia fraud. An attacker could convince a customer to insert his payment card into a proxy-reader, perhaps to pay for a low-value item sold to the customer by the attacker. The attacker's accomplice, in the meantime, purchases a high-value item and inserts his proxy-card into the merchant's reader. The high-value transaction is then conducted, via the proxy devices, with the legitimate payment card. The proxy-reader only displays the low value amount for customer approval, who thinks that he is authorizing the transaction by entering his PIN on the proxy-reader. This PIN is transmitted to the accomplice and it is entered into the merchant's reader,

which then verifies the PIN through the relay setup with the legitimate card. As a causality the customer ends up paying for the attacker's item. This attack scenario was implemented against the "Chip and Pin" card payment system in the United Kingdom by Drimer and Murdoch [29], and illustrates that mafia fraud can be a serious threat even when systems use strong cryptography and two-factor authentication. The implementation of near-field communication (NFC) in mobile phones has potentially decreased the complexity of implementing mafia fraud. An NFC-enabled mobile phone can act as a token and a reader, so it can either act as a proxy-token or proxy-reader, while offering multiple options with regards to communication channels for relaying messages. The potential use of NFC devices in mafia fraud is documented by Kfir and Wool [47], and a practical mafia fraud using NFC-enabled mobile phones has already been demonstrated by Francis et al. [37]. Some additional attack scenarios and a discussion on the practical implementation of mafia fraud can be found in [36, 44].

Real-time location systems (RTLS) are increasingly used to track high-value assets and people. A RTLS relies on the fact that the physical relation between reference nodes, with fixed and known locations, and the target can be estimated. If these estimates are somehow modified by an attacker then the overall localization process will be adversely affected and the location of the target could be misrepresented. Čapkun and Hubaux [21] have shown that in the case of trilateration, and the principle extends to multilateration, a target located outside a triangle of reference nodes cannot prove that it is inside the triangle without shortening the distance measured to at least one of the reference nodes. Similarly, a node located inside the triangle cannot prove it is at a different location without decreasing the measured distance to at least one of the nodes. This means that a fraudulent prover, wishing to misrepresent his own location, must perpetrate distance fraud against at least one reference node. In practice, distance fraud is relatively simple in certain RTLS systems. For example, if the distances are estimated using received signal strength, then an attacker could selectively attenuate or amplify his communication with a specific reference node. Some practical distance fraud strategies enabling a fraudulent prover to decrease the round-trip-time of his responses are discussed in [43] and [24].

Finally, relay attacks are particularly relevant in the field of digital rights management (DRM), although this issue is rarely discussed in the literature. For example, a provider may refuse to deliver a content to the customer if the latter is not in a clearly defined location, as stated for example in [1, 19, 58].

1.3 Countermeasures to Relay Attacks

To counteract relay attacks, we need to look beyond the data exchanged and incorporate the physical context of the interaction between verifier and prover into the protocol. Desmedt was the first to introduce solutions of this type [27]; he proposed to sign the prover GPS coordinates. In a second proposal [11], the notion of timed message exchanges was introduced. Using precise timing, Beth and Desmedt managed to detect the little girl fraud via the delay introduced by the relay. Desmedt, alone in a first time, then later along with Beth, was the first to remark that countering mafia fraud implies relying on physical properties

(localization, or timing) rather than only depending on the cryptographic parts. This observation yields to several propositions to measure this physical property. Among them, distance bounding protocols were the most promising countermeasure. Distance bounding protocols can be built on the Received Signal Strength (RSS) [8], by measuring the Angle-of-Arrival (AoA) [38], the noise level [23], the physical property of the communication channel [67], the ambient environment [39, 72], or the measure of the Time Of the Flight (ToF).

The RSS, and the AoA methods are usually discarded due to implicit security flaws. Indeed, an adversary can by increasing its signal strength or building special antenna deceive these measurements [22]. Methods based on the noise level or on channel properties work in theory. However, they are not practical to implement.

ToF methods are more reliable, and often used to evaluate the distance d between two parties by calculating $d = s_{pr} \cdot t_p$ where s_{pr} is the propagation speed of signals on the medium of the communication channel and t_p is the one-way propagation time between the transmitter and the receiver [42]. An attacker committing mafia fraud will unavoidably increase the time that the message takes to travel between the prover and verifier. Even simply forwarding and transmitting messages increases the ToF. Measuring this time and checking for unexpected delay in a response is therefore recognized as a feasible method for detecting mafia fraud [11]. ToF distance estimation comprises both Time-of-arrival (ToA) or round-trip-time (RTT) approaches. ToA requires both a verifier and prover to share a synchronized, high-precision clock and only the propagation time of a single message is measured. For example, the verifier sends a challenge *chall* to the prover, and records the time t_0 it was sent. The prover records the time $t_0 + t_p$ the challenge was received and responds with the authenticated message $\{t_0 + t_p, chall\}$. If both the prover and the verifier are trusted, this protocol is effective in detecting mafia frauds. However, it is vulnerable to distance fraud as the prover can simply decrease the value of $t_0 + t_p$ in the response to appear closer. From a practical perspective, both the prover and the verifier might not have a synchronized precise clock, e.g., an RFID reader could have such a precise clock but an RFID tag not.

1.4 Distance-Bounding Based on RTT

Both these issues can be addressed using an RTT distance-estimation approach. In RTT, the verifier measures the time t_m from the moment he has sent a challenge to the moment the response is received. The verifier is therefore completely in control of the measurement and he is also the only entity that requires a precise clock. In this case the verifier can estimate the distance $d = c \cdot (t_m - t_d)/2$, where t_m is the round-trip-time, equal to $2 \cdot t_p + t_d$, and t_d is the time the prover takes to calculate the response. For example, the verifier sends a challenge *chall* at t_0, which the prover receives at $t_0 + t_p$. The prover sends a response back at $t_0 + t_p + t_d$ and this is received at $t_v = t_0 + 2 \cdot t_p + t_d$, allowing for the RTT to be calculated as $t_m = t_v - t_0$. The fraudulent prover can no longer directly influence the measurement, as is the case with ToA, but he could try to send his response earlier than he receives the challenge.

To prevent this, a protocol must be designed in such a way that the response depends on the challenge, i.e., $r = f(chall)$, so that the prover has to wait for the challenge before responding. This response function f also determines the length of the processing time t_d, which must be minimal and deterministic, to achieve an accurate estimate. The response function must therefore be of minimal complexity and should be processable in a short and predictable time.

Distance-bounding protocols are closely linked to aspects of the physical communication channel, a side effect of requiring accurate timing measurements. The channel on which the challenges and responses are to be transmitted must therefore be chosen in such a way that it does not adversely affects the security of the protocol or the accuracy of the distance estimated. Conventional communication channels have been shown to be unsuitable for secure distance-bounding protocols, due to the possibility that an attacker could exploit the latency introduced in these channels by error-resistant measures, such as framing/integrity data and filters in transceivers [24, 43]. In practice, building a distance-bounding channel is a hard problem. Even if we only consider the distance estimation requirements, a timing measurement error of 1 ns could result in a distance estimation error of approximately 30 cm, and measuring the RTT to this level of accuracy is not feasible in systems often suggested to benefit from distance bounding. Implementing suitable channels is still an open research question, although there are several proposals already described and practically demonstrated in the literature [29, 41, 62–64].

In 2006, Clulow et al. [24] proposed four principles for implementing a secure channel for timed challenge-response exchanges:

(1) Use a communication medium with a propagation speed as close as possible to the physical limit, i.e., speed of light.
(2) Use a communication format in which only a single symbol is transmitted as challenge of response.
(3) Minimize the length of this symbol, or the time taken to decide the value of the symbol.
(4) Design the protocol such that it copes with errors during the challenge-response exchange.

These principles have historical significance, as this work was the first to look at the security implications of the underlying implementation of the exchange channel. However, there is a growing opinion that these principles, aiming for theoretical security, are not fully achievable in practice. As such it is perhaps better to consider the intentions behind the principles' definition, which helps us understand potential security threats and evaluate the effectiveness of a channel used for distance bounding, rather than considering these as hard conditions for secure distance bounding. The first principle advises against the use of channels with a relatively low propagation speed as this would allow an attacker to use a faster channel to relay the communication and not be detected. For example, if distance bounding is conducted across a sound channel the attacker can execute an undetectable relay attack using wired or radio communication. The second principle advises against sending multiple challenges and responses during a

single timed exchange, and against the transmission of any additional information even for purposes of error detection or formatting, e.g., any parity bits, cyclic redundancy checks (CRC), headers/trailers or start/stop bits. In both cases it is shown that a dishonest prover could exploit such exchanges to correctly send a reply earlier than what is expected from a honest prover adhering to the channel rules. The nature of the attack depends on the format of the message but the general idea is that the dishonest prover can calculate and prepare the response before the entire challenge message is received, thus shortening the response time compared to a honest prover waiting for the entire message. The third principle advises that the decision as to the value of the symbol should be made as quickly as possible. If the symbol modulation/encoding is such that the entire symbol must be received the symbol period must be minimized or the receiver should determine the value early on in the symbol period. This is meant to protect against early detect/late send relay attacks, where the attacker can take advantage of the duration between the start of the symbol and when the receiver actually determines its value. For example, when using non-return to zero (NRZ) coding the receiver usually samples the symbol after $t_s/2$, where t_s is the symbol period, which allows for the maximum tolerance to data clock differences between the sender and receiver. If the attacker can sample the symbol at $t_s/10$, he has $4 \cdot t_s/10$ to relay its value and transmit it to the receiver. In such a case, there will be no detectable delay in the communication and distance bounding would be ineffective. To minimize the amount of time available to the attacker the receiver must therefore make its decision as early as possible during the symbol period. The fourth principle, taking into account that principles two and three would not allow for conventional error detection/correction measures and reduces the receiver's tolerances for reliably decoding of data, advises that the protocol cannot expect that the exchange channel will have no communication errors and that this has to be taken into consideration elsewhere in the system.

1.5 Protocol Evolution

Distance-bounding protocols are based on the Round-Trip-Time (RTT) of challenge-response messages, and are essentially meant to detect any unexpected delay in the provers response inherently caused by the messages being relayed over a larger distance by a third party. To effectively achieve this goal, protocols must meet some simple requirements to obtain an accurate propagation time measurement, as explained in the previous section: the response and challenge must be single bits, the response must be dependent on the challenge and the time taken to calculate the response must be minimal and predictable. There are a number of protocols that aim to implement distance-bounding but do not adhere to these requirements, e.g., [11, 58, 75]. However, they are not capable of providing accurate distance estimates because of the variation in the time taken to calculate the response, which makes them unsuitable for many use cases. For example, the time taken by a smart token to encrypt a message or perform a digital signature differs each time. If such a token usually takes 100 ms to calculate a response and if there is even a 0.1% variation, this results in a RTT variation of 0.1 ms and hence a 30000 km distance estimate error. This paper

only considers protocols proposals adhering to the prescribed requirements for distance-bounding.

In a distance-bounding protocol, not all exchanged messages are subject to round-trip-time measurements. The protocol can be divided into three distinct phases: setup, exchange, and verification. During the *setup* phase, the verifier and the prover exchange some initial information and determine the cryptographic material used during the rest of the protocol. During the *exchange* stage, the verifier measures the round-trip-time of the challenge-response pairs. The validity of the responses and the distance-bound is checked during the *verification* stage. The setup and verification phases are commonly referred to as the "slow" phases, while the exchange phase is referred to as the "fast" phase, due to the nature of the communication during these phases. The slow phase uses a conventional channel while the fast phase requires a special channel.

The first distance-bounding protocol was proposed by Brands and Chaum [17]. This protocol, based on Beth and Desmedt's [11] idea that RTT can detect mafia fraud, bounds the distance between the parties by measuring the RTT of single-bit challenges and responses. During the setup phase, the prover cryptographically commits to a random string that he will use to calculate the responses using an XOR operation. During the verification stage the prover signs a message containing the challenges received and the responses sent. The protocol achieves an optimal $\left(\frac{1}{2}\right)^n$ resistance against both mafia fraud and distance fraud, where n is the number of challenge-response exchanges. This concept formed the basis for numerous protocols, whose evolution is represented on Figure 1.

There are four direct descendants of Brands and Chaum's protocol: [20, 42, 59, 63], each of which improved Brands and Chaum in its own way. Peris-Lopez et al. [59] propose that cryptographic puzzles should be used to provide privacy in distance-bounding protocols. Rasmussen and Čapkun protocol [63] is based on XOR and a comparison function, and has the benefit that the prover does not need to demodulate the signal to answer to the verifier's challenges. The MAD protocol proposed by Čapkun et al. [20] allows for mutual distance-bounding. This protocol was enhanced by the protocol of Singelée and Preneel [65], which added bit-error resilience to MAD by using error correcting codes. The Hancke and Kuhn's protocol [42], originally designed to be used in the RFID environment and is thus optimized for execution time and minimal prover complexity, uses pre-computation, instead of a commitment step, during the setup phase in such a way that no additional messages need to be transmitted during the verification stage.

Hancke and Kuhn's protocol has two issues: it does not take terrorist attacks into account, and it achieves a sub-optimal performance-security trade-off with respect to mafia and distance frauds of $\left(\frac{3}{4}\right)^n$. Subsequently, numerous proposals based on the pre-computation method used by Hancke and Kuhn were proposed in an effort to improve its performance. Bussard and Bagga's protocol [18] and all its descendants introduce resistance to terrorist fraud. These protocols are based on Bussard and Bagga's idea that the prover long-term secret is incorporated into the pre-computed response options in such a way that if the prover reveals

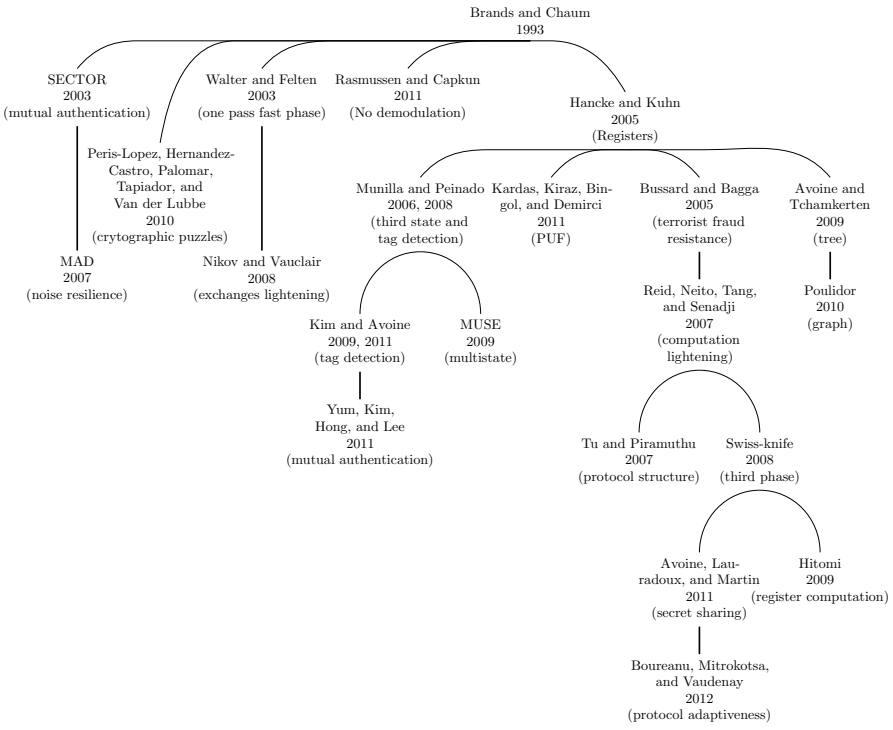


Fig. 1. Distance-bounding evolution

all the options his accomplice would also get the prover's key. This therefore discourages the prover to participate in a terrorist fraud, but at the cost of a complex proof-of-knowledge operation. Its descendants aim to achieve the same functionality but with decreased computational complexity. Reid et al. [64] so improves the computational efficiency but the fraud resistance is $(\frac{3}{4})^n$ in comparison to Bussard and Bagga's $\left(\frac{1}{2}\right)^n$. Tu and Piramuthu's protocol [71] proposes a protocol compounded by a succession of fast and slow phases. However, this protocol suffers from several vulnerabilities, discussed in [50, 56], that reveal the secret to an eavesdropper during a legitimate protocol run. The *swiss-knife* protocol [50] fixes the poor mafia fraud resistance problem by adding a third phase to Reid et al.'s protocol, and it also provides mutual authentication. In [60], the authors claim that they found an attack on this protocol based on nonce repetitions, and thus propose the Hitomi variation. However, if the assumption is made that nonces repeat then Hitomi suffers, to a lesser extent, of a similar flaw. A further variation of the swiss-knife protocol [5] explicitly introduces secret-sharing to counter terrorist fraud, and studies the best settings in which to use it. Avoine and Tchamkerten's protocol [7] introduces binary trees to compute the prover answers during the exchange phase, and succeeds in improving mafia fraud resilience to almost $(\frac{1}{2})^n$. Indeed, various graph structures

can be used instead of a tree structure. The interest of cyclic and q-partite graphs has been demonstrated in [69] and [51, 52], respectively. Finally, Trujillo et al. [70] show that precomputation-based protocols can also deal with noise without sacrificing security.

Munilla and Peinado [54, 55] initiated a new branch of the Hancke-Kuhn pre-computation family. Their protocol communicates during the exchange phase using binary symbols, 0 and 1, and also an additional "nothing" state. MUSE [3] is a generalization of this idea relaxing the number of possible states. Kim and Avoine's protocols [48, 49] enhance the attack detection mechanism. Its descendant [76] uses the detection mechanism to also provide mutual authentication. Finally, Kardaş et al. [46] introduce PUFs in Hancke and Kuhn's protocol and claim the protocol now resists to terrorist fraud.

1.6 Provable Security

Most distance-bounding protocols have been analyzed without a formal approach. Instead, generic best-known attacks are usually adapted to the specific features of the protocol at hand, which has led to unsound analyzes and unfair comparisons. Examples are the protocols proposed in [54], [71], and [76], whose flaws are explored in [2], [56], and [4], respectively. The first comprehensive formalization for analyzing distance-bounding protocols was proposed by Avoine et al. [2]; this is not a provable security formalism, but it is a framework that can describe attack-scenarios in a unitary fashion, and thus offer a systematic manner of computing upper-bounds on the probabilities of typical attacks in distance-bounding and its variants. This unified framework [2] defines the following important objects: the prover model (depending on the prover tampering-resistance, it can be either *black-box* or *white-box*); the prover's computing capabilities (e.g., whether the prover can exploit latencies between the slow and fast phases); and the attacker's strategies (e.g., *pre-ask*, *post-ask*, and *early-reply*).

Recent efforts have been made on proving security for distance-bounding [14–16, 30, 34, 73]. However, this is still a very young field that needs to overcome three main, inter-dependent challenges: (i) the introduction of sound communication, network and adversarial models that capture the notion of time-of-flight, (ii) the definition of clear and rigorous specifications of the classical frauds (i.e., formal definitions of these frauds that can be *proven* to hold or to be refuted within the model), and (iii) formal security proofs based on cryptographic assumptions. To illustrate for instance the difficulty of the third challenge, [13] proved that many protocols fall short in having their security based on the pseudorandom function (PRF) assumption of some underlying primitive.

The first formalism in this direction was put forward by Dürholz et al. [30]. The authors formalize the impossibility of illegitimate yet sufficiently fast round-trip communications using the notion of *tainted sessions*; to encode timing-restrictions, tainted sessions only allow certain flows of communication. Then, a protocol is said to be secure if no adversary executing it with tainted sessions can violate its security properties. The model comprises a formalization of all the classical frauds and provides several (partial) security proofs for some

protocols [30, 33]. This formal model is a step in the right direction towards provably secure distance-bounding.

Another line on provably secure distance-bounding, which builds on the model by Dürholz et al., is in [34]. One addition therein is proposing a distance-bounding protocol that uses not one but two different secret keys for the slow and fast phases. This bypasses the aforementioned problem of using just the PRF-assumption to argue the security of (one-key) distance-bounding.

In [14, 16], the authors provide a rather general model that captures the notion of concurrency (i.e., allowing adversaries to interact with many provers and verifiers, sometimes with the same keys). Their notions of distance and mafia frauds additionally capture the one of distance hijacking [26] and impersonation [30], respectively. Furthermore, their definition for terrorist-fraud is more general than the notion of terrorist fraud adopted in this manuscript: after the initial collusion, the possible threats to protect against may be stronger in [14, 16] (e.g., MiM in concurrent settings). The authors also propose a set of distance-bounding schemes offering provable security against all forms of attacks within their model. A simplified version of [14], where the elements of provable security are played down to best-attack scenarios, is available in [15].

As depicted above, attention to provably secure distance-bounding is increasing. However, we underline that there is little consensus on which formalizations are appropriate, by different metrics. This is evident for instance in the formalization of terrorist-fraud (TF) resistance, arguably due in part to its non-falsifiable nature which –in turn– renders it hard to (provably) attain in distance bounding designs.

Firstly, we underline a distinction between a commonplace view on TF resistance and the formal expressions of this. There is a wide-spread acceptation in the distance bounding literature that TF resistance ought to repose on the reduction to the impossibility to protect against the "trivial vulnerability" whereby a prover gives away his secret-key to the adversary. That is, a protocol is often popularly understood to be TF resistant if the dishonest prover who helps the adversary authenticate fraudulently in one run leaks his secret key to the adversary. Whilst this is a valid acceptation, the formal models above generally do not formalize precisely this commonplace view on terrorist-fraud resistance. Some approaches, e.g., [12, 34], formalize the following statement: the protocol is resistant to TF or some generalizations thereof if whenever the dishonest prover helps the adversary authenticate fraudulently, the adversary gains advantages in future authentication attempts in the absence of the illicit help. Other approaches [73] encode formally that the protocol is sound (or terrorist-fraud resistant in a generalized sense) if the following holds: for all protocol-runs with a verifier, there exists an extractor who reconstructs the secret when he is given the knowledge of all participants which were close to the verifier in several successful executions.

Secondly, authors have changed and abridged their own formal definitions of these expressions of TF resistance. Dürholz et al. define $SimTF$, $StrongSimTF$, and $GameTF$ terrorist-fraud resistance [30, 34]. Bureanu et al. put forward formalizations of terrorist-fraud resistance in [15], as well as formalizations of

generalizations of TF resistance in notions of collusion-fraud resistance [12, 14]. To this end, Vaudenay took collusion-fraud resistance further into a notion of soundness [73] akin to similar expressions in interactive proofs.

Thirdly, one can argue that some of these formal definitions for TF resistance might yield a too strong requirement, disproving security all-throughout (like the $SimTF$ formulation of terrorist-fraud resistance in [30]), or might be too general (like aforementioned collusion-fraud resistance in [14] that suits the provable security of the SKI schemes [12]), or less realistic (like the $SimTF$ formulation for terrorist-fraud resistance in [30] in which the dishonest prover and the adversary are not allowed to communicate during the fast rounds). On the one hand, when we fix the model, we can nonetheless see that some of these definitions imply one another ($StrongSimTF$ in [34] implies $SimTF$ in [30], and soundness in [73] implies collusion-fraud resistance in [12], for certain parameters). On the other hand, even in one such fixed model, other definitions remain however incomparable (e.g., $GameTF$ and $StrongSimTF$ in [34]), underlying further the unsettlement of formalizing terrorist-fraud resistance even within one and the same formalism.

Last but not least, formal comparisons between the session-based model in [30, 34] and the model inspired by interactive proofs in [12, 73] do not exist. In the absence of a formal proof aligning the two models and their security definitions, it appears that $SimTF$ resistance in [30] is equivalent to the notion of terrorist-fraud resistance in [15] and that $GameTF$ resistance in [34] is equivalent to collusion-fraud resistance in [14] (for some parameters).

Similar discussions apply –of course– to the formalizations of threats other than terrorist-fraud in the aforementioned formalisms. As such, formal relations between the existing formal models for distance-bounding and their formal definitions of security is an avenue of future research.

Due to such differences between the formal models, we decided to carry out our analyses in the general framework by Avoine et al. [2]. This framework does not repose on such fine-grained formalizations of the distance-bounding threats[1], but instead it formalizes classes of interactions between the provers and the attackers in order to best classify attack strategies, towards an unitary approach to assessing the security/insecurity of distance-bounding.

1.7 Contributions

This article provides an in-depth security comparison of many existing distance bounding protocols. After the introduction of the notation and the methodology in Section 2, the next twelve sections present several important published distance bounding protocols. Each section presents in a unified way the considered protocol and its security analysis. Those who are not familiar with the presented protocols will be able to consult Appendix A, which provides thorough descriptions of the twelve protocols. Section 15 presents the comparison methodology

[1]For instance, the popular take on TF resistance by reduction to impossible protection against the "trivial vulnerability" is not attainable in the "white box model for TF" from [2], whilst some of the formal expressions for TF resistance summarized above would be.

and results. The article also includes Appendix B, which discusses about variants and extensions that can be applied to most of the considered protocols.

2 ANALYSIS METHODOLOGY AND NOTATIONS

This paper analyzes twelve distance-bounding protocols using a unique methodology, based on the distance-bounding framework published in [2]. Table 1 contains the unified notations used throughout the paper. Each protocol description is divided into 3 steps, namely initialization, protocol, and final phase, and includes a table that summarizes the protocol parameters. Protocols consist of slow phases that are not time-constrained, and fast phases where the verifier measures the round-trip times of exchanged messages. The fast phases are identified with a left square bracket. Anything in the bracket is repeated n times, except if stated otherwise. Each protocol description is followed by a security analysis according to the template provided in Section 2.2. The properties and performance are analyzed according to Section 2.3 and 2.4.

2.1 Fraud definitions

A distance bounding protocol is a process whereby a party (known as *verifier*) is assured (i) of the identity of a second party (known as *prover*) and (ii) that the prover is located in his close vicinity (known as *neighborhood*). Four frauds against distance bounding are usually considered, *impersonation*, *distance*, *mafia*, and *terrorist* frauds [2], which are introduced below.

Impersonation. An *impersonation fraud* is an attack where an adversary acting alone purports to be a legitimate prover.

Distance fraud. A *distance fraud* is an attack where a dishonest prover purports to be in the neighborhood of the verifier. He cheats without help of other entities located in the neighborhood.

Mafia fraud. A *mafia fraud* is an attack where an adversary defeats a distance-bounding protocol using a man-in-the-middle between the verifier and an honest prover located outside the neighborhood.

Terrorist fraud. A *terrorist fraud* is an attack where an adversary defeats a distance-bounding protocol using a man-in-the-middle between the verifier and a dishonest prover located outside of the neighborhood under the following circumstances. The dishonest prover actively helps the adversary to maximize her current attack success probability, but without giving her any advantage for future man-in-the-middle attacks. (In such attacks, the man-in-the-middle (MiM) would attempt to pass the distance-bounding protocol as a valid prover/tag which the MiM does not represent/possess.)

Note that protocols that are known to suffer from a key-recovery attack are not analyzed in this article. This includes Tu and Piramuthu's protocol [71] whose flaws are discussed in [50, 56], Reid et al.'s protocol [64] broken in [5, 53], and Hitomi whose vulnerabilities are described in [66]. While [9] points out a key recovery attack on Bussard and Bagga's protocol [18], this protocol is kept in this analysis because the attacks presented in [9] could be applied to other

Table 1. Notations

Prover and Verifier	
P, ID_P	Prover, Prover identity
V, ID_V	Verifier, Verifier identity
N_V, N_P	Nonces sent by verifier and prover, respectively.
Rounds	
n	Number of rounds in the fast phase
i	Index of the current round
Secrets	
K	Long-term secret key shared by prover and verifier
K_e, K_d	Public/Private keys for Encryption/Decryption
K_s, K_v	Private/Public keys for Signature/Verification
Time	
Δt_i	Round Trip Time (RTT) measured during round i
$t_{\max}$	Threshold on the round-trip time (typically, there is a round failure if $\Delta t_i > t_{\max}$)
Challenges and Responses	
c_i	Challenge sent by the verifier in round i
c'_i	Challenge received by the prover in round i
r_i	Response sent by the prover in round i
r'_i	Response received by the verifier in round i
Registers	
R^0, R^1	Main registers
Z^0,Z^1,...	Additional registers, when needed.
H	Crypto function output, usually viewed as a register, e.g., $H = h(N_V, N_P)$
Sizes	
σ	Size of the signature, commitment, or MAC (in the slow phase)
ι_P,ι_V,ι	Size of ID_P and ID_V. If $\lvert\mathrm{ID}_P\rvert = \lvert\mathrm{ID}_V\rvert$ then the value is denoted ι (bits)
κ	Size of K (bits)
δ_P, δ_V, δ	Size of the nonces N_V and N_P. If $\lvert N_V\rvert = \lvert N_P\rvert$ then the value is denoted δ (bits)
Errors	
e_X	Number of errors of type X, e.g., e_C, e_R, e_T
$e_{\max}$	Threshold on the number of errors
Functions	
$d_{\mathcal{H}}(.,.)$	Hamming distance
$\mathcal{H}(.)$	Hamming weight
$\mathrm{Sign}_{K_s}(.)$	Signature function with private key K_s
$\mathrm{Verif}_{K_v}(.)$	Public-key signature verification function with public key K_v
$\mathrm{Commit}(.)$	Commitment function
$\mathrm{Open}(.)$	Open commitment function
$h(.)$	Cryptographic hash function
$h_K(.)$	Cryptographic hash function keyed with the secret key K
$\mathrm{MAC}_K(.)$	Message authentication code keyed with the secret key K
$f_K(.)$	Pseudorandom function keyed with the secret key K
$E_K(.)$	Encryption function keyed with the secret key K
$D_K(.)$	Decryption function keyed with the secret key K
Misc	
$\mathbb{E}(.)$	Mathematical expectation
$\in_{\mathsf{R}}$	Randomly and uniformly picked in...
$\in_{\mathsf{R}} \{0,1\}^x$	Randomly and uniformly picked in the set $\{0,1\}^x$, typically $x = \delta$
$\lvert\rvert$	Concatenation of words (possibly 1-bit words)
p	Number of runs of the cryptographic function, in the analyzes
$\mathfrak{p}$,$\mathfrak{q}$	Prime numbers.
p_X	Probability of event X
w	Hamming weight, e.g., $w = \mathcal{H}(x)$

protocols and designers must be aware of their existence to avoid them. Note also that the length of the long-term secret keys of the parties, the length of the signatures (when appropriate), and the length of the nonces are assumed to be large enough, such that exhaustive search and replay attack are not relevant. Finally, the pseudo random functions used in the protocols are assumed to be without design flaws, i.e., no trapdoor pseudo random functions, like those discussed in [13].

Another type of fraud, known as *distance hijacking*, has recently been introduced in [26]. The fraud considers a dishonest prover who aims to convince a verifier that he is located within the verifier's neigborhood, abusing for that some other provers who are indeed in the verifier's neigborhood. For example, a dishonest prover can reach his goal by hijacking the fast phase of a distance-bounding protocol executed between an honest (closer) prover and the verifier. Conceptually, distance hijacking can be placed between distance fraud and terrorist fraud. Unlike terrorist fraud, where a dishonest prover colludes with another attacker, distance hijacking considers a dishonest prover who interacts with (abuses) other honest provers. Unlike distance fraud that only involves a dishonest prover and a verifier, distance hijacking also involves other honest provers. These seemingly subtle differences have significant consequences, e.g., the countermeasures proposed against terrorist fraud strictly depend on the fact that the dishonest prover needs to share data with another attacker. In fact, the protocols BC [17], MAD [20], and RC [63] are not resistant against hijacking fraud according to [26]. The version of RC presented in Section 10 comes from [61]. This is a version that has been modified to be resilient to distance hijacking. Cremers et al. provide in [26] a clear analysis of existing protocols that resist to the hijacking fraud. Vaudenay analyzes additional protocols in [74]. We consequently refer the reader to these articles to get more information about distance hijacking.

2.2 Security

The analyses usually performed in distance-bounding do not provide a security proof, but state the resistance of a protocol given a clearly defined scenario, which includes the type of fraud, but also the adversary's capabilities and strategies, described below and summarized in Table 2.

Prover model. Depending on the tamper-resistance of the prover, two models are defined: *black-box* and *white-box*. In the black-box model, the prover can neither observe nor tamper with the execution of the algorithm. In the white-box model, the prover has full access to the implementation of the algorithm and a complete control over the execution environment, as detailed in [2].

Prover computing capabilities. The prover computing capabilities may affect the security of the protocol when considering distance and terrorist fraud in the white box model, given that the prover is also the attacker in such frauds. For example, in HK protocol, the prover may exploit a latency between the slow and fast phases to generate registers with a low Hamming distance [2].

Table 2. Attack scenarios

Fraud	Prover Model	P's Computing Capability	Adversary Strategy	Success Probability
Impers.	(1)	(1)	(4)	$\Pr_{\text{Imp}}$
Mafia	(1)	(1)	*pre-ask*	$\Pr_{\text{MF}\|\text{pre}}$
			post-ask	$\Pr_{\text{MF}\|\text{post}}$
Distance	*black-box*	(2)	*pre-ask & early-reply*	$\Pr_{\text{DF}\|\text{BB}\|\text{pre\&early}}$
			post-ask & early reply	$\Pr_{\text{DF}\|\text{BB}\|\text{post\&early}}$
	white-box	*single run*	*early-reply*	$\Pr_{\text{DF}\|\text{WB(1)}\|\text{early}}$
		multiple run		$\Pr_{\text{DF}\|\text{WB(p)}\|\text{early}}$
Terrorist	*black-box*	(2)	(3)	(3)
	white-box	*single run*	*early-provide*	$\Pr_{\text{TF}\|\text{WB(1)}}$
		multiple run		$\Pr_{\text{TF}\|\text{WB(p)}}$

(1) The computing capability is not relevant with an honest prover.
(2) The computing capability is not relevant in the black box model.
(3) This case is equivalent to the mafia fraud case.
(4) No strategy is defined in [2] for impersonation.

Adversary strategies. The framework [2] points out that three relevant adversary's strategies should be considered when analyzing a distance-bounding protocol: *pre-ask*, *post-ask*, and *early-reply* strategies. In the pre-ask strategy, the adversary relays the first slow phase between the verifier and the prover, then executes the fast phase with the prover before the verifier starts it. In the post-ask strategy, the adversary relays the first slow phase, then executes the fast phase with the verifier without involving the prover. The adversary then queries the prover with the correct challenges received during the fast phase. This strategy is meaningful when the protocol is completed with a second slow phase used to check that the challenges received by the prover are correct. In the early-reply strategy, the adversary anticipates the replies to make them arrive on time, which is particularly relevant with distance fraud. No strategy for the terrorist fraud is defined in [2]. We introduce here the *early-provide* strategy: in this strategy, the adversary located inside the neighborhood, first relays the slow phase to the prover. The latter then provides to the adversary some information to help him to improve her success probability during the fast phase with the verifier. Finally, the adversary relays the final slow phase, if any.

Remark 1 (Circle analysis). *A prover located outside the neighborhood of the verifier but not too far may receive some challenges while the protocol is still running. When the rounds of the fast phase are independent, this late information is useless. However, the adversary may use this information to increase her success probability when the rounds are not independent. Consequently, when analyzing the resistance of a protocol against distance and terrorist frauds the area the prover is located should be considered. However, in all the analyzed*

protocols, either this scenario is not relevant due to the round independency, or the calculation of the success probability is still an open problem.

Remark 2 (Multiple-execution). *The framework also points out that some information could leak when the protocol is executed several times. Typically, this case occurs when the prover and the verifier generate two registers without involving randomness from the prover. None of the protocols analyzed in the paper are known to suffer from this weakness. Consequently, it is not explicitely addressed in the analysis.*

2.3 Properties

The protocol properties considered in the paper are described below and summarized in Table 3. Note that the type of data exchanged during the fast phase is usually binary. This is the case for all protocols considered in this analysis, except the one discussed in Section 14.

Adaptiveness. Indicates whether the protocol provides an adjustable trade-off between resistance to mafia and distance frauds.

Mutual authentication. Indicates whether the protocol provides *mutual* authentication. Note that, mutual authentication does not imply mutual distance-bounding: while the identity proof is bilateral in that case, the distance proof is unilateral in all the analyzed protocols.

Second slow phase. Indicates whether there is a *second slow phase* in the protocol after the fast phase.

Independence of the rounds. Indicates whether each expected response during the fast phase depends on the *current challenge only*.

2.4 Performance

The protocol performance is described below and summarized in Table 4.

Cryptographic primitives. Type of cryptographic primitives needed to be implemented on the prover side: cryptographically-secure pseudo-random number generator, hash, encryption, commitment, and signature. Hash functions and ciphers are actually aggregated into a single category that is denoted *symmetric primitive*.

Exchanged bits (slow phase). Number of exchanged bits during the slow phase(s).

Exchanged bits (fast phase). Number of exchanged bits during the fast phase.

Memory consumption. Amount of memory that is needed during the entire fast phase by the prover.

3 BRANDS AND CHAUM'S PROTOCOL (1993)

In 1993, Brands and Chaum designed several distance-bounding protocols [17]. This analysis focuses on their protocol (Algorithm 1) that mitigates both mafia and distance fraud.

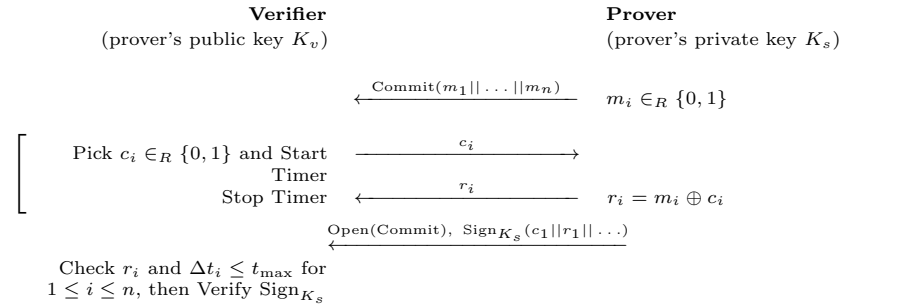
Table 3. Properties

Property	**Value**
Adaptiveness	Yes/No
Mutual authentication	Yes/No/Optional
Second slow phase	Yes/No
Round independence	Yes/No

Table 4. Performance

Performance	**Value**
Cryptographic primitives	Type
Exchanged bits (slow phase)	bits
Exchanged bits (fast phase)	bits
Memory consumption	bits

Algorithm 1: Brands and Chaum's Protocol



3.1 Impersonation

Assuming that the signature scheme is secure, impersonating the prover can only be done by sending a randomly selected correct signature. Such a naive attack has success probability $\Pr_{\text{Imp}} = (1/2)^{\ell}$. However, while nonce-based replay attacks are not addressed in this paper (Section 2.1), a challenge-based replay attack should be considered. Indeed, if the challenges sent by the verifier are used twice, then the adversary can reuse the same m_i's and thus obtains the correct Open(Commit) and $\text{Sign}_{K_s}(c_1||r_1||\ldots||c_n||r_n)$. After eavesdropping one execution before the attack, the success probability becomes $\Pr_{\text{Imp}} = (1/2)^n$.

3.2 Mafia Fraud

Pre-ask strategy. The adversary gets the commitment and queries the prover with random bits (c_i) during the fast phase. The adversary then receives the responses (r_i) and the final signature. With this information, the adversary computes $m_i = c_i \oplus r_i$, then sends the valid responses to the verifier during the fast phase, and finally the commitment and the signature during the second

slow phase. However, the signature received from the prover is not valid for this protocol run, except if the challenges sent by the adversary to the prover and the challenges sent by the verifier to the adversary are the same. The success probability of this strategy is the probability of guessing the challenges correctly: $\Pr_{\mathrm{MF|pre}} = \left(\frac{1}{2}\right)^n$ [2].

Post-ask strategy. The adversary must predict the correct responses to be sent to the verifier during the fast phase without any assistance. We thus have: $\Pr_{\mathrm{MF|post}} = \left(\frac{1}{2}\right)^n$.

3.3 Distance Fraud (White Box)

Early-reply strategy with one run. Given that the adversary must predict the current challenge correctly beforehand, her success probability is provided by the formula: $\Pr_{\mathrm{DF|WB(1)|early}} = \left(\frac{1}{2}\right)^n$ [2].

Early-reply strategy with p runs. No cryptographic function is used to compute registers, contrary to Hancke and Kuhn's approach. This fact trivially yields: $\Pr_{\mathrm{DF|WB(p)|early}} = \Pr_{\mathrm{DF|WB(1)|early}}$.

Circle strategy. Rounds being independent, the circle analysis offers no benefit to an adversary.

3.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. With the pre-ask strategy, the adversary learns all the possible answers. However, she does not know the challenges, so when she sends her answers in advance, two cases occur: a) the verifier uses the same challenge as she did with the verifier. Therefore she always succeeds, b) the verifier picks another challenge and she has sent an incorrect answer to the verifier. Hence, the success probability of this strategy is: $\Pr_{\mathrm{DF|BB|pre\&early}} = \left(\frac{1}{2}\right)^n$.

Post-ask combined with early-reply strategy. Given that the adversary must commit during the first slow phase, she cannot just answer randomly during the fast phase and she will therefore need to predict the responses expected by the verifier. Hence we have: $\Pr_{\mathrm{DF|BB|post\&early}} = \left(\frac{1}{2}\right)^n$.

Circle strategy. We previously stressed that the circle analysis is worthless for this protocol.

3.5 Terrorist Fraud (White Box)

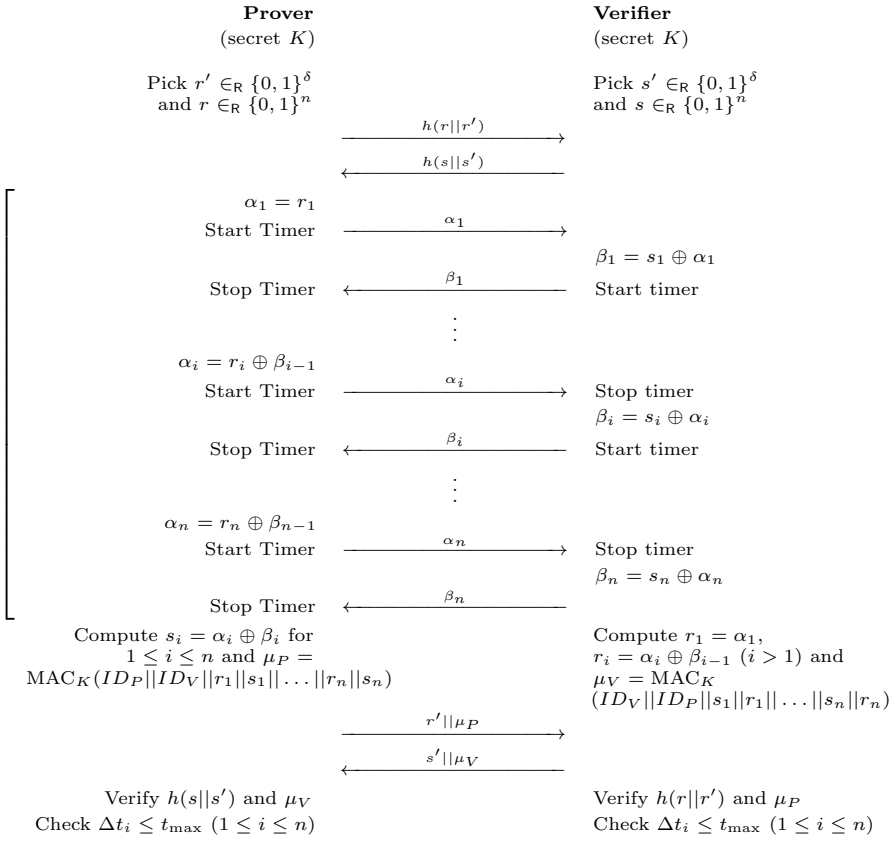
This protocol is not designed to resist to terrorist fraud in the white box model. Indeed, the prover without revealing his secret K_s, is able to provide to his accomplice the commitment and the signature, which are required to succeed. Consequently: $\Pr_{\mathrm{TF|WB}} = 1$ [50].

4 ČAPKUN, BUTTYÁN, AND HUBAUX'S PROTOCOL (2003)

In 2003, Čapkun, Buttyán, and Hubaux introduced MAD [20], a protocol that works quite similarly to the BC protocol [17], but provides mutual authentication.

Although denoted by P and V, the two parties act as both prover and verifier during the execution of the protocol (Algorithm 2). The notations used in [20] are kept in the description below.

Algorithm 2: MAD Protocol



4.1 Impersonation

The basic way to impersonate the prover is to generate the random numbers r and r', and to complete the first slow phase and the fast phase. The adversary must then guess the output of the MAC function in the second slow phase. The probability of a correct guess is: $\Pr_{\mathrm{Imp}} = \left(\frac{1}{2}\right)^\sigma$.

4.2 Mafia Fraud

Without loss of generality, we assume that the adversary seeks to impersonate P against V.

Pre-ask strategy. To succeed in the mafia fraud, the output of the MAC function in the second slow phase needs to be valid. Since the adversary cannot

compute this value, she needs to ensure that P sends the correct output of the MAC function to V. This will only be the case if the adversary has guessed the values s_i correctly during the pre-ask stage. Hence: $\Pr_{\text{MF}|\text{pre}} = \left(\frac{1}{2}\right)^n$ [20].

Post-ask strategy. Similarly, the adversary needs to ensure that P sends the correct output of MAC_K. This will only be the case if she guessed all correct r_i values in advance: $\Pr_{\text{MF}|\text{post}} = \left(\frac{1}{2}\right)^n$ [20].

4.3 Distance Fraud (White Box)

Without loss of generality, we assume that P wants to perform a distance fraud (the distance fraud success probability of V is equal to the one of P).

Early-reply strategy with one run. The responses α_i are computed by XORing the values of the responses r_i, which are completely controlled by the adversary, and the challenges β_i. The latter are uniformly distributed, and the values α_i inherit the same statistical distribution. So even if the adversary fully controls her hardware, the best strategy is to guess the challenges β_i in advance. We have thus: $\Pr_{\text{DF}|\text{WB(1)}|\text{early}} = \left(\frac{1}{2}\right)^n$.

Early-reply strategy with p runs. Similarly to Algorithm 1, no cryptographic function is used to compute registers, and so: $\Pr_{\text{DF}|\text{WB(p)}|\text{early}} = \Pr_{\text{DF}|\text{WB(1)}|\text{early}}$.

Circle strategy. The rounds of the MAD protocol are not independent, so the circle analysis should be applied. Assuming that P knows the challenges of the first $i-1$ rounds, but not the challenge β_i, P must compute the response α_{i+1} without knowing β_i in order to perform a successful distance fraud. However, the rounds are dependent and the following equation holds: $\beta_i = s_i \oplus \alpha_i$. Therefore, P can compute the response α_{i+1} as follows: $\alpha_{i+1} = r_{i+1} \oplus s_i \oplus \alpha_i$. In this equation, everything is known except the value s_i, which is uniformly distributed. Consequently, P has only probability of $1/2$ to compute α_{i+1} correctly. As a result, P does not gain any advantage by knowing the challenges and responses of the previous rounds.

4.4 Distance Fraud (Black Box)

We assume that the fraudulent party that performs the distance fraud is P.

Pre-ask combined with early-reply strategy. The best strategy consists in guessing the n challenges β_i. By querying itself in advance, the prover learns the values r_i, and computes the responses α_i. The adversary uses these responses in the early-reply strategy. They are correct when the values β_i were guessed correctly and consequently the MAC computed by the prover in the second slow phase is correct as well. If one of the challenges is guessed incorrectly, the prover will compute incorrect values s_i, and $\text{MAC}_K(.)$ will be wrong. The distance fraud success probability is: $\Pr_{\text{DF}|\text{BB}|\text{pre\&early}} = \left(\frac{1}{2}\right)^n$.

Post-ask combined with early-reply strategy. The adversary has no information on the bits r_i. The best strategy is to send n random responses α_i. In each round, the adversary has a $1/2$ probability of being successful. This occurs when both r_i and β_i are guessed correctly, or when both guesses were wrong. When

one of these values is correct and the other one is incorrect, the response of the adversary will be wrong. As a result, the distance fraud success probability is: $\Pr_{\text{DF|BB|post\&early}} = \left(\frac{1}{2}\right)^n$.

Circle strategy. The rounds of the MAD protocol are not independent, so the circle analysis can be applied. However, as already demonstrated in the white box case, P does not gain any advantage by knowing the challenges and responses of the previous rounds.

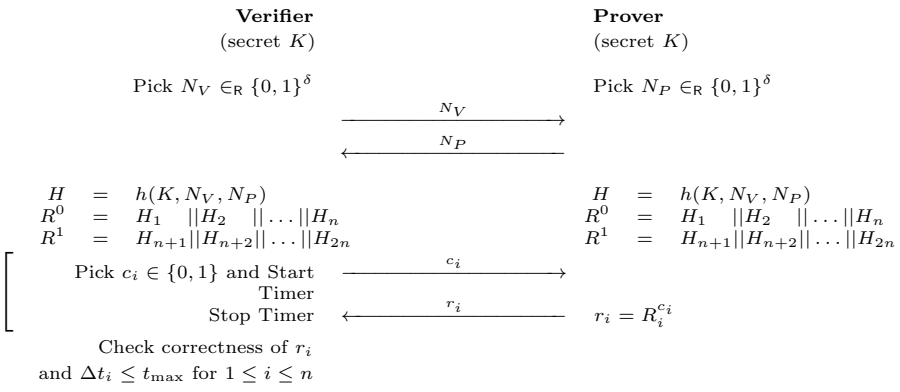
4.5 Terrorist Fraud (White Box)

This protocol is not designed to resist to terrorist fraud in the white box model. Indeed, the prover without revealing her secret K, gives her accomplice the output of the commitment, and the values α_i and r', or β_i and s'. After the fast phase, the accomplice gives the observed values s_i or r_i to the prover, who can then computes the MAC. This output is then sent back to the accomplice, who finally forwards it to the verifier. Hence: $\Pr_{\text{TF|WB}} = 1$ [50].

5 HANCKE AND KUHN'S PROTOCOL (2005)

In 2005 Hancke and Kuhn published the first distance-bounding protocol [42] (Algorithm 3) clearly dedicated to RFID. The protocol relies on the original ideas of Desmedt et al. [10, 28] but is different from Brands and Chaum's work [17] in the sense that Hancke and Kuhn's protocol does not have any final signature after the fast phase.

Algorithm 3: Hancke and Kuhn's Protocol



5.1 Impersonation

The common attack consists in guessing all the answers during the fast phase: $\Pr_{\text{Imp}} = \left(\frac{1}{2}\right)^n$.

5.2 Mafia Fraud

Pre-ask strategy. We have: $\Pr_{\text{MF|pre}} = \left(\frac{3}{4}\right)^n$ [42].

Post-ask strategy. This protocol does not contain any second slow phase and the first slow phase consists of nonce exchanges only. As per Section 2 we have: $\Pr_{\mathrm{MF|post}} = \Pr_{\mathrm{Imp}}$.

5.3 Distance Fraud (White Box)

Early-reply strategy with one run. We have: $\Pr_{\mathrm{DF|WB(1)|early}} = \left(\frac{3}{4}\right)^n$ [69].

Early-reply strategy with p runs. The formula expressing the attack success probability for this strategy was originally presented in [2], but it contained a typing error. The correct formula is:

$$\Pr_{\mathrm{DF|WB(p)|early}} = \frac{1}{2^{pn}} \cdot \left(\sum_{i=0}^{i=n-1} \left(\frac{1}{2}\right)^i \cdot \left[\left(\sum_{j=i}^{j=n} \binom{n}{j} \right)^p - \left(\sum_{j=i+1}^{j=n} \binom{n}{j} \right)^p \right] + \left(\frac{1}{2}\right)^n \right).$$

Circle strategy. Rounds being independent, the circle analysis offers no benefit to an adversary.

5.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. With the pre-ask strategy, the adversary learns half of the possible responses. However, she does not know the challenge, so when she sends her responses in advance, two situations can occur: 1) the verifier asks her the same challenge that she asked the prover and therefore her response is correct, 2) the verifier sends a different challenge in which case she succeeds if the two possible responses were the same, i.e., her response is the same as the alternative response, and fails if the possible responses are different. Hence, the distance fraud success probability is: $\Pr_{\mathrm{DF|BB|pre\&early}} = \left(\frac{3}{4}\right)^n$

Post-ask combined with early-reply strategy. This protocol does not contain any second slow phase and the first slow phase consists of nonce exchanges only. As per Section 2 we have: $\Pr_{\mathrm{DF|BB|post\&early}} = \Pr_{\mathrm{Imp}}$.

Circle strategy. We previously stressed that the circle analysis is worthless for this protocol.

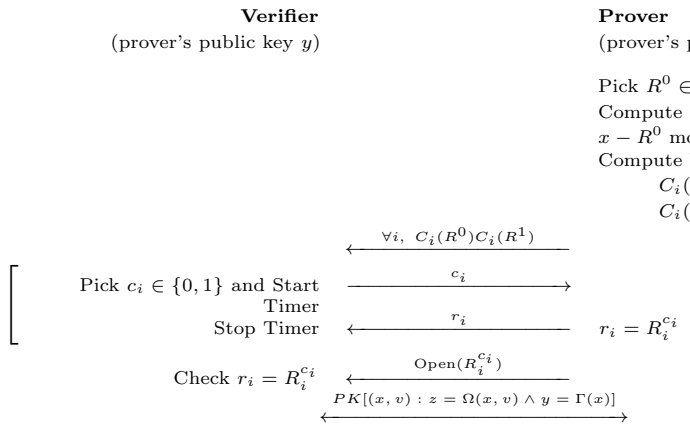
5.5 Terrorist Fraud (White Box)

This protocol is not designed to resist to terrorist fraud in the white box model. Indeed, the prover is able to provide to his accomplice the two registers required to successfully execute the protocol, without revealing his secret K: $\Pr_{\mathrm{TF|WB}} = 1$ [50].

6 BUSSARD AND BAGGA'S PROTOCOL (2005)

Bussard and Bagga published the DBPK-Log protocol (Algorithm 4), which is a distance-bounding protocol based on a proof of knowledge and a commitment scheme [18].

Algorithm 4: DBPK-Log Protocol



6.1 Impersonation

In [18], the authors described a statistical key recovery attack. They established the success probability of this attack: $\Pr_{\text{Imp}} = (1/2)^{-4m'}$, where m' is a security parameter.

6.2 Mafia Fraud

Pre-ask strategy. The adversary must pass the final slow phase to defeat the protocol. Forging the Open function is definitely not the best option. Instead the adversary should try to send the correct challenges to the prover during the pre-ask attack, and then relay the final slow phases. Her success probability with such a strategy is: $\Pr_{\text{MF}|\text{pre}} = (1/2)^n$.

Post-ask strategy. Due to the presence of a complex second slow phase, the post-ask strategy is as good as the pre-ask strategy against DBPK-Log, which yields the success probability: $\Pr_{\text{MF}|\text{post}} = (1/2)^n$.

6.3 Distance Fraud (White Box)

Early-reply strategy with one run. The adversary can search for a random R^0 that minimizes the Hamming distance between R^0 and R^1. Denoting a as the Hamming distance between R^0 and R^1 ($a = d_{\mathcal{H}}(R^0, R^1)$), we have: $\Pr_{\text{DF}|\text{WB}(1)|\text{early}} = (1/2)^a$. This points out that the analysis provided in [18] under-evaluates the success probability of the adversary because the white box model is not considered.

Example 6.1. Let us consider the safe prime $p = 59$ ($q = 29$ and $n = 6$) and $x = 27$. If $R^0 = 32$, then $d_{\mathcal{H}}(R^0, R^1) = 1$ and so the success probability is $(1/2)$.

Early-reply strategy with p runs. Running the pseudo-random generator for choosing R^0 once, or several times, has no impact in the protocol security: the

malicious prover can choose an appropriate value for R^0 in order to maximize its success probability in distance fraud. Hence: $\Pr_{\mathrm{DF|WB(p)|early}} = \Pr_{\mathrm{DF|WB(1)|early}}$.

Circle strategy. Rounds being independent, the circle analysis offers no benefit to an adversary.

6.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. The adversary must succeed in the second slow phase and has the same success probability as mafia mraud. We then have: $\Pr_{\mathrm{DF|BB|pre\&early}} = (1/2)^n$.

Post-ask combined with early-reply strategy. The adversary must succeed in the fast phase without the knowledge of the challenge. Then, the prover is queried by the adversary to gain information for the final slow phase. Similar to mafia fraud, the success probability is: $\Pr_{\mathrm{DF|BB|post\&early}} = (1/2)^n$.

Circle strategy. We previously stressed that the circle analysis is worthless for this protocol.

6.5 Terrorist Fraud (White Box)

Early-provide strategy with one run. This protocol is designed to resist to terrorist fraud in the white box model. Indeed, the prover cannot reveal R^0 and R^1 without exposing the key, making him able to provide only R^0 or R^1 to the external adversary. Note that the prover cannot try to optimize the Hamming distance between R^0 and R^1 as in distance fraud.

Remark 3. *The probability of terrorist fraud calculated in [18] is lower than the one provided here. Indeed, the authors considers that the final slow phase cannot be relayed by the adversary.*

Early-provide strategy with p runs. Similarly to distance fraud strategy with early provide one run, we have $\Pr_{\mathrm{TF|WB(p)}} = \Pr_{\mathrm{TF|WB(1)}}$.

A Dedicated Distance Fraud and Terrorist Fraud. We describe here a distance fraud attack from [9]. The key idea is that a malicious prover could select $R^0 \approx \frac{x}{2} \bmod (p-1)$. That is, if x is even, he takes $R^0 = \frac{x}{2}$ and gets $R^1 = R^0$. Otherwise, he takes $R^0 = \frac{x \pm 1}{2}$ and get $R^1 = R^0 \pm 1$ so that R^0 and R^1 differ in their least significant bit only. He can then run the protocol normally. We note that $R_i^0 = R_i^1$ except for one single round. So, the answers to the received challenges do not depend on it, except in one round. By sending the answer before the challenge arrives, the malicious prover can succeed in an early-reply strategy to run a distance fraud with a success probability larger than $\frac{1}{2}$.

The paper [9] also describes a terrorist fraud attack. The idea of the attack is that the malicious prover starts the protocol but does not give the commit values. Instead, he computes z and discloses it to the adversary through an early-provide strategy. The adversary will commit to random bits for R_i^0 and R_i^1 except for round $i = 1$. Then, he guesses the value c_1 and commit to a random bit for $R_1^{c_1}$. Finally, the commit value for $R_1^{1-c_1}$ is adjusted so that the equation $z = \prod_{i=1}^{n} (C_i(R^0)C_i(R^1))^{2^{i-1}} \bmod p$ holds. Clearly, the adversary can answer all

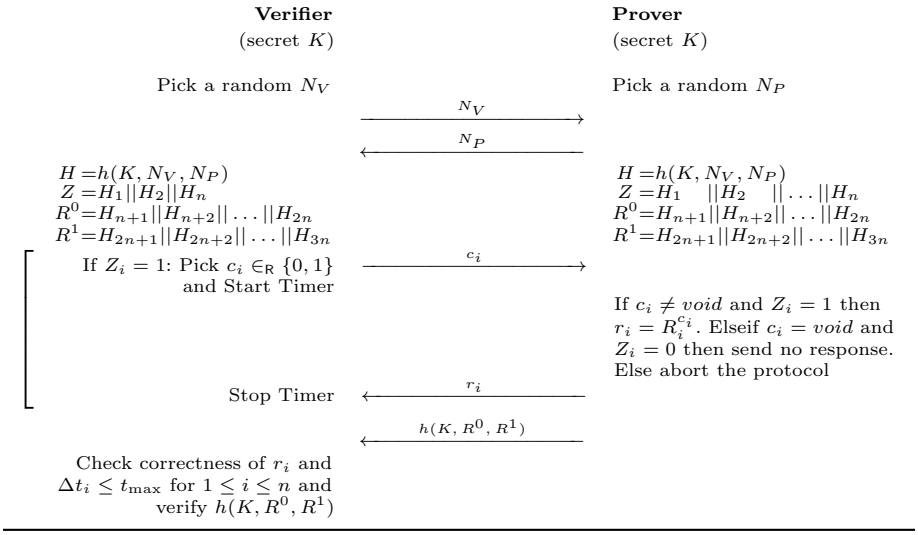
challenges (if his guess for c_1 is correct), since he knows the bits he committed to. Next, he can get the help of the malicious prover to run the PK protocol through the slow phase. Due to the zero-knowledge property of the PK protocol, this leaks no information about x. This attack works with probability $\frac{1}{2}$ (due to the guess of c_1).

Finally, the paper [9] proposes some man-in-the-middle attacks against variants of this protocol which are not using public-key cryptography, i.e., where PK is not used and x is shared.

7 MUNILLA AND PEINADO'S PROTOCOL (2006)

Munilla and Peinado introduced in [54, 57] the concept of void challenges as a tool to improve distance-bounding protocols. These void challenges can also be used to decrease the mafia fraud success probability when applied to Hancke and Kuhn's protocol [55], which is the case analyzed in this section. Thus, for this protocol (Algorithm 5), the challenges can be 0, 1 or void, where a void challenge means that no challenge is sent. Void challenges are used to detect a mafia fraud attack using the pre-ask strategy.

Algorithm 5: Munilla and Peinado's Protocol



7.1 Impersonation

The adversary must guess the responses to the non-void challenges and the signature. Hence:

$$\Pr_{\text{Imp}} = (1 - \frac{p_f}{2})^n \cdot (\frac{1}{2})^{3n}$$

7.2 Mafia Fraud

Pre-ask strategy. The calculation of the success probability of the pre-ask strategy is:

$$\Pr_{\mathrm{MF|pre}} = \begin{cases} \left(1 - p_f\right)^n & \text{if } p_f < 4/7 \\ \left(p_f \cdot \frac{3}{4}\right)^n & \text{if } p_f \geq 4/7 \end{cases} \qquad [2]$$

Note that $\Pr_{\mathrm{MF|pre}}$ calculated in [2] and provided above is an approximation of the real value. Indeed, once the adversary is detected by the device, she does not receive any useful information any more. However, she can still guess the correct answers to be sent to the verifier. Given that being detected by the device forces the adversary to guess the final signature, this case is nevertheless negligible (Section 2.1).

Post-ask strategy. The adversary must predict the correct responses to the non-void challenges. We so have:

$$\Pr_{\mathrm{MF|post}} = \left(1 - \frac{p_f}{2}\right)^n. \qquad [2]$$

Remark 4 (Best strategy). *The best strategy is post-ask when $p_f < 4/5$, and pre-ask when $p_f > 4/5$.*

7.3 Distance Fraud (White Box)

Early-reply strategy with one run. When the challenge is not void, the adversary can correctly respond to the verifier with probability 1 if $R_i^0 = R_i^1$, and with probability 1/2 if $R_i^0 \neq R_i^1$. Consequently:

$$\Pr_{\mathrm{DF|WB(1)|early}} = \left(1 - \frac{p_f}{4}\right)^n. \qquad [2]$$

Early-reply strategy with p runs. This strategy is efficient against the protocol if the verifier sends his nonce first. This weakness can be easily fixed though. The success probability is provided in [2]:

$$\Pr_{\mathrm{DF|WB(p)|early}} = \left((1 - p_f) + p_f \cdot \left(1 - \frac{1}{2} \cdot \frac{\mathbb{E}(d_H(v^0, v^1))}{n}\right)\right)^n$$

where $\mathbb{E}(d_H(R^0, R^1))$ is the expected minimum Hamming distance between R^0 and R^1 for the non-void challenges after the hash function is run p times with a different N_P . We have: $\lim_{p \to \infty}[\Pr_{\mathrm{DF|WB(p)|early}}] = 1$.

Circle strategy. Rounds being independent, the circle analysis offers no benefit to an adversary.

7.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. With the pre-ask strategy in the black box model, the adversary carries out an attack similar to mafia fraud but on its own device: $\Pr_{\mathrm{DF|BB|pre\&early}} \approx \Pr_{\mathrm{MF|pre}}$. The approximation is due to a small difference in the two frauds as explained hereafter. As long as the adversary is not detected by the device, she has the same strategy (and same probability of success) in both mafia fraud and distance fraud. In particular, she no longer receives useful information from the device once she is detected. However, in mafia fraud, she can still determine whether or not a round contains a void challenge when communicating with the verifier, as she does not have

time to get this information in distance fraud attacks. The difference is however negligible because she has to guess the final signature in both cases.

Post-ask combined with early-reply strategy. In this strategy, the adversary definitely obtains the correct final signature. However, she does not know when a void challenge or a non-void challenge is expected. Therefore, if the probability of a non-void challenge is lower (resp. higher) than 2/3 then her best strategy is to keep quiet (resp. try to guess every response, with probability 1/2).

$$\Pr_{\mathrm{DF|BB|post\&early}} = \begin{cases} \left(1-p_f\right)^n & \text{if } p_f < 2/3 \\ \left(\frac{p_f}{2}\right)^n & \text{if } p_f \geq 2/3 \end{cases} \quad [2]$$

Circle strategy. We previously stressed that the circle analysis is worthless for this protocol.

7.5 Terrorist Fraud (White Box)

This protocol is not designed to resist to terrorist fraud in the white box model. Indeed, the prover, without revealing his secret K, is able to provide to his accomplice the two registers required to successfully complete the protocol. We so have: $\Pr_{\mathrm{TF|WB}} = 1$ [2].

7.6 Published Attacks

A technique to reduce the required memory [54] consists in using only one $(n+1)$-bit register, where the responses are selected from the two edges. However, [2] demontrated that this technique opens the door to an attack where the adversary queries in advance the two values of the edges.

8 KIM, AVOINE, KOEUNE, STANDAERT AND PEREIRA'S PROTOCOL (2008)

Kim, Avoine, Koeune, Standaert and Pereira introduced a protocol in [50] known as the *Swiss-knife distance-bounding protocol*[2] (Algorithm 6). We only consider in this analysis the case where $T = 1$, that is when the protocol is not noise-resilient.

8.1 Impersonation

The attacker could impersonate the prover by guessing all the answers during the fast phase and T_B in the second slow phase. To succeed the adversary would need to guess $\sigma + n$ bits. Therefore, it is better for the adversary to guess K which size is σ. Consequently, we have: $\Pr_{\mathrm{Imp}} = \left(\frac{1}{2}\right)^{\sigma}$.

[2]Like Swiss-army knives used during WWII, the Swiss-knife protocol is a multi-purpose tool. The authors claim their protocol *"resists against both mafia fraud and terrorist attacks, reaches the best known false acceptance rate, preserves privacy, resists to channel errors, uses symmetric-key cryptography only, requires no more than 2 cryptographic operations to be performed by the tag, can take advantage of precomputation on the tag, and offers an optional mutual authentication"* [50].

Algorithm 6: Swiss-knife Protocol

Verifier		**Prover**
(secret K, constant C)		(secret K, identifier ID, constant C)
Pick $N_V \in_R \{0,1\}^\delta$		Pick $N_P \in_R \{0,1\}^\delta$
Pick a random D s.t. $\mathcal{H}(D) = n$		$a = f_K(C, N_P)$
		$Z^0 = a, Z^1 = a \oplus K$
	$\xrightarrow{N_V,\, D}$	
		For $i = 1$ to n: $R_i^0 = Z_j^0$ and $R_i^1 = Z_j^1$, where j is the index of the next "1" in the binary representation of D.
	$\xleftarrow{N_P}$	
Pick $c_i \in_R \{0,1\}$ and Start Timer	$\xrightarrow{c'_i}$	
Stop Timer	$\xleftarrow{r_i}$	$r_i = R_i^{c'_i}$
	$\xleftarrow{T_B,\, c'_1, \ldots, c'_n}$	$T_B = f_K(c'_1, \ldots, c'_n, ID, N_V, N_P)$
Check ID via DB		
Compute R^0, R^1		
Compute $err_C := \sharp\{i : c_i \neq c'_i\}$, $err_R := \sharp\{i : c_i = c'_i \wedge r_i \neq v_i{}^{c_i}\}$, $err_T := \sharp\{i : c_i = c'_i \wedge \Delta t_i > t_{\max}\}$		
If $err_C + err_R + err_T \geq T$, then REJECT		
$T_A = f_K(N_B)$	$\xrightarrow{T_A}$	Compute and check T_A

8.2 Mafia Fraud

Pre-ask strategy. The success probability of the pre-ask strategy is $\Pr_{\text{MF|pre}} = \left(\frac{1}{2}\right)^n$ [50].

Post-ask strategy. The adversary must guess the responses expected in the fast phase: $\Pr_{\text{MF|post}} = \left(\frac{1}{2}\right)^n$.

8.3 Distance Fraud (White Box)

Early-reply strategy with one run. As the prover can access to the internal state of the registers, she knows the content of the two registers. If $v_i^0 = v_i^1$, she always responds correctly, otherwise she has to guess the correct answer with probability $\frac{1}{2}$. Hence, $\Pr_{\text{DF|WB(1)|early}} = \left(\frac{3}{4}\right)^n$.

Early-reply strategy with p runs. The Swiss-knife Protocol generates only one register. Hence, multiple-run of PRF does not increase the adversary success probability: $\Pr_{\text{DF|WB(p)|early}} = \Pr_{\text{DF|WB(1)|early}}$.

Circle strategy. Rounds being independent, the circle analysis offers no benefit to an adversary.

8.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. The adversary sends her own challenges to the prover in advance. To obtain the correct signature in the second slow phase, the challenges sent by the adversary must be the same as the challenges sent by the verifier. We consequently have: $\Pr_{\text{DF|BB|pre\&early}} = \left(\frac{1}{2}\right)^n$.

Post-ask combined with early-reply strategy. The adversary has to correctly guess the response in each round. Hence: $\Pr_{\text{DF|BB|post\&early}} = \left(\frac{1}{2}\right)^n$.

Circle strategy. We previously stressed that the circle analysis is worthless for this protocol.

8.5 Terrorist Fraud (White Box)

Early-provide strategy with one run. $\Pr_{\text{TF|WB(1)}} = \left(\frac{3}{4}\right)^n$ [50].

Early-provide strategy with p runs. For the same reason given at Section 8.3: $\Pr_{\text{TF|WB(p)}} = \Pr_{\text{TF|WB(1)}}$.

8.6 Published Attacks

Peris-Lopez et al. proposed a passive full disclosure attack on the Swiss-knife RFID distance-bounding protocol [60]. However, their assumption is not correct: they assume that the size of the secret key (K) and random nonces (N_V and N_P) are equal to n (number of iterations in the fast phase) and n is insecurely short, for example 32 bits or less in the Swiss-knife protocol. Based on this assumption, they assert that the Swiss-knife protocol is insecure. The authors of the Swiss-knife RFID distance-bounding protocol never claimed that their protocol is secure when the size of the long-term key and random nonces are so short. Under this assumption, all the distance-bounding protocols can be broken.

9 AVOINE AND TCHAMKERTEN'S PROTOCOL (2009)

The protocol (Algorithm 7) introduced by Avoine and Tchamkerten in [7] is a generalization of Hancke and Kuhn's protocol that is more secure in terms of mafia and distance fraud.

9.1 Impersonation

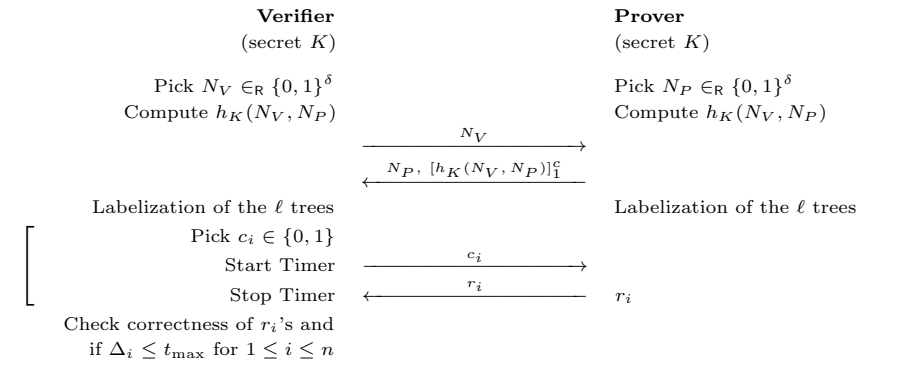
To impersonate a legitimate prover one needs to guess the c authentication bits and the n replies of the fast phase. Hence: $\Pr_{\text{Imp}} = \left(\frac{1}{2}\right)^{c+n}$.

9.2 Mafia Fraud

Pre-ask strategy. $\Pr_{\text{MF|pre}} = 2^{-n}\left(\frac{d}{2}+1\right)^{\frac{n}{d}} = 2^{-d\cdot\ell}\left(\frac{d}{2}+1\right)^{\ell} \quad \text{with} \quad n = d\ell$ [7].

Post-ask strategy. Without any final slow phase, a post-ask strategy is useless: $\Pr_{\text{MF|post}} = \left(\frac{1}{2}\right)^{c+n}$.

Algorithm 7: Tree-based Protocol



9.3 Distance Fraud (White Box)

Early-reply strategy with one run. The analysis of the distance fraud probability in the case of the tree-based protocol is very similar to the analysis of the Poulidor protocol (Section 11) that is provided in [69]. Unfortunately, this analysis only yields rough upper bounds. To find such an upper bound on the adversary success probability for distance fraud for the tree-based protocol, Theorem 3 available in [69] is used. This theorem is related to Poulidor but the only difference between Poulidor and the tree-based protocol is that the latter creates a full tree as graph. Therefore, the distance fraud success probability of the tree-based protocol is upper bounded by:

$$\frac{1}{2}\left(\frac{1}{2^n}+\sqrt{\frac{1}{2^{2n}}-\frac{4}{2^n}+4q}\right) \quad \text{where} \quad q=\prod_{i=1}^{i=n}\left(\frac{1}{2}+\frac{1}{2^{2i+1}}\sum_{k=0}^{k=2n-1}(A^i[0,k])^2\right).$$

The authors of [69] define $A^i[0,k]$ for a tree, considering that the nodes in the tree are labeled between 0 and 2^n-1 using a breadth-first algorithm, then:

$$A^i[0,k]=\begin{cases}1 & \text{if} \quad 2^i-1\leq k<2^{i+1}-1,\\ 0 & \text{otherwise,}\end{cases} \quad \text{and finally: } q=\prod_{i=1}^{i=n}\left(\frac{1}{2}+\frac{1}{2^{i+1}}\right).$$

Early-reply strategy with p runs. Similar to the Poulidor case (Section 11), this strategy makes sense for this protocol but so far neither $\Pr_{\text{DF|WB(1)|early}}$ nor $\Pr_{\text{DF|WB(p)|early}}$ have been calculated.

Circle strategy. Although it makes sense to consider the circle analysis for this protocol, the calculation of the distance fraud success probability in this scenario is also an open problem. Indeed, when the number of circles is greater than n, this problem is as hard as the calculation of $\Pr_{\text{DF|WB(1)|early}}$.

9.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. $\Pr_{\text{DF|BB|pre\&early}} = 2^{-n}(\frac{d}{2}+1)^{\frac{n}{d}}$ [7]

Post-ask combined with early-reply strategy. Note that the post-ask strategy will not allow the adversary to gain any information, i.e., $\Pr_{\mathrm{DF|BB|post\&early}} = \left(\frac{1}{2}\right)^{c+n}$.

Circle strategy. Since we are in a black box setting, the prover does not have access to the labeling of the trees, hence the circle strategy yields probability of success of $\left(\frac{1}{2}\right)^{n}$.

9.5 Terrorist Fraud (White Box)

This protocol is not designed to resist to terrorist fraud in the white box model. An attacker can reveal the tree node labelization to an accomplice who so successfully passes the fast phase with $\Pr_{\mathrm{TF|WB}} = 1$.

10 RASMUSSEN AND ČAPKUN'S PROTOCOL (2010)

The protocol (Algorithm 8) was introduced by Rasmussen and Čapkun and originally appeared in [63]. In this paper we consider the updated version that appeared in [61].

Algorithm 8: RC Protocol

Verifier		Prover
	$\xleftarrow{\mathrm{Commit}(N_P, ID_P)}$	Pick a random N_P
Pick a random N_V		
Start Timer	$\xrightarrow{N_V}$	
Stop Timer	$\xleftarrow{CRCS(N_V, N_P)}$	
From channels extract N'_P		Measure delay n
From signal extract N'_V		
From signal extract delay n'		
	$\xleftarrow{\mathrm{Sign}(M)}$	$M = \mathrm{Commit}(N_P, ID_P)\|\|$ $n\|\|ID_V\|\|N_P\|\|N_V$
Verify $\{\Delta t,\ n = n',$ $N'_V = N_V,\ N'_P = N_P,$ $\mathrm{Sign}(M)\}$		

10.1 Impersonation

We assume here that key size and nonce size are large enough to ensure that the probability of a key-recovery attack and a replay attack are negligible. The easiest manner to impersonate a prover is by forging the final signature. The success probability of this attack is: $\Pr_{\mathrm{Imp}} = \left(\frac{1}{2}\right)^{\sigma}$.

10.2 Mafia Fraud

Pre-ask strategy. In order to implement a mafia fraud attack using a pre-ask strategy an attacker has to guess the nonce N_V of the verifier. Otherwise the final signature will not be valid. So, $\Pr_{\mathrm{MF|pre}} = \left(\frac{1}{2}\right)^{\delta_V}$.

Post-ask strategy. An attacker wishing to execute a mafia fraud attack must guess all the bits of the prover's nonce in order to be able to reply correctly. Thus, $\Pr_{\mathrm{MF|post}} = \left(\frac{1}{2}\right)^{\delta_P}$.

10.3 Distance Fraud (White Box)

Early-reply strategy with one run. A malicious prover wishing to execute a distance fraud attack must guess all the bits of the verifier's nonce to reply correctly. Hence: $\Pr_{\mathrm{DF|WB(1)|early}} = \left(\frac{1}{2}\right)^{\delta_V}$ [61].

Early-reply strategy with p runs. The concept of round does not exist in this protocol, therefore: $\Pr_{\mathrm{DF|WB(p)|early}} = \Pr_{\mathrm{DF|WB(1)|early}}$.

Circle strategy. The concept of rounds does not exist in this protocol.

10.4 Distance Fraud (Black Box)

The security of this protocol does not depend on a well behaved prover. Consequently black-box success probabilities are the same as in the white-box model.

10.5 Terrorist Fraud (White Box)

This protocol is not designed to resist to terrorist fraud in the white box model. Indeed, the prover without revealing his secret K, is able to provide his accomplice with N_P, which is sufficient to successfully execute the fast phase. Hence, $\Pr_{\mathrm{TF|WB}} = 1$.

11 TRUJILLO-RASUA, MARTIN AND AVOINE'S PROTOCOL (2010)

Poulidor, the graph-based distance-bounding protocol (Algorithm 9) designed by Trujillo-Rasua, Martin, and Avoine [69], uses specific node and edge dependencies in the tree of the AT protocol [7] – which then can alternatively be represented by an acyclic graph. Poulidor benefits from a lower memory requirement compared to the AT protocol. Security is also reduced.

11.1 Impersonation

The common manner to impersonate a prover is by guessing all the answers during the fast phase. Hence, we have: $\Pr_{\mathrm{Imp}} = \left(\frac{1}{2}\right)^n$.

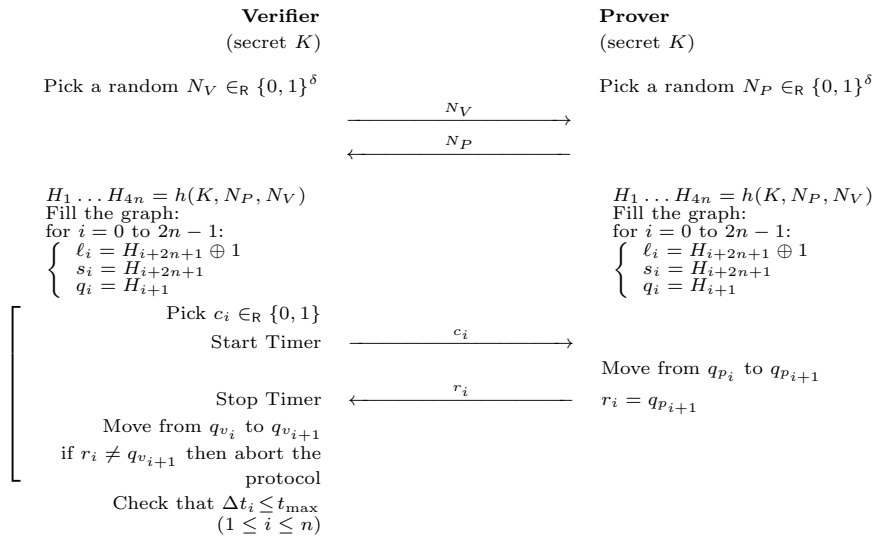
11.2 Mafia Fraud

Pre-ask strategy. Let:

$$g(i,j,k) = \frac{1}{2} + \frac{1}{2^{j+i-2k+2}} \times \sum_{t=0}^{t=2n-1} \left(A^{j-t}[1,t]A^{i-t}[2,t] + A^{j-k}[2,t]A^{i-k}[1,t]\right)$$

where A is the adjacency matrix of the graph which represents the graph-based protocol [69]. Also, let:

Algorithm 9: Poulidor Protocol



$$f(i,j,k) = \begin{cases} 1 & \text{if} \quad j < k \quad \text{and} \quad i = j, \\ \frac{1}{2} & \text{if} \quad j < k \quad \text{and} \quad i \neq j, \\ \frac{1}{2} & \text{if} \quad j \geq k \quad \text{and} \quad i < k, \\ g(i,j,k) & \text{if} \quad j \geq k \quad \text{and} \quad i \geq k. \end{cases}$$

We then have: $\Pr_{\text{MF|pre}} = \sum_{k=1}^{k=n} \frac{1}{2^k} \left(\prod_{j=k}^{j=n} \max(f(1,j,k), \cdots, f(n,j,k)) \right) + \frac{1}{2^n}$. [69]

Post-ask strategy. This protocol does not contain any second slow phase and the first slow phase consists of nonce exchanges only. As per Section 2 we have: $\Pr_{\text{MF|post}} = \Pr_{\text{Imp}}$.

11.3 Distance Fraud (White Box)

Early-reply strategy with one run. $\Pr_{\text{DF|WB(1)|early}}$ is upper bounded by [69]:

$$\frac{1}{2}\left(\frac{1}{2^n} + \sqrt{\frac{1}{2^{2n}} - \frac{4}{2^n} + 4q}\right) \text{ where } q = \prod_{i=1}^{i=n} \left(\frac{1}{2} + \frac{1}{2^{2i+1}} \sum_{k=0}^{k=2n-1} (A^i[0,k])^2 \right)$$

Computing in a similar way than in [69], we find the following relation for $A^i[0,k]$:

$$A^i[0,k] = \begin{cases} \binom{i}{k-i} & \text{if} \quad i \leq k \leq 2i, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and finally: } q = \prod_{i=1}^{i=n} \left(\frac{1}{2} + \frac{\binom{2i}{i}}{2^{2i+1}} \right).$$

Remark that finding an exact value for $\Pr_{\text{DF|WB(1)|early}}$ is an NP-hard problem [68].

Early-reply strategy with p runs. This strategy makes sense for this protocol, but so far, neither has been computed $\Pr_{\mathrm{DF|WB(1)|early}}$ nor can be computed $\Pr_{\mathrm{DF|WB(p)|early}}$.

Circle strategy. Although it makes sense to consider the circle analysis for this protocol, the calculation of the distance fraud success probability in this scenario is also an open problem.

11.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. With the pre-ask strategy, the adversary may learn the values of a walk in the graph. Note that, this is exactly the same knowledge obtained for an adversary attempting to perform a mafia fraud attack by using the pre-ask strategy. However, contrary to the mafia fraud attack, the adversary does not receive any challenge from the verifier when she is performing a distance fraud attack. We consequently have $\Pr_{\mathrm{DF|BB|pre\&early}} \leq \Pr_{\mathrm{MF|pre}}$. The equality of this equation holds when the adversary actually receives every challenge before sending its corresponding response, i.e., when the adversary is in the close vicinity of the verifier. Therefore, for this protocol the circle strategy makes sense. The closer to the verifier the adversary is, the higher her probability of success is, but it is still upper-bounded by $\Pr_{\mathrm{MF|pre}}$.

Post-ask combined with early-reply strategy. $\Pr_{\mathrm{DF|BB|post\&early}} = \Pr_{\mathrm{Imp}}$.

Circle strategy. As explained above, the circle strategy makes sense for this protocol. Nevertheless, the adversary's success probability by using this strategy is upper-bounded by $\Pr_{\mathrm{MF|pre}}$.

11.5 Terrorist Fraud (White Box)

This protocol is not designed to resist to terrorist fraud in the white box model. Indeed, the prover is able, without revealing his secret K, to provide his accomplice with the graph required to successfully pass thorough the protocol. Hence: $\Pr_{\mathrm{TF|WB}} = 1$.

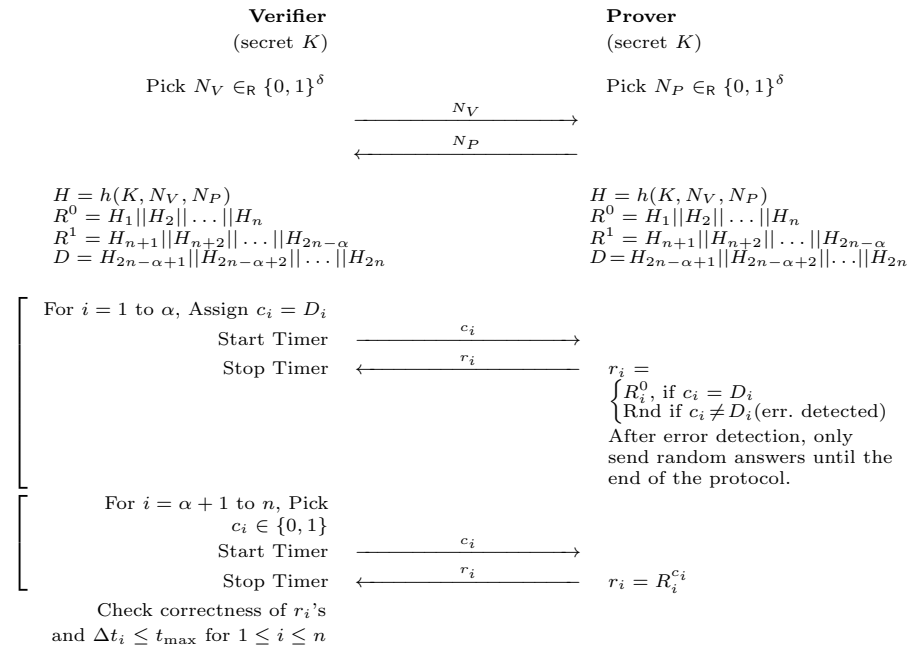
12 KIM AND AVOINE'S PROTOCOL (KA2) (2011)

Kim and Avoine introduced in 2009 a distance-bounding protocol with mixed challenges [48], namely challenges known and challenges unknown in advance by the prover. Challenges known in advance allow the prover to help the verifier to detect an attack, but these challenges also allow the prover to succeed in performing a distance fraud. Kim and Avoine improved their protocol in 2011, yielding a new variant known as KA2 [49], which is analyzed in this section (Algorithm 10).

12.1 Impersonation

Guessing the n answers r_i is enough to impersonate the prover: $\Pr_{\mathrm{Imp}} = \left(\frac{1}{2}\right)^n$.

Algorithm 10: KA2 Protocol



12.2 Mafia Fraud

Pre-ask strategy. $\Pr_{\text{MF}|\text{pre}} = \left(\frac{3}{4}\right)^{n-\alpha}\left(\frac{1}{2}\right)^{\alpha} + \alpha\left(\frac{1}{2}\right)^{n+1}$ [49].

Post-ask strategy. This protocol does not contain any second slow phase and the first slow phase consists of nonce exchanges only. As per Section 2 we have: $\Pr_{\text{MF}|\text{post}} = \Pr_{\text{Imp}}$.

12.3 Distance Fraud (White Box)

Early-reply strategy with one run. $\Pr_{\text{DF}|\text{WB}(1)|\text{early}} = \left(\frac{3}{4}\right)^{n-\alpha}$ [49].

Early-reply strategy with p runs. The success probability in case of early-reply strategy with p runs of the pseudo-random function is provided in [2]:

$$\frac{1}{2^{p(n-\alpha)}} \cdot \left(\sum_{i=0}^{i=n-\alpha-1} \left(\frac{1}{2}\right)^i \cdot \left[\left(\sum_{j=i}^{j=n-\alpha} \binom{n-\alpha}{j} \right)^p - \left(\sum_{j=i+1}^{j=n-\alpha} \binom{n-\alpha}{j} \right)^p \right] + \left(\frac{1}{2}\right)^n \right).$$

Circle strategy. Rounds being independent, the circle analysis offers no benefit to an adversary.

12.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. As with mafia fraud with pre-ask strategy, the success probability is $\Pr_{\text{DF}|\text{BB}|\text{pre\&early}} = \Pr_{\text{MF}|\text{pre}}$.

Post-ask combined with early-reply strategy. This protocol does not contain any second slow phase and the first slow phase consists of nonce exchanges only. As per Section 2 we have: $\Pr_{\mathrm{DF|BB|post\&early}} = \Pr_{\mathrm{Imp}}$.

Circle strategy. We previously stressed that the circle analysis is worthless for this protocol.

12.5 Terrorist Fraud (White Box)

This protocol is not designed to resist to terrorist fraud in the white box model. The prover can give the registers to his accomplice to successfully pass the protocol: $\Pr_{\mathrm{TF|WB}} = 1$.

13 YUM, KIM, HONG AND LEE'S PROTOCOL (2010)

Yum, Kim, Hon and Lee created a distance-bounding protocol with mutual authentication [76].

Algorithm 11: YKHL Protocol

User A		**User B**
(secret K)		(secret K)
Pick $N_A \in_{\mathsf{R}} \{0,1\}^\delta$		Pick $N_B \in_{\mathsf{R}} \{0,1\}^\delta$
	$\xrightarrow{N_A}$	
	$\xleftarrow{N_B}$	
$H = h(K, N_A, N_B)$ $D = H_1 \quad \|\|H_2 \quad \|\| \ldots \|\|H_n$ $R^0 = H_{n+1}\|\|H_{n+2}\|\| \ldots \|\|H_{2n}$ $R^1 = H_{2n+1}\|\|H_{2n+2}\|\| \ldots \|\|H_{3n}$		$H = h(K, N_A, N_B)$ $D = H_1 \quad \|\|H_2 \quad \|\| \ldots \|\|H_n$ $R^0 = H_{n+1}\|\|H_{n+2}\|\| \ldots \|\|H_{2n}$ $R^1 = H_{2n+1}\|\|H_{2n+2}\|\| \ldots \|\|H_{3n}$
Case I: $D_i = 0$ Pick $c_i \in \{0,1\}$		
Start Timer	$\xrightarrow{c_i}$	
Stop Timer	$\xleftarrow{r_i}$	$r_i = \begin{cases} R_i^0, \text{ if } c_i = 0 \\ R_i^1, \text{ if } c_i = 1 \end{cases}$
If $r_i \neq R_i^{c_i}$ or a collision is detected,		If a collision is detected B enters into the protection mode.
If $r_i \neq R_i^{c_i}$ or a collision is detected,		
Case II: $D_i = 1$		Pick $c_i \in \{0,1\}$
	$\xleftarrow{c_i}$	Start Timer
$r_i = \begin{cases} R_i^0, \text{ if } c_i = 0 \\ R_i^1, \text{ if } c_i = 1 \end{cases}$	$\xrightarrow{r_i}$	Stop Timer
If a collision is detected, A enters into the protection mode.		If $r_i \neq R_i^{c_i}$ or a collision is detected, B enters into the protection mode.
Check correctness of r_i's and $\Delta t_i \leq t_{\max}$ for Case I		Check correctness of r_i's and $\Delta t_i \leq t_{\max}$ for Case II

13.1 Impersonation

The only known way to succeed at the impersonation consists of guessing all the answers during the fast phase, which leads to the probability $\Pr_{\mathrm{Imp}} = \left(\frac{1}{2}\right)^n$.

13.2 Mafia Fraud

Pre-ask strategy. Avoine and Kim proposed a new attack that yields a higher adversary success probability [4]. Their attack depends on the probability of finding D_i's, $\Pr_D$, which varies according to the system parameters. Following this attack, the success probability of a mafia fraud attack is at least:

$$\Pr_{\mathrm{MF|pre}} = \begin{cases} (\frac{5}{8})^n + \sum_{i=1}^{n} \frac{1}{4} \cdot (\frac{5}{8})^{i-1} \cdot (\frac{3}{4})^{n-i}, \text{if} \Pr_D = 1 \\ (\frac{1}{2})^n + \sum_{i=1}^{n} \frac{3}{8} \cdot (\frac{1}{2})^{i-1} \cdot (\frac{5}{8})^{n-i}, \text{if} \Pr_D = \frac{3}{4} \end{cases} . \quad [4]$$

Post-ask strategy. This protocol does not contain any second slow phase and the first slow phase consists of nonce exchanges only. As per Section 2 we have: $\Pr_{\mathrm{MF|post}} = \Pr_{\mathrm{Imp}}$.

13.3 Distance Fraud (White Box)

Early-reply strategy with one run. $\Pr_{\mathrm{DF|WB(1)|early}} = \left(\frac{7}{8}\right)^n$ [76].

Early-reply strategy with p runs. An attacker who impersonates B wins when $D_i = 1$ or $R_i^0 = R_i^1$. Indeed, when $D_i = 1$, the prover sends a challenge to the verifier (A) and so trivially wins the round. When $D_i = 0$, the roles are inverted. To win a round, the prover must send his response in advance. When $R_i^0 = R_i^1$, the potential answers are the same and the prover definitely wins. Running the cryptographic function p times allows the prover to find D with a higher Hamming weight than the average one, and R^0 and R^1 with a lower Hamming distance. In conclusion, the probability of success is higher with the YKHL Protocol than with the HK protocol, where the prover wins only when $R_i^0 = R_i^1$. The probability of success can be calculated by considering $\Pr(X = x) = \binom{n}{x}(\frac{1}{4})^x(\frac{3}{4})^{n-x}/2^{3n}$ instead of $\Pr(X = x) = \binom{n}{x}/2^n$ in [2].

Circle strategy. Rounds being independent, the circle analysis offers no benefit to an adversary.

13.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. A distance fraud attack with the pre-ask strategy is similar to a mafia fraud in this case. Hence: $\Pr_{\mathrm{DF|BB|pre\&early}} = \Pr_{\mathrm{MF|pre}}$.

Post-ask combined with early-reply strategy. This protocol does not contain any second slow phase and the first slow phase consists of nonce exchanges only. As per Section 2 we have: $\Pr_{\mathrm{DF|BB|post\&early}} = \Pr_{\mathrm{Imp}}$.

Circle strategy. We previously stressed that the circle analysis is worthless for this protocol.

13.5 Terrorist Fraud (White Box)

This protocol is not designed to resist to terrorist fraud in the white box model. Indeed, the prover can give the registers to his accomplice to successfully pass the protocol: $\Pr_{\mathrm{TF|WB}} = 1$.

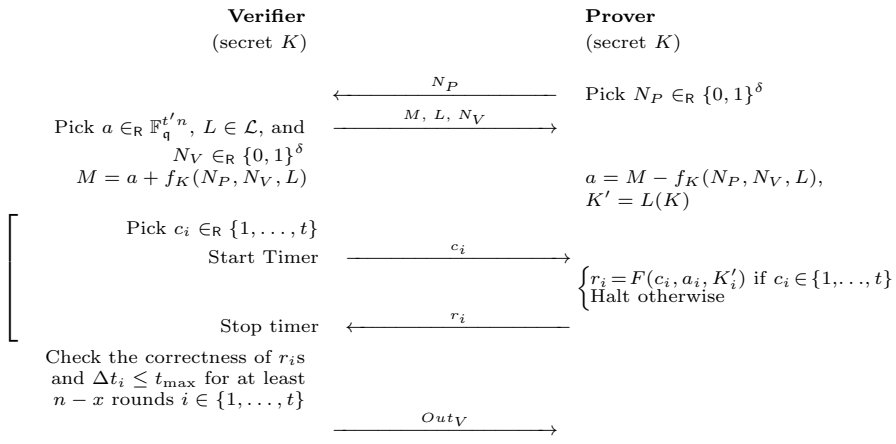
13.6 Published Attacks

Avoine and Kim demonstrated in [4] that the security of YKHL protocol is far below what is claimed in [76] and could be worse than the HK protocol w.r.t. mafia fraud resistance.

14 SKI PROTOCOLS (2013)

In [14–16], the authors introduced a series of protocols called *SKI*. These protocols are presented in Algorithm 12.

Algorithm 12: The **SKI** Protocols



14.1 Impersonation

The only known way to do this attack is to guess all the answers during the fast phase. So: $\Pr_{\text{Imp}} = \left(\frac{1}{\mathfrak{q}}\right)^n$.

14.2 Mafia Fraud

Pre-ask strategy. With this strategy, the adversary can obtain one set of answers from the prover before executing the fast phase with the verifier. Without loss of generality, we assume that he obtains $\{F(1, a_1, K'_1), \ldots, F(1, a_n, K'_n)\}$, i.e., the answers corresponding to the challenges c_i's equal to 1. Hence, at each rounds two cases occur: (a) the verifier's challenge is 1 and she knows the answer, this happens with probability $1/t$, or (b) the verifier's challenge is not 1, thus she has to guess the answer, and succeeds with probability $1/\mathfrak{q}$. Thus, the rounds independence yields to:

$\Pr_{\text{MF|pre}} = \left(\frac{1}{t} \cdot 1 + (1 - \frac{1}{t}) \cdot \frac{1}{\mathfrak{q}}\right)^n = \left(\frac{\mathfrak{q} + t - 1}{\mathfrak{q}t}\right)^n$. For SKI_{pro}, this is $\left(\frac{2}{3}\right)^n$.

Post-ask strategy. This protocol does not contain any second slow phase and the first slow phase consists of nonce exchanges only. As per Section 2 we have: $\Pr_{\text{MF|post}} = \Pr_{\text{Imp}}$.

14.3 Distance Fraud (White Box)

Early-reply strategy with one run. Using this strategy, the adversary has to send her answers in advance. Due to the similarity between Hancke and Kuhn's protocol and SKI's protocol, the adversary applies a similar strategy to maximize her success probability. At each rounds she answers the most probable value among the possible registers. The most probable answer for a given round is the one that appears the most within the set $\{F(1, a_i, K'_i), F(2, a_i, K'_i), \ldots, F(t, a_i, K'_i)\}$ of possible answers for this round. In order to compute the adversary success probability, let define the following events:

- $\mathcal{W}$: the adversary provides the correct answer to the verifier at a given round.
- $\mathcal{B}_j$: $j = \max\limits_{1 \leq l \leq \mathfrak{q}} \{X_l\}$,

where X_l is the number of appearance times of the l-th element from $\mathbb{F}_\mathfrak{q}$ among the set $\{F(1, a_i, K'_i), F(2, a_i, K'_i), \ldots, F(t, a_i, K'_i)\}$. We then trivially have: $\Pr(\mathcal{W}) = \sum_{j=1}^{j=t} \Pr(\mathcal{W}|\mathcal{B}_j) \Pr(\mathcal{B}_j)$, with $\Pr(\mathcal{W}|\mathcal{B}_i) = \frac{i}{t}$. Thus, using the above equation, we deduce: $\Pr(\mathcal{W}) = \mathbb{E}\left(\max\limits_{1 \leq l \leq \mathfrak{q}} \{X_l\}\right) \cdot \frac{1}{t}$. The tricky task consists in computing $\mathbb{E}(\max\limits_{1 \leq l \leq \mathfrak{q}} \{X_l\})$. This is done below for the **SKI** protocol configurations suggested in [14].

- $\mathfrak{q} = 2$, and $t = 2$: $\mathbb{E}(\max\limits_{1 \leq l \leq \mathfrak{q}} \{X_l\}) = \frac{3}{2}$, and $\Pr(\mathcal{W}) = \frac{3}{4}$.
- $\mathfrak{q} = 2$, and $t = 3$: $\mathbb{E}(\max\limits_{1 \leq l \leq \mathfrak{q}} \{X_l\}) = \frac{9}{4}$ and $\Pr(\mathcal{W}) = \frac{3}{4}$.
- $\mathfrak{q} = 2$, and $t = 4$: $\mathbb{E}(\max\limits_{1 \leq l \leq \mathfrak{q}} \{X_l\}) = 3$ and $\Pr(\mathcal{W}) = \frac{3}{4}$.
- $\mathfrak{q} = 4$, and $t = 3$: $\mathbb{E}(\max\limits_{1 \leq l \leq \mathfrak{q}} \{X_l\}) = \frac{15}{8}$ and $\Pr(\mathcal{W}) = \frac{5}{8}$.

Finally, the independence of the rounds provides $\Pr_{\mathrm{DF|WB(1)|early}} = (\Pr(\mathcal{W}))^n$. For $\mathrm{SKI}_{\mathrm{pro}}$, this is $\left(\frac{3}{4}\right)^n$.

Early-reply strategy with p runs. This strategy does not make sense against these protocols. Indeed, since the prover does not have the verifier's nonce before he sends its, he cannot compute several outputs of the the pseudo-random function.

Circle strategy. Rounds being independent, the circle analysis offers no benefit to an adversary.

14.4 Distance Fraud (Black Box)

Pre-ask combined with early-reply strategy. Distance fraud with the pre-ask strategy is here similar to mafia fraud. Hence: $\Pr_{\mathrm{DF|BB|pre\&early}} = \Pr_{\mathrm{MF|pre}}$.

Post-ask combined with early-reply strategy. This protocol does not contain any second slow phase and the first slow phase consists of nonce exchanges only. As per Section 2 we have: $\Pr_{\mathrm{DF|BB|post\&early}} = \Pr_{\mathrm{Imp}}$.

Circle strategy. We previously stressed that the circle analysis is worthless for this protocol.

14.5 Terrorist Fraud (White Box)

Early-provide strategy with one run. Using this strategy, the adversary obtains register(s) before the start of the fast phase. First, note that the setting in which $t' = t = \mathfrak{q} = 2$ (i.e., SKI_{lite}) does not resist against terrorist fraud. Second, to compute the success probability in the other cases, let denote k, the number of registers given by the prover to the adversary. As stated in [5], the insurance that no information could leak, is furnished by the following equality: $k = t - 2$.

Once the adversary gets the $t - 2$ registers, she starts the fast phase with the verifier. Two cases occur, (a) the verifier asks an answer coming from one of the $t - 2$ known registers. Thus, the adversary definitely knows the correct answer. Or (b) the verifier asks her an answer coming from one the two unknown registers, and she has to guess the correct answer. The adversary consequently succeeds with probability $\frac{1}{\mathfrak{q}}$. Given the rounds are independent, we finally have:

$\Pr_{\text{TF|WB(1)}} = \left(\frac{t-2}{t} \cdot 1 + \frac{2}{t} \cdot \frac{1}{\mathfrak{q}}\right)^n = \left(\frac{\mathfrak{q}t + 2(1-\mathfrak{q})}{\mathfrak{q}t}\right)^n$. For SKI_{pro}, this is $\left(\frac{2}{3}\right)^n$.

Early-provide strategy with p runs. This strategy does not make sense against these protocols. Indeed, since the prover does not have the verifier's nonce before he sends its answers, he cannot compute several outputs of the the pseudo-random function.

15 PROTOCOL COMPARISON

This section provides a summary of the analyses done in Sections 3 to 14. It then provides two approaches to compare the protocols: the first one consists of charts, while the second one is based on the concept of clusters. The charts depict the variation of a single parameter regarding another one, e.g., the mafia fraud success probability as a function of the number of rounds. The second approach introduces clusters of protocols sharing common security resistances and properties. It is worth remarking that a similar comparison approach based on decision theory has been recently published in [6]. The findings of that work do not contradict ours; indeed every protocol found relevant there is also considered relevant here.

15.1 Summary of Properties and Performance

Table 5 presents the properties and performance of every protocol analyzed through Sections 3 to 14. The description of the properties is provided in Section 2. Table 6 and 7 summarize which cryptographic building blocks are used and which properties are expected by each considered protocol. Greyed cells in Table 7 contain results already known, while other cells contain values provided by this survey.

On Table 7, we can see that only two protocols do not have any attack with probability 1: Swiss-knife and SKI. In fact, there exists two other protocols which are not in this table: TDB [5] (on which SKI is based) and the protocol by [34] (which is based on Swiss-knife).

15.2 Chart-based Comparison

Figure 2 depicts the mafia fraud success probability as a function of the number of rounds. The relative positions between the curves remain unchanged when the number of rounds increases, except for the tree-based protocol (with $l = \sqrt{n}$), which suffers from a step effect due to its structure. Several behaviors are observed in the figure: BC, for example, has a success probability of $(1/2)^n$, which is the optimal case, while HK, by contrast, has a $(3/4)^n$ success probability. The extreme case would be probability equals to 1 but no protocol falls into this category. Other intermediate behaviors are also present: protocols whose associated probability is not $(1/2)^n$ but tends to $(1/2)^n$ when n is large enough (e.g., AT($\sqrt{n}$)), protocols whose associated probability is between $(1/2)^n$ and $(3/4)^n$, and finally those whose associated probability is between $(3/4)^n$ and 1. These categories are summarized in Table 8.

Figure 3 represents the distance fraud success probability as a function of the number of rounds. As previously, several behaviors can be distinguished; reported in Table 8. The worst protocols in terms of distance fraud share a common mechanism where the prover helps the verifier to detect mafia fraud. As a consequence, the prover knows (at least partially) the expected challenges, which unfortunately helps him in mounting a successful distance fraud attack. The best protocols in terms of distance fraud have a final slow phase. The best ones without final slow phase have dependent rounds (see Section 18.4), namely the tree-based protocol (with $l = 1$ and with $l = \sqrt{n}$). Note that the tree-based protocol and Poulidor do not have close formulas to express the associated distance fraud success probability.

Figure 4 presents the memory needed to store intermediate values during the execution of the protocol, including the registers. The memory consumption is expressed as a function of the number of rounds. The curves can be classified into three categories: linear curves, affine curves which are not linear due to a fixed overhead, and non-affine curves. In the latter case, which includes the tree-based protocol (with $l = 1$ and with $l = \sqrt{n}$), the memory consumption is prohibitive. The overhead appears in the protocols that end with a final slow phase (Swiss-knife, MAD, and RC). In such a phase, the value of the challenges, or commitments used in the first slow phase, are usually stored all along the protocol execution because they are required for the final cryptographic operations. A final slow phase is consequently a handicap for implementations.

Figure 5 represents the mafia fraud success probability as a function of the distance fraud success probability, with the number of rounds n equal to 36. The figure clearly shows that the protocols are more resistant to mafia fraud than to distance fraud. Several reasons could probably explain this phenomenon. In particular, the original objective of distance bounding was – early in the nineties – to protect authentication protocols against relay attacks. Distance fraud was then a side effect of distance bounding. It has been only recently, when the need to protect geolocalisation applications against distance fraud arose, especially when the prover is a mobile device, that distance fraud has started to be considered seriously.

Table 5. Properties and performance

Protocol	Adaptiveness	Mutual Authentication	Second Slow Phase	Independence of the Rounds	Exchanged Bits during Slow Phase	Exchanged Bits during Fast Phase	Memory
BC	No	No	Yes	Yes	$2\ell + n$	$2n$	$2n$
MAD	No	Yes	Yes	No	$2(\ell + \delta + \sigma)$	$2n$	$2n$
HK	No	No	No	Yes	2δ	$2n$	$2n$
MP	No	Yes	Yes	Yes	$2\delta + 3n$	$2(n - n_{void})$	$3n$
Swiss-knife	No	Option	Yes	Yes	$2(\delta + \sigma) + n$ ($+\sigma$ if mutual)	$2n$	$3n + 2\delta$
Tree-based	No	No	No	No iff $d \geq 2$	$2\delta + c$	$2n$	$\ell(2^{d+1} - 2)$
RC	No	No	Yes	No rounds	σ	$2\delta_V$	$\delta_V + \delta_P$
Poulidor	No (*)	No	No	No	2δ	$2n$	$4n$
KA2	Yes	Yes	No	Yes	2δ	$2n$	$2n$
YKHL	No	Yes	No	Yes	2δ	$2n$	$3n$
SKI_{pro}	No	No	No	Yes	$2\delta + 2n$	$3n$	$2n$

(*) See Section 11 for a refined analysis about the adaptiveness.

Table 6. Cryptographic building blocks

Protocol	PRNG	Sym. Primitive	Commitment	Signature
BC	Yes		Yes	Yes
MAD	Yes	Yes		
HK	Yes	Yes		
MP	Yes	Yes		
Swiss-knife	Yes	Yes		
Tree-based	Yes	Yes		
RC	Yes		Yes	Yes
Poulidor	Yes	Yes		
KA2	Yes	Yes		
YKHL	Yes	Yes		
SKI	Yes	Yes		

15.3 Cluster-based Comparison

Comparing distance bounding protocols is quite a tricky task given the large number of parameters that can be considered. A given protocol P_1 can be better in terms of resistance against mafia fraud than another protocol P_2, but at the same time worse in terms of resistance against distance fraud. Thus, ranking P_1 and P_2 is very complicated. This section introduces a hierarchical clustering of the distance bounding protocols. The key-point of the method relies on the observation that a protocol P_1 is undeniably better than a protocol P_2 if and only if P_1 is better than P_2 for every considered parameter. In such a case, P_1 should be used, instead of P_2, whatever the considered scenario.

Table 7. Adversary success probabilities

Protocol	Imp	Mafia		Distance				Terrorist	
		pre-ask	post-ask	early-reply		pre & early	post & early	early-provide	
		BB	BB	WB(1)	WB(p)	BB	BB	WB(1)	WB(p)
BC	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	1	1
MAD	$\left(\frac{1}{2}\right)^\sigma$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	1	1
HK	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^n$	Sect. 5.3	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^n$	1	1
MP	$\left(1-\frac{p_f}{2}\right)^{n}$ $\times\left(\frac{1}{2}\right)^{\gamma n}$	$\left(1-p_f\right)^n$ if $p_f<\frac{4}{7}$ $\left(p_f\cdot\frac{3}{4}\right)^n$ if $p_f\geq\frac{4}{7}$	$\left(1-\frac{p_f}{2}\right)^n$	$\left(1-\frac{p_f}{4}\right)^n$	$\left(\frac{(1-p_f)+p_f\cdot\left(1-\frac{\mathbb{E}(d_H(\hat{v}^0,v^1))}{2n}\right)}{2}\right)^n$	$\left(1-p_f\right)^n$ if $p_f<\frac{2}{3}$ $\left(\frac{p_f}{2}\right)^n$ if $p_f\geq\frac{2}{3}$	$\left(1-p_f\right)^n$ if $p_f<\frac{4}{7}$ $\left(p_f\cdot\frac{3}{4}\right)^n$ if $p_f\geq\frac{4}{7}$	1	1
Swiss-knife	$\left(\frac{1}{2}\right)^\sigma$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^n$	$\left(\frac{3}{4}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^n$	$\left(\frac{3}{4}\right)^n$
Tree-based	$\left(\frac{1}{2}\right)^{c+n}$	$2^{-d\ell}(d/2+1)^\ell$	$\left(\frac{1}{2}\right)^n$	Sect. 9.3	Open Prob	$\left(\frac{1}{2}\right)^{c+n}$	$2^{-n}(d/2+1)^{n/d}$	1	1
RC	$\left(\frac{1}{2}\right)^\sigma$	$\left(\frac{1}{2}\right)^{\delta V}$	$\left(\frac{1}{2}\right)^{\delta P}$	$\left(\frac{1}{2}\right)^{\delta V}$	$\left(\frac{1}{2}\right)^{\delta V}$	$\left(\frac{1}{2}\right)^{\delta V}$	$\left(\frac{1}{2}\right)^{\delta V}$	1	1
Poulidor	$\left(\frac{1}{2}\right)^n$	Sect. 11.2	$\left(\frac{1}{2}\right)^n$	Sect. 11.3	Open Prob	$\left(\frac{1}{2}\right)^n$	$\leq \Pr_{\mathrm{MF}\mid\mathrm{pre}}$	1	1
KA2	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^{n-\alpha}\left(\frac{1}{2}\right)^\alpha+$ $\alpha\left(\frac{1}{2}\right)^{n+1}$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^{n-\alpha}$	Sect. 12.3	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^{n-\alpha}\left(\frac{1}{2}\right)^\alpha+$ $\alpha\left(\frac{1}{2}\right)^{n+1}$	1	1
YKHL	$\left(\frac{1}{2}\right)^n$	Sect. 13.2	$\left(\frac{1}{2}\right)^n$	$\left(\frac{7}{8}\right)^n$	Sect. 13.3	$\left(\frac{1}{2}\right)^n$	Sect. 13.4	1	1
$\mathrm{SKI}_{\mathrm{pro}}$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{2}{3}\right)^n$	$\left(\frac{1}{2}\right)^n$	Sect. 14.3	Sect. 14.3	$\left(\frac{2}{3}\right)^n$	$\left(\frac{1}{2}\right)^n$	$\left(\frac{2}{3}\right)^n$	$\left(\frac{2}{3}\right)^n$

Seven parameters are considered for the cluster-based comparison: mafia fraud resistance ($\mathcal{M}a$), terrorist fraud resistance ($\mathcal{T}$), distance fraud resistance ($\mathcal{D}$), the presence of a final slow phase ($\mathcal{L}$), single bit exchanges during the fast phase

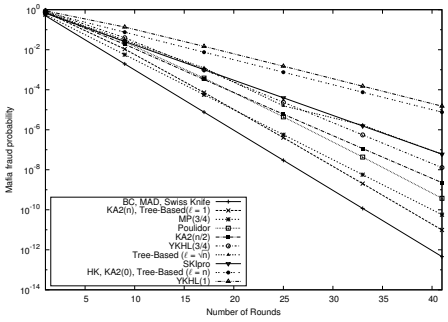


Fig. 2. $\Pr_{\mathrm{MF}|\mathrm{pre}}$

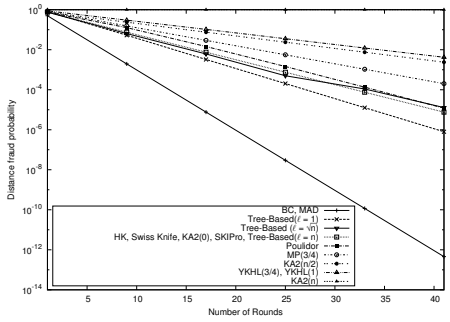


Fig. 3. $\Pr_{\mathrm{DF}|\mathrm{WB}(1)|\mathrm{early}}$

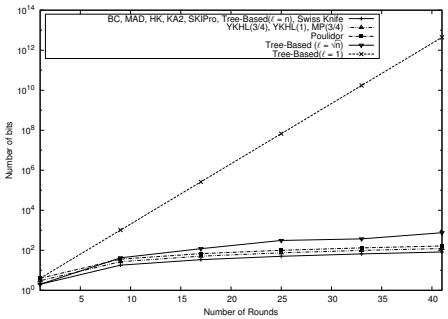


Fig. 4. Memory consumption

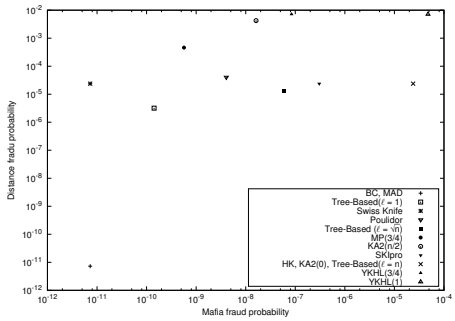


Fig. 5. $\Pr_{\mathrm{MF}|\mathrm{pre}}$ vs. $\Pr_{\mathrm{DF}|\mathrm{WB}(1)|\mathrm{early}}$ (n=36)

Table 8. Parameters and their values

	$\mathcal{M}a$	$\mathcal{T}$	$\mathcal{D}$	$\mathcal{L}$	$\mathcal{B}$	$\mathcal{E}$	$\mathcal{M}e$
Val 1	$\mathsf{ma} = \left(\frac{1}{2}\right)^n$	$\mathsf{t} = \left(\frac{1}{2}\right)^n$	$\mathsf{d} = \left(\frac{1}{2}\right)^n$	N	Y	$\mathsf{e} = 2n + \mathrm{cst}$	linear
Val 2	$\lim_{n\to\infty} \mathsf{ma} = \left(\frac{1}{2}\right)^n$	$\lim_{n\to\infty} \mathsf{t} = \left(\frac{1}{2}\right)^n$	$\lim_{n\to\infty} \mathsf{d} = \left(\frac{1}{2}\right)^n$	Y	N	$\mathsf{e} = 3n + \mathrm{cst}$	affine
Val 3	$\left(\frac{1}{2}\right)^n < \mathsf{ma} < \left(\frac{3}{4}\right)^n$	$\left(\frac{1}{2}\right)^n < \mathsf{t} < \left(\frac{3}{4}\right)^n$	$\left(\frac{1}{2}\right)^n < \mathsf{d} < \left(\frac{3}{4}\right)^n$			$\mathsf{e} \geq 4 \cdot n$	non affine
Val 4	$\mathsf{ma} = \left(\frac{3}{4}\right)^n$	$\mathsf{t} = \left(\frac{3}{4}\right)^n$	$\mathsf{d} = \left(\frac{3}{4}\right)^n$				
Val 5	$\left(\frac{3}{4}\right)^n < \mathsf{ma} < 1$	$\left(\frac{3}{4}\right)^n < \mathsf{t} < 1$	$\left(\frac{3}{4}\right)^n < \mathsf{d} < 1$				
Val 6	$\mathsf{ma} = 1$	$\mathsf{t} = 1$	$\mathsf{d} = 1$				

($\mathcal{B}$), the number of bits exchanged by the two parties during the whole protocol[3] ($\mathcal{E}$), and the memory dependency on the prover side regarding the number of rounds ($\mathcal{M}e$). Note the implementation complexity is not considered as it would be difficult to find a technology suitable to implement fairly all the protocols (some protocols depend on a given technology).

[3]Note that in Table 8, no value is given to the constant. Since we are interested in how the number of exchanged bits scales the number of rounds, the actual value of the constant does not really matter.

It is worth mentioning that each parameter can be assigned with a value belonging to an (obviously) ordered set. The method, for example, sorts the resistance to the mafia fraud $\mathcal{M}a$ according to six values and, to illustrate the concept, it is clearly better for a protocol to have a mafia fraud success probability equal to $(\frac{1}{2})^n$ than $(\frac{3}{4})^n$. Also, it is better not to need a final slow phase in the protocol, and to use binary messages than ternary messages, etc. For each parameter, the values can be ordered: (Value 6) $\prec$ (Value 5) $\prec$ (Value 4) $\prec$ (Value 3) $\prec$ (Value 2) $\prec$ (Value 1), where (Value i) $\prec$ (Value j) means that (Value j) is better than (Value i) or, said differently, (Value j) is more convenient than (Value i) when implementing a distance bounding protocol. The parameters and their values are provided in Table 8.

The *configuration* of a protocol is an element of the cartesian product $\mathcal{M}a \times \mathcal{T} \times \mathcal{D} \times \mathcal{L} \times \mathcal{B} \times \mathcal{E} \times \mathcal{M}e$. All the possible configurations can be deduced from Table 8, and the configuration of each protocol presented in this work is provided in Table 9. Note that there is no total order relation in $\mathcal{M}a \times \mathcal{T} \times \mathcal{D} \times \mathcal{L} \times \mathcal{B} \times \mathcal{E} \times \mathcal{M}e$, but a configuration $(\mathsf{ma}, \mathsf{d}, \mathsf{t}, \mathsf{l}, \mathsf{b}, \mathsf{e}, \mathsf{me})$ is better than a configuration $(\mathsf{ma}', \mathsf{d}', \mathsf{t}', \mathsf{l}', \mathsf{b}', \mathsf{e}', \mathsf{me}')$ if it is better for every considered parameter: $\mathsf{ma}' \prec \mathsf{ma}$, $\mathsf{d}' \prec \mathsf{d}$, ..., $\mathsf{me}' \prec \mathsf{me}$. As a consequence, there exist several *best* configurations in $\mathcal{M}a \times \mathcal{T} \times \mathcal{D} \times \mathcal{L} \times \mathcal{B} \times \mathcal{E} \times \mathcal{M}e$.

A *cluster* is a set (possibly empty) of protocols which have the same configuration $(\mathsf{ma}, \mathsf{d}, \mathsf{t}, \mathsf{l}, \mathsf{b}, \mathsf{e}, \mathsf{me})$. A cluster is said to be better than another one if its configuration is better. The total number of clusters is large; it is actually equal to the cardinalility of $\mathcal{M}a \times \mathcal{T} \times \mathcal{D} \times \mathcal{L} \times \mathcal{B} \times \mathcal{E} \times \mathcal{M}e$, which is $6^3 \cdot 2^2 \cdot 3^2 = 7776$. However, 5774 clusters are empty, meaning that no protocol matches the configuration of these clusters. The remaining 2002 non-empty clusters still represent a large amount of information, which is difficult to condense in a paper. To further reduce this information, only *best* configurations are kept.

This process can be easily automated. A hierarchy of clusters is built and the best cluster of every branch is kept. After performing this operation, only 5 clusters remain: {Poulidor}, {Swiss-Knife}, {$\text{SKI}_{\text{shamir}}$}, {RC}, and {BC, MAD}. Four of the remaining clusters are actually singletons, which means that among all the published protocols, none of them are equivalent with respect to the seven considered parameters. In the remaining cluster, {BC, MAD}, BC and MAD are equivalent since the mutual authentication is not considered in the configurations. We can also raise that, given constraints on memory, probabilities, etc. the best known protocol to be used belongs to these 6 finalists.

It is finally interesting to compare these 6 finalists with the distance-bounding evolution provided in Figure 1. A protocol that is not a finalist should not necessarily be blamed: most of them have been useful at some point and led to more evolved protocols. However, protocols published today should be new finalists in the cluster-based comparison, possibly after considering additional parameters in the comparison.

16 CONCLUSION

Distance bounding authentication protocols represent a new class of protocols aiming to thwart distance-based attacks whose feasability is rendered possible

Table 9. Protocol configurations

Protocols	ma	d	t	l	b	e	me
BC	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	1	Y	Y	$3n$+cst	linear
MAD	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	1	Y	Y	$3n$+cst	linear
HK	$\left(\frac{3}{4}\right)^n$	$\left(\frac{3}{4}\right)^n$	1	N	Y	$2n$+cst	linear
MP $p_f = 0.5$	$\left(\frac{3}{4}\right)^n$	$> \left(\frac{3}{4}\right)^n$ and < 1	1	Y	Y	$\geq 4n$	linear
MP $p_f = 0.75$	$> \left(\frac{3}{4}\right)^n$ and < 1	$> \left(\frac{3}{4}\right)^n$ and < 1	1	Y	Y	$\geq 4n$	linear
Swiss-Knife	$\left(\frac{1}{2}\right)^n$	$\left(\frac{3}{4}\right)^n$	$\left(\frac{3}{4}\right)^n$	Y	Y	$3n$+cst	affine
Tree-Based $\ell = \sqrt{n}$	$\lim_{n\to\infty} = \left(\frac{1}{2}\right)^n$	$\lim_{n\to\infty} = \left(\frac{1}{2}\right)^n$	1	N	Y	$3n$+cst	non affine
Tree-Based $\ell = 1$	$\lim_{n\to\infty} = \left(\frac{1}{2}\right)^n$	$\lim_{n\to\infty} = \left(\frac{1}{2}\right)^n$	1	N	Y	$3n$+cst	non affine
RC	$\left(\frac{1}{2}\right)^n$	$\left(\frac{1}{2}\right)^n$	1	Y	N	$2n$+cst	affine
Poulidor	$\lim_{n\to\infty} = \left(\frac{1}{2}\right)^n$	$\lim_{n\to\infty} = \left(\frac{1}{2}\right)^n$	1	N	Y	$2n$+cst	linear
KA2 $\alpha = n$	$\lim_{n\to\infty} = \left(\frac{1}{2}\right)^n$	1	1	N	Y	$2n$+cst	linear
KA2 $\alpha = \frac{n}{2}$	$> \left(\frac{1}{2}\right)^n$ and $< \left(\frac{3}{4}\right)^n$	$> \left(\frac{3}{4}\right)^n$ and < 1	1	N	Y	$2n$+cst	linear
YKHL	$> \left(\frac{3}{4}\right)^n$ and < 1	$> \left(\frac{3}{4}\right)^n$ and < 1	1	N	Y	$2n$+cst	linear
$\text{SKI}_{\text{shamir}}$	$\left(\frac{1}{2}\right)^n$	$> \left(\frac{1}{2}\right)^n$ and $< \left(\frac{3}{4}\right)^n$	$\left(\frac{1}{2}\right)^n$	N	N	$\geq 4n$	linear
SKI_{pro}	$> \left(\frac{1}{2}\right)^n$ and $< \left(\frac{3}{4}\right)^n$	$\left(\frac{3}{4}\right)^n$	$> \left(\frac{1}{2}\right)^n$ and $< \left(\frac{3}{4}\right)^n$	N	N	$\geq 4n$	linear
SKI_4	$> \left(\frac{1}{2}\right)^n$ and $< \left(\frac{3}{4}\right)^n$	$> \left(\frac{1}{2}\right)^n$ and $< \left(\frac{3}{4}\right)^n$	$\left(\frac{3}{4}\right)^n$	N	N	$\geq 4n$	linear

by emerging technologies. This survey provides a thorough state-of-the-art of existing protocols and introduces refined security analyses. The comparisons made provide designers with new means to evaluate their performance in a unified manner according to several security and resource paramaters. It may be worthwhile pointing out that the provided cluster-based comparison can easily be modified to better reflect specific practical considerations and/or to include other protocols. Finally, we are aware that attacks other than those considered in this paper might exist. Addressing provable security of distance bounding protocols is therefore a challenge for future research.

REFERENCES

[1] Imad M. Abbadi and Chris J. Mitchell. 2007. Digital Rights Management Using a Mobile Phone. In *Proceedings of the Ninth International Conference on Electronic Commerce (ICEC '07)*. ACM, New York, NY, USA, 185–194.

[2] Gildas Avoine, Muhammed Ali Bingöl, Süleyman Kardaş, Cédric Lauradoux, and Benjamin Martin. 2011. A Framework for Analyzing RFID Distance Bounding Protocols. *Journal of Computer Security – Special Issue on RFID System Security* 19, 2 (March 2011), 289–317.

[3] Gildas Avoine, Christian Floerkemeier, and Benjamin Martin. 2009. RFID Distance Bounding Multistate Enhancement. In *Proceedings of the 10th International Conference on Cryptology in India (LNCS)*, Bimal K. Roy and Nicolas Sendrier (Eds.), Vol. 5922. Springer, New Delhi, India, 290–307.

[4] Gildas Avoine and Chong Hee Kim. 2013. Mutual Distance Bounding Protocols. *IEEE Transactions on Mobile Computing* 12, 5 (May 2013), 830–839.

[5] Gildas Avoine, Cédric Lauradoux, and Benjamin Martin. 2011. How Secret-sharing can Defeat Terrorist Fraud. In *Proceedings of the 4th ACM Conference on Wireless Network Security – WiSec'11*. ACM, Hamburg, Germany, 145–156.

[6] Gildas Avoine, Sjouke Mauw, and Rolando Trujillo-Rasua. 2015. Comparing distance bounding protocols: A critical mission supported by decision theory. *Computer Communications* 67 (2015), 92–102.

[7] Gildas Avoine and Aslan Tchamkerten. 2009. An efficient distance bounding RFID authentication protocol: balancing false-acceptance rate and memory requirement. In *Information Security Conference (LNCS)*, Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio Agostino Ardagna (Eds.), Vol. 5735. Springer, Pisa, Italy, 250–261.

[8] Paramvir Bahl and Venkata N. Padmanabhan. 2000. RADAR: An In-Building RF-Based User Location and Tracking System. In *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies – INFOCOM'00*. Tel Aviv, Israel, 775–784.

[9] Asli Bay, Ioana Cristina Boureanu, Aikaterini Mitrokotsa, Iosif-Daniel Spulber, and Serge Vaudenay. 2012. The Bussard-Bagga and Other Distance-Bounding Protocols under Attacks. In *China International Conference on Information Security and Cryptology – Inscrypt'12 (LNCS)*, Vol. 7763. Springer, Beijing, China.

[10] Samy Bengio, Gilles Brassard, Yvo Desmedt, Claude Goutier, and Jean-Jacques Quisquater. 1991. Secure Implementation of Identification Systems. *Journal of Cryptology* 4, 3 (1991), 175–183.

[11] Thomas Beth and Yvo Desmedt. 1990. Identification Tokens - or: Solving the Chess Grandmaster Problem. In *Advances in Cryptology – CRYPTO '90 (LNCS)*, Vol. 537. Springer, Santa Barbara, California, USA, 169–177.

[12] Ioana Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay. 2015. Practical and Provably Secure Distance-Bounding. *Journal of Computer Security* 23 (2015), 229–257. Issue 2.

[13] Ioana Cristina Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay. 2012. On the Pseudorandom Function Assumption in (Secure) Distance-Bounding Protocols. In *Progress in Cryptology LATINCRYPT 2012 (LNCS)*, Alejandro Hevia and Gregory Neven (Eds.), Vol. 7496. Springer, Santiago, Chile, 100–120.

[14] Ioana Cristina Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay. 2013. Practical & Provably Secure Distance-Bounding. In *Information Security Conference – ISC'13 (LNCS)*, Yvo Desmedt (Ed.), Vol. 7807. Springer, Dallas, Texas, USA, 248–258.

[15] Ioana Cristina Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay. 2013. Secure & Lightweight Distance-Bounding. In *Second International Workshop on Lightweight Cryptography for Security & Privacy - LightSec 2013 (LNCS)*, Gildas Avoine and Orhun Kara (Eds.), Vol. 8162. Springer, Gebze, Turkey, 97–113.

[16] Ioana Cristina Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay. 2013. Towards Secure Distance Bounding. In *Fast Software Encryption – 20th International Workshop, FSE 2013 (LNCS)*, Shiho Moriai (Ed.), Vol. 8424. Springer, Singapore, Republic of Singapore. Invited Talk by Serge Vaudenay.

[17] Stefan Brands and David Chaum. 1993. Distance-Bounding Protocols. In *Advances in Cryptology – EUROCRYPT'93 (LNCS)*, Tor Helleseth (Ed.), Vol. 765. Springer, Lofthus, Norway, 344–359.

[18] Laurent Bussard and Walid Bagga. 2005. Distance-bounding proof of knowledge to avoid real-time attacks. In *Security and Privacy in the Age of Ubiquitous Computing (IFIP)*, Ryoichi Sasaki, Sihan Qing, and Eiji Okamoto (Eds.), Vol. 181. Springer, Chiba, Japan, 223–238.

[19] W. Camp. 2007. Digital rights management based on device proximity. (May 2007). US Patent App. 11/164, 289.

[20] Srdjan Čapkun, Levente Buttyán, and Jean-Pierre Hubaux. 2003. SECTOR: secure tracking of node encounters in multi-hop wireless networks. In *ACM Workshop on Security of Ad Hoc and Sensor Networks – SASN'03*. ACM, Fairfax, Virginia, USA, 21–32.

[21] Srdjan Čapkun and Jean-Pierre Hubaux. 2006. Secure Positioning in Wireless Networks. *IEEE Journal of Selected Areas in Communications* 24, 2 (February 2006), 221–232.

[22] Humphrey Cheung. 2004. How To: Building a BlueSniper Rifle. http://www.tomsguide.com/us/how-to-bluesniper-pt1,review-408.html. (2004).

[23] Omar Choudary and Frank Stajano. 2011. Make Noise and Whisper: A Solution to Relay Attacks. In *Security Protocols 19th International Workshop (LNCS)*, Vol. 7114. Springer, Cambridge, UK, 271–283.

[24] Jolyon Clulow, Gerhard P. Hancke, Markus Kuhn, and Tyler Moore. 2006. So Near and yet So Far: Distance-Bounding Attacks in Wireless Networks. In *European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS) (LNCS)*, Vol. 4357. Springer, 83–97.

[25] John H. Conway. 1976. *On Numbers and Games*. Number 6 in London Mathematical Society Monographs. Academic Press, London-New-San Francisco.

[26] Cas J. F. Cremers, Kasper Bonne Rasmussen, Benedikt Schmidt, and Srdjan Čapkun. 2012. Distance Hijacking Attacks on Distance Bounding Protocols. In *IEEE Symposium on Security and Privacy – S&P'12*. San Francisco, California, USA, 113–127.

[27] Yvo Desmedt. 1988. Major securitv problems with the unforgeable (Feige)-Fiat-Shamir proofs of identity and how to overcome them. In *Worldwide Congress on Computer and Communications Security and Protection – SecuriCom'88*. 147–159.

[28] Yvo Desmedt, Claude Goutier, and Samy Bengio. 1988. Special Uses and Abuses of the Fiat-Shamir Passport Protocol. In *Advances in Cryptology – CRYPTO'87 (LNCS)*, Carl Pomerance (Ed.), Vol. 293. Springer, Santa Barbara, California, USA, 21–39.

[29] Saar Drimer and Steven J. Murdoch. 2007. Keep your enemies close: distance bounding against smartcard relay attacks. In *USENIX Security Symposium – USENIX'07*. USENIX Association, Boston, Massachusetts, USA, 1–16.

[30] Ulrich Dürholz, Marc Fischlin, Michael Kasper, and Cristina Onete. 2011. A Formal Approach to Distance Bounding RFID Protocols. In *Proceedings of the 14th Information Security Conference – ISC'11 (LNCS)*, Vol. 7001. Springer, Xián, China, 47–62.

[31] Uriel Feige, Amos Fiat, and Adi Shamir. 1987. Zero Knowledge Proofs of Identity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing – STOC'87*. New York City, New York, USA, 210–217.

[32] Amos Fiat and Adi Shamir. 1986. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology – CRYPTO'86 (LNCS)*, Vol. 263. Springer, Santa Barbara, California, USA, 186–194.

[33] Marc Fischlin and Cristina Onete. 2013. Subtle kinks in distance-bounding: an analysis of prominent protocols. In *Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks – WISEC'13*. Budapest, Hungary, 195–206.

[34] Marc Fischlin and Cristina Onete. 2013. Terrorism in Distance Bounding: Modeling Terrorist-Fraud Resistance. In *International Conference on Applied Cryptography and Network Security – ACNS 2013 (LNCS)*, Michael Jacobson, Michael Locasto, Payman Mohassel, and Reihaneh Safavi-Naini (Eds.), Vol. 7954. Springer, Banff, AB, Canada, 414–431.

[35] Aurélien Francillon, Boris Danev, and Srdjan Čapkun. 2011. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. In *Network and Distributed System Security Symposium*. San Diego, California, USA.

[36] Lishoy Francis, Gerhard Hancke, and Keith Mayes. 2013. A Practical Generic Relay Attack on Contactless Transactions by Using NFC Mobile Phones. *International Journal of RFID Security and Cryptography* 2, 1 (December 2013), 92–106.

[37] Lishoy Francis, Gerhard P. Hancke, Keith Mayes, and Konstantinos Markantonakis. 2010. Practical NFC Peer-to-Peer Relay Attack using Mobile Phones. In *Workshop on RFID Security (LNCS)*, Sıddıka Berna Örs Yalçın (Ed.), Vol. 6370. Springer, Istanbul, Turkey,

35–49.

[38] Mohammad Ghavami, Lachlan B. Michael, and Ryuji Kohno. 2004. *Ultra Wideband Signals and Systems in Communication Engineering*. Wiley Press.

[39] Tzipora Halevi, Haoyu Li, Di Ma, Nitesh Saxena, Jonathan Voris, and Tuo Xiang. 2013. Context-Aware Defenses to RFID Unauthorized Reading and Relay Attacks. *IEEE Transactions on Emerging Topics in Computing* 1, 2 (December 2013), 307–318.

[40] Gerhard P. Hancke. 2006. Practical Attacks on Proximity Identification Systems (Short Paper). In *IEEE Symposium on Security and Privacy IEEE – S&P'06*. IEEE, Oakland, California, USA, 328–333.

[41] Gerhard P. Hancke. 2010. Design of a Secure Distance-Bounding Channel for RFID. *Journal of Network and Computer Applications* 34, 3 (May 2010), 877–887.

[42] Gerhard P. Hancke and Markus Kuhn. 2005. An RFID Distance Bounding Protocol. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*. IEEE, Athens, Greece, 67–73.

[43] Gerhard P. Hancke and Markus Kuhn. 2008. Attacks on Time-of-Flight Distance Bounding Channels. In *Proceedings of the 1st ACM Conference on Wireless Network Security – WiSec'08*, Virgil D. Gligor, Jean-Pierre Hubaux, and Radha Poovendran (Eds.). ACM, Alexandria, Virginia, USA, 194–202.

[44] Gerhard P. Hancke, Keith Mayes, and Konstantinos Markantonakis. 2009. Confidence in Smart Token Proximity: Relay Attacks Revisited. *Elsevier Computers & Security* 28, 7 (June 2009), 615–627.

[45] Orhun Kara, Süleyman Kardaş, Muhammed Ali Bingöl, and Gildas Avoine. 2010. Optimal Security Limits of RFID Distance Bounding Protocols. In *Workshop on RFID Security (LNCS)*, Sıddıka Berna Örs Yalçın (Ed.), Vol. 6370. Springer, Istanbul, Turkey, 220–238.

[46] Süleyman Kardaş, Mehmet Sabir Kiraz, Muhammed Ali Bingöl, and Hüseyin Demirci. 2011. A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions. In *Workshop on RFID Security (LNCS)*, Vol. 7055. Springer, Amherst, Massachusetts, USA, 78–93.

[47] Ziv Kfir and Avishai Wool. 2005. Picking Virtual Pockets Using Relay Attacks on Contactless Smartcard Systems. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*. IEEE, Athens, Greece, 47–58.

[48] Chong Hee Kim and Gildas Avoine. 2009. RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks. In *International Conference on Cryptology And Network Security – CANS'09 (LNCS)*, Juan A. Garay, Atsuko Miyaji, and Akira Otsuka (Eds.), Vol. 5888. Springer, Kanazawa, Ishikawa, Japan, 119–133.

[49] Chong Hee Kim and Gildas Avoine. 2011. RFID Distance Bounding Protocols with Mixed Challenges. *IEEE Transactions on Wireless Communications* 10, 5 (2011), 1618–1626.

[50] Chong Hee Kim, Gildas Avoine, François Koeune, François-Xavier Standaert, and Olivier Pereira. 2008. The Swiss-Knife RFID Distance Bounding Protocol. In *International Conference on Information Security and Cryptology – ICISC 2008 (LNCS)*, Pil Joong Lee and Jung Hee Cheon (Eds.), Vol. 5461. Springer, Seoul, Korea, 98–115.

[51] Sjouke Mauw, Jorge Toro Pozo, and Rolando Trujillo-Rasua. 2016. A class of precomputation-based distance-bounding protocols. In *1st IEEE European Symposium on Security and Privacy (Euro S&P), Saarbrücken, Germany, March 21-24, 2016*.

[52] Sjouke Mauw, Jorge Toro Pozo, and Rolando Trujillo-Rasua. 2016. Optimality results on the security of lookup-based protocols. In *Workshop on RFID Security (LNCS)*. Springer, Hong Kong.

[53] Aikaterini Mitrokotsa, Christos Dimitrakakis, Pedro Peris-Lopez, and Julio C. Hernandez-Castro. 2010. Reid et al.'s Distance Bounding Protocol and Mafia Fraud Attacks over Noisy Channels. *IEEE Communications Letters* 14, 2 (February 2010), 121–123.

[54] Jorge Munilla, Andres Ortiz, and Alberto Peinado. 2006. Distance Bounding Protocols with Void-Challenges for RFID. In *Workshop on RFID Security*. Ecrypt, Graz, Austria.

[55] Jorge Munilla and Alberto Peinado. 2008. Distance Bounding Protocols for RFID Enhanced by using Void-Challenges and Analysis in Noisy Channels. *Wireless Communications and Mobile Computing* 8, 9 (January 2008), 1227–1232.

[56] Jorge Munilla and Alberto Peinado. 2008. Security Analysis of Tu and Piramuthu's Protocol. In *New Technologies, Mobility and Security – NTMS'08*. IEEE, Tangier, Morocco, 1–5.

[57] Jorge Munilla and Alberto Peinado. 2009. Enhanced Low-cost RFID Protocol to Detect Relay Attacks. *Wireless Communications and Mobile Computing* 10, 3 (March 2009), 361–371.

[58] Ventzislav Nikov and Marc Vauclair. 2008. Yet Another Secure Distance-Bounding Protocol. Cryptology ePrint Archive, Report 2008/319. (2008).

[59] Pedro Peris-Lopez, Julio C. Hernandez-Castro, Juan M. Estevez-Tapiador, Esther Palomar, and Jan C. A. van der Lubbe. 2010. Cryptographic Puzzles and Distance-bounding Protocols: Practical Tools for RFID Security. In *IEEE RFID 2010*. Orlando, Florida, USA, 45–52.

[60] Pedro Peris-Lopez, Julio C. Hernandez-Castro, Juan M. Estevez-Tapiador, and Jan C. A. van der Lubbe. 2009. Shedding Some Light on RFID Distance Bounding Protocols and Terrorist Attacks. arXiv.org. (2009).

[61] Kasper Bonne Rasmussen. 2011. *Primitives for secure localization and location verification*. Ph.D. Dissertation. ETH Zurich, ETH Zurich, Switzerland.

[62] Kasper Bonne Rasmussen, Claude Castelluccia, Thomas S. Heydt-Benjamin, and Srdjan Čapkun. 2009. Proximity-based Access Control for Implantable Medical Devices. In *ACM Conference on Computer and Communications Security*, Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis (Eds.). ACM, Chicago, Illinois, USA, 43–53.

[63] Kasper B. Rasmussen and Srdjan Čapkun. 2010. Realization of RF Distance Bounding. In *USENIX Security Symposium – USENIX'10*. USENIX, Washington, DC, USA, 389–402.

[64] Jason Reid, Juan Gonzalez Nieto, Tee Tang, and Bouchra Senadji. 2007. Detecting Relay Attacks with Timing Based Protocols. In *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security – ASIACCS '07*, Feng Bao and Steven Miller (Eds.). ACM, Singapore, Republic of Singapore, 204–213.

[65] Dave Singelée and Bart Preneel. 2007. Distance Bounding in Noisy Environments. In *Security and Privacy in Ad-hoc and Sensor Networks – ESAS 2007 (LNCS)*, Frank Stajano, Catherine Meadows, Srdjan Čapkun, and Tyler Moore (Eds.), Vol. 4572. Springer, Cambridge, UK, 101–115.

[66] Mohammad Reza Sohizadeh Abyaneh. 2011. Security Analysis of two Distance-Bounding Protocols. In *Workshop on RFID Security (LNCS)*, Ari Juels and Christof Paar (Eds.), Vol. 7055. Springer, Amherst, Massachusetts, USA, 94–107.

[67] Frank Stajano, Ford-Long Wong, and Bruce Christianson. 2010. Multichannel Protocols to Prevent Relay Attacks. In *Financial Cryptography and Data Security, 14th International Conference – FC'10 (LNCS)*, Vol. 6052. Springer, Tenerife, Canary Islands, 4–19.

[68] Rolando Trujillo-Rasua. 2013. Complexity of distance fraud attacks in graph-based distance bounding. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, Tokyo, Japan.

[69] Rolando Trujillo-Rasua, Benjamin Martin, and Gildas Avoine. 2010. The Poulidor Distance-Bounding Protocol. In *Workshop on RFID Security (LNCS)*, Sıddıka Berna Örs Yalçın (Ed.), Vol. 6370. Springer, Istanbul, Turkey, 239–257.

[70] Rolando Trujillo-Rasua, Benjamin Martin, and Gildas Avoine. 2014. Distance Bounding Facing Both Mafia and Distance Frauds. *IEEE Transactions on Wireless Communications* 13, 10 (2014), 5690–5698.

[71] Yu-Ju Tu and Selwyn Piramuthu. 2007. RFID Distance Bounding Protocols. In *International EURASIP Workshop on RFID Technology*. Vienna, Austria.

[72] Pascal Uriena and Selwyn Piramuthu. 2014. Elliptic curve-based RFID/NFC authentication with temperature sensor input for relay attacks. *Decision Support Systems* 59, 0 (October 2014), 28–36.

[73] Serge Vaudenay. 2013. On Modeling Terrorist Frauds. In *International Conference on Provable Security – ProvSec (LNCS)*, Willy Susilo and Reza Reyhanitabar (Eds.), Vol. 8209. Springer, Melaka, Malaysia, 1–20.

[74] Serge Vaudenay. 2015. Private and Secure Public-Key Distance Bounding - Application to NFC. In *Financial Cryptography and Data Security, 19th International Conference – FC'15 (LNCS)*, Rainer Böhme and Tatsuaki Okamoto (Eds.), Vol. 8975. Springer, San Juan, Puerto Rico, 207–216.

[75] Brent R. Waters and Edward W. Felten. 2003. *Secure, Private Proofs of Locations*. Technical Report TR-667-03. Princeton Computer Science, Princeton, New Jersey, USA.

[76] Dae Hyun Yum, Jin Seok Kim, Sung Je Hong, and Pil Joong Lee. 2011. Distance Bounding Protocol for Mutual Authentication. *IEEE Transactions on Wireless Communications* 10, 2 (2011), 592–601.

17 SUPPLEMENTARY MATERIALS: APPENDIX A

This section provides the full description of the protocols considered in this paper.

17.1 Brands and Chaum's Protocol (1993)

In 1993, Brands and Chaum designed several distance-bounding protocols [17]. This analysis focuses on their protocol (Algorithm 1) that mitigates both mafia fraud and distance fraud.

Initialization. The prover should own a signature public/private key.

Protocol. The prover randomly generates n commitment bits $m_i \in_R \{0,1\}$ and the verifier randomly generates n challenge bits $c_i \in_R \{0,1\}$ $(i = 1 \dots n)$. The prover commits on $m_1 || \dots || m_n$ and sends this commitment to the verifier. Then, a phase of n rapid bit exchanges starts. In each round, the verifier starts his timer and sends c_i to the prover, who replies with $r_i = c_i \oplus m_i$. Upon receiving the response bit the verifier stops its timer. Finally, the prover concatenates c_i and r_i, signs the $2n$ bits result, $\text{Sign}_k(c_1||r_1|| \dots ||c_n||r_n)$, and sends it to the verifier together with the opening of the commitment.

Final Decision. Upon reception of the signature, the verifier concatenates the $2n$ bits c_i and r_i, and verifies the received signature, the commitment, the measured Δt_i's and whether $r_i = c_i \oplus m_i$ for $i = 1 \dots n$.

Table 10. Parameters and functions (Algorithm 1)

n	Number of iterations in the fast phase
ℓ	Lower bound for the size of the commitment and the signature ($\ell >> n$)
$t_{\max}$	Threshold of the round-trip time
Commit	Secure commitment function that outputs ℓ bits
$Sign_{K_s}$	Signature function whose private key is K_s

17.2 Čapkun, Buttyán, and Hubaux's Protocol (2003)

In 2003, Čapkun, Buttyán, and Hubaux introduced MAD [20], a protocol that works quite similarly to the BC protocol [17], but provides mutual authentication. Although denoted by P and V, the two parties act as both prover and verifier during the execution of the protocol (Algorithm 2). The notations used in [20] are kept in the description below.

Initialization. Prior to the protocol execution, the two parties (P and V) agree on the security parameters and functions described in Table 11, and a common secret key K.

Protocol. In the first slow phase, P and V generate two random numbers (r, r' and s, s' respectively) and send a commitment to the other party on the two random numbers ($h(r||r')$ and $h(s||s')$ respectively). During the fast phase the following steps are repeated n times:

- P sends the bit α_i to V, where $\alpha_1 = r_1$ and $\alpha_i = r_i \oplus \beta_{i-1}$ for $i > 1$;
- V sends the bit $\beta_i = s_i \oplus \alpha_i$ to P.

In the second slow phase, P retrieves the bit sequence s by assuming that $s_i = \alpha_i \oplus \beta_i$ for every $i \in \{1, 2, \cdots, n\}$ and computes using the secret key K the value $\mu_P = \mathrm{MAC}_K(ID_P||ID_V||r_1||s_1|| \ldots ||r_n||s_n)$. Similarly, V computes the bits $r_1 = \alpha_1$ and $r_i = \alpha_i \oplus \beta_{i-1}$ for $i > 1$, with which V computes $\mu_V = \mathrm{MAC}_K(ID_V||ID_P||s_1||r_1|| \ldots ||s_n||r_n)$. Finally, P and V open the commitment sent in the first slow phase by transmitting r' and s', and exchange the values μ_P and μ_V.

Final Decision. The users P and V accept each other's entity only if:

- the n responses of the fast phase are correct,
- the commitment that was sent in the first slow phase is correctly opened in the second slow phase and corresponds to the bit sequence (r or s) exchanged during the fast phase,
- the output of the MAC function is correct, and
- the time constraint $\Delta t_i \leq t_{\max}$ is met for $i \in \{1, 2, \ldots, n\}$ and some threshold $t_{\max} > 0$.

Table 11. Parameters and functions (Algorithm 2)

n	Number of iterations in the fast phase, which is also the size of the random numbers r and s
δ	Size of the random numbers r' and s'
κ	Size of the secret key K
$t_{\max}$	Threshold of the round-trip time
MAC_K	MAC function keyed with K
σ	Output size of the MAC function
h	Collision-resistant one-way hash function used to compute the commitment

17.3 Hancke and Kuhn's Protocol (2005)

In 2005 Hancke and Kuhn published the first distance-bounding protocol [42] (Algorithm 3) clearly dedicated to RFID. The protocol relies on the original ideas of Desmedt et al. [10, 28] but is different from Brands and Chaum's work [17] in the sense that Hancke and Kuhn's protocol does not have any final signature after the fast phase.

Initialization. Prior to the protocol execution, the legitimate prover and the verifier agree on the security parameters and functions described in Table 12, and a common secret key K.

Protocol. During the slow phase, the verifier sends to the prover a nonce N_V and the prover sends to the verifier a nonce N_P. Both the prover and the verifier then use the pseudo-random function h and the secret key K in order to generate two n-bit sequences R^0 and R^1. For each of the n rounds of the fast phase, the

verifier generates and sends a random challenge bit c_i, and the prover replies instantly with a one-bit response that is either R_i^0 or R_i^1, selected by the value of c_i.

Final Decision. The verifier accepts the prover's identity only if the n responses of the fast phase are correct while meeting the time constraint $\Delta t_i \leq t_{\max}$, $i \in \{1, 2, \ldots, n\}$, for some threshold $t_{\max} > 0$.

Table 12. Parameters and functions (Algorithm 3)

n	Number of iterations in the fast phase
κ	Size of the secret key K
δ	Size of nonces N_V and N_P
$t_{\max}$	Threshold of the round-trip time
h	Hash function whose output size is $2n$

17.4 Bussard and Bagga's Protocol (2005)

Bussard and Bagga published the DBPK-Log protocol (Algorithm 4), which is a distance-bounding protocol based on a proof of knowledge and a commitment scheme [18].

Initialization. Prior to protocol execution, the prover and the verifier agree on the security parameters and functions described in Table 13. A trusted authority then chooses and publishes the following values: p, a large safe prime such that $p = 2q + 1$ with q a large prime; g, a generator of $\mathbb{Z}_p^\star$; and h, a random value in $\mathbb{Z}_p^\star$. Once done, the prover selects a secret $x \in \mathbb{Z}_{p-1} \setminus \{q\}$ and the trusted authority then needs to create and publish a certificate for his public key $y = g^x$.

Protocol. The prover P possesses a private key x which is an odd secret, randomly chosen in $\mathbb{Z}_{p-1} \setminus \{q\}$, whose corresponding public key is $y = g^x \bmod p$. The prover picks a random one-time key $R^0 \in \{0,1\}^n$, and encrypts his private key x with R^0, using the encryption scheme E, i.e., he gets $R^1 = E_{R^0}(x) = x - R^0 \bmod (p-1)$. The prover then commits to each bit of R^0 and R^1 independently using the Commit function (see Remark 5). Then, for each of the n rounds of the fast phase, the verifier generates and sends a random challenge bit c_i, and the prover replies instantly with a 1-bit response that is either R_i^0 or R_i^1, selected by the value c_i. A second slow phase then starts, where the prover allows the verifier to open the commitment of each bit $R_i^{c_i}$, for each challenge c_i that has been sent in the previous phase.

Final Decision. The verifier checks the timing and verifies that the received values correspond to the committed ones (Open function is described in the original paper). A verification protocol is finally executed between P and V using a proof of knowledge. Note that [18] only states that "*at the end of distance-bounding stage, the verifier V is able to compute an upper bound on the distance to P.*"

Table 13. Parameters and functions (Algorithm 4)

n	Number of iterations in the fast phase, $n = m + m'$
m	Security parameter $m = \lceil \log_2 p \rceil$
m'	Security parameter
E	Encryption scheme (additive cipher): $E(x) = x - k \bmod (p-1)$
Commit	Commit function of the commitment scheme
Open	Open function of the commitment scheme
$PK[(x,v) : z = \Omega(x,v) \wedge y = \Gamma(x)]$	Proof of knowledge for x and v

Remark 5 (Commitment). *The suggested Commit function works as follows: (a) a value h is randomly chosen in $\mathbb{Z}_p^\star$, (b) the values $v_{R^0,i}$ and $v_{R^1,i}$, $\forall i \in \{0, ..., N-1\}$, are randomly chosen in $\mathbb{Z}_{p-1}$, (c) $C_i(R^0) = g^{R_i^0} h^{v_{R^0,i}} \bmod p$ and $C_i(R^1) = g^{R_i^1} h^{v_{R^1,i}} \bmod p$.*

17.5 Munilla and Peinado's Protocol (2006)

Munilla and Peinado introduced in [54, 57] the concept of void challenges as a tool to improve distance-bounding protocols. These void challenges can also be used to decrease the mafia fraud success probability when applied to Hancke and Kuhn's protocol [55], which is the case analysed in this section. Thus, for this protocol (Algorithm 5), the challenges can be 0, 1 or void, where a void challenge means that no challenge is sent. Void challenges are used to detect a mafia fraud using the pre-ask strategy.

Initialization. Prior to the protocol execution, the prover and the verifier agree on the security parameters and the functions described in Table 14, and a common secret key K.

Protocol. During a first slow phase, the verifier sends to the prover a nonce N_V and the prover sends to the verifier a nonce N_P (Remark 6). They both then use the pseudo-random function h and the secret key K to generate three n-bit sequences: R^0, R^1 and Z. The values R^0 and R^1 are, as in Hancke and Kuhn's protocol, the responses to the challenges, while Z defines which challenges are void. In the fast phase, the verifier sends random challenges c_i when $Z_i = 1$, and the prover instantly replies with 1-bit responses r_i that are either R_i^0 or R_i^1, depending on the value of c_i: $r_i = R_i^{c_i}$. If the prover receives a challenge for an interval where $Z_i = 0$, he assumes that the system is being attacked and aborts the protocol. Finally, the prover sends $h(K, R^0, R^1)$ in a final slow phase to confirm that no adversary has been detected.

Final Decision. The verifier accepts the prover as genuine only if the final signature is correct and all the responses r_i are correct and timely: $\Delta t_i \leq t_{\max}$, $i \in 1, 2, \cdots, n$, for some threshold $t_{\max} > 0$.

Table 14. Parameters and functions (Algorithm 5)

n	Number of iterations in the fast phase (it coincides with the length of vectors R^0, R^1 and Z)
κ	Size of the secret key K (not defined in the original paper)
δ	Size of random numbers N_V and N_P (not defined in the original paper)
$t_{\max}$	Threshold of the round-trip time
p_f	Probability of an interval being non-void (optimal value $p_f = 4/5$, and practical value $p_f = 3/4$)
h	Hash (or pseudo-random) function whose output size is $3n$

REMARK 6. *The paper does not specify whether the verifier or the prover sends its nonce in first.*

REMARK 7. *During the fast phase,* $2(n - n_{void})$ *bits are exchanged, where* n_{void} *is the number of void challenges for the protocol run. Given the average number of void challenges, namely* $n(1 - p_f)$*, the average number of exchanged bits is* $2np_f$*.*

17.6 Kim, Avoine, Koeune, Standaert and Pereira's Protocol (2008)

Kim, Avoine, Koeune, Standaert and Pereira introduced a protocol in [50] known as the *Swiss-knife distance-bounding protocol*[4] (Algorithm 6).

Initialization. Prior to the protocol execution, the legitimate prover and the verifier agree on the security parameters and functions described in Table 15, a system-wide constant C known to the verifier and the prover, and a common secret key K.

Protocol. During the first slow phase, the verifier chooses a nonce $N_V \in_{\mathsf{R}} \{0,1\}^\delta$ and a random binary vector D with Hamming weight n and length σ. Intuitively, D corresponds to a mask pointing to the positions on which the prover will be questioned during the fast phase. He transmits N_V and D to the prover. The prover chooses a nonce $N_P \in_{\mathsf{R}} \{0,1\}^\delta$ and computes $a := f_K(C, N_P)$. The prover then computes two registers using its permanent key K as follows: $Z^0 := a$ and $Z^1 := a \oplus K$. He finally prepares the possible answers by extracting the relevant parts of Z^0, Z^1 according to the mask D, building the n-bit vectors R^0 and R^1. The prover ends the slow phase transmitting N_P to the verifier. During the fast phase, the verifier generates and sends a random challenge bit c_i, and the prover replies instantly with a 1-bit response that is either v_i^0 or v_i^1, selected by the value of c_i. After n iterations, the prover computes $T_B := f_K(c'_1, \ldots, c'_n, ID, N_V, N_P)$ and transmits T_B and the challenges $c'_1, \ldots, c'_n$ received during the fast

[4]Like Swiss-army knives used during WWII, the Swiss-knife protocol is a multi-purpose tool. The authors claim their protocol *"resists against both mafia fraud and terrorist attacks, reaches the best known false acceptance rate, preserves privacy, resists to channel errors, uses symmetric-key cryptography only, requires no more than 2 cryptographic operations to be performed by the tag, can take advantage of precomputation on the tag, and offers an optional mutual authentication"* [50].

phase. The verifier performs a search over its database until he finds a pair (ID, K) and computes R^0, R^1. If *mutual* authentication is expected, the verifier computes $T_A := f_K(N_P)$, sends it to the prover who checks its correctness.

Final Decision. The authentication succeeds if and only if $err_C + err_R + err_T < T$.

Table 15. Parameters and functions (Algorithm 6)

n	Number of iterations in the fast phase
σ	Size of the output of f and consequently size of the secret key K
δ	Size of nonces N_V and N_P
$t_{\max}$	Threshold of the round-trip time
T	Threshold of tolerable errors
f	Pseudo-random function whose output size is σ

17.7 Avoine and Tchamkerten's Protocol (2009)

The protocol (Algorithm 7) introduced by Avoine and Tchamkerten in [7] is a generalization of Hancke and Kuhn's protocol that is more secure in terms of mafia and distance fraud.

Initialization. Prior to the protocol execution, the legitimate prover and the verifier agree on the security parameters and functions described in Table 16, in addition to a common secret key K.

Protocol. It consists of a slow *authentication* phase followed by a fast *proximity check* phase. Both phases have their own security parameters: the credential size c for the authentication and the number of bit exchanges n between the prover and the verifier during the fast phase.

Authentication. The verifier sends a nonce N_V to the prover, in the form of a uniformly random bit-string of size δ. The prover then generates a δ-bit nonce N_P and, based on N_V and N_P, computes a keyed-hash value $h_K(N_V, N_P)$ whose output is a string of at least $c + \ell \cdot (2^{d+1} - 2)$ bits where $d, \ell \geq 1$ are such that $d \cdot \ell = n$. The prover sends to the verifier both N_P and the first c bits of $h_K(N_V, N_P)$ denoted $[h_K(N_V, N_P)]_1^c$.

Proximity Check. Using the subsequent $q = \ell \cdot (2^{d+1} - 2)$ bits of the hash value $h_K(N_V, N_P)$, denoted by $[h_K(N_V, N_P)]_{c+1}^{c+q}$, the prover and the verifier label ℓ full binary trees of depth d as follows (see Figure 6 for an example). The left and the right edges of each tree are labeled 0 and 1 respectively, and each node of each tree, except the root, is associated with the value of a particular bit in $[h_K(N_V, N_P)]_{c+1}^{c+q}$ in a one-to-one fashion.[5] This labeling is possible since each

[5]To do this, one sequentially assigns the bit values of $[h_K(N_V, N_P)]_{c+1}^{c+q}$ to all the nodes of each tree, starting with the lowest level nodes, moving left to right, and moving up after assigning the nodes of the current level.

tree has $2^{d+1}-2$ nodes (excluding the root), which gives a total of $\ell \cdot (2^{d+1}-2)$ nodes to be labeled.

An n-round fast bit exchange between the verifier and the prover proceeds using the trees: the edge and the node values represent the verifier's challenges and the prover's replies, respectively. At each step $i \in \{1, 2, \ldots, n\}$ the verifier generates a challenge in the form of a randomly uniform bit c_i and sends it to the prover. Now let $j \geq 1$ be such that $(j-1)(2^{d+1}-2)+1 \leq i \leq j(2^{d+1}-2)$. Upon receiving c_i, the prover replies r_i, which corresponds to the value of the node in the j-th tree whose edge path from the root is given by $c_{(j-1)(2^{d+1}-2)+1}, c_{(j-1)(2^{d+1}-2)+2}, \ldots, c_i$. The example illustrated by Figure 6 uses the following parameters: $n = 6$, $\ell = 2$, and $d = 3$. The sequence of challenges is $(1, 1, 0, 0, 1, 0)$, which corresponds to the two thick edge paths in the trees starting with the tree on the left. The corresponding sequence of replies is $(1, 1, 1, 0, 1, 0)$. Note that each reply r_i is a function of at most d previous c_j's. Finally, for all $i \in \{1, 2, \ldots, n\}$, the verifier measures the time interval Δt_i between the instant c_i is sent until the instant r_i is received.

Final Decision. The verifier accepts the prover's identity only if the c authentication bits are correct and if the n replies of the fast phase are correct while meeting the time constraint $\Delta t_i \leq t_{\max}$, $i \in \{1, 2, \ldots, n\}$, for some threshold $t_{\max} > 0$.

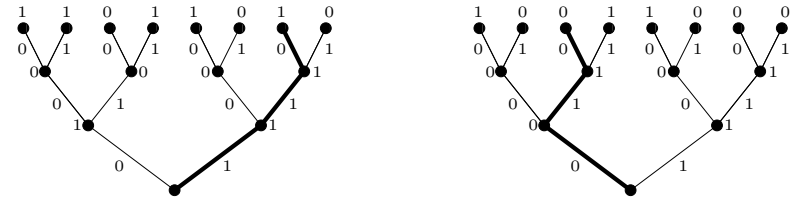


Fig. 6. Two decision trees of depth 3, i.e., $\ell = 2$, $d = 3$

REMARK 8. *The case $n = d \cdot \ell$ is the maximum situation where all d replies of the ℓ-th tree are used. We impose this constraint only to have somewhat simpler performance expressions. It is easy to see that this constraint can be replaced by $d \cdot \ell \geq n$, which is the situation where the last tree is only partly used.*

REMARK 9. *When $d = 1$ and $\ell = n$, the fast phase of the protocol reduces to the HK Protocol.*

17.8 Rasmussen and Čapkun's Protocol (2010)

The protocol (Algorithm 8) was introduced by Rasmussen and Čapkun and originally appeared in [63]. In this paper we consider the updated version that appeared in [61].

Initialization. Prior to the protocol execution, the legitimate prover and the verifier agree on the security parameters, the functions described in Table 17, and a common secret key K.

Table 16. Parameters and functions (Algorithm 7)

n	Number of iterations in the fast phase
κ	Size of secret key
δ	Size of nonces N_V and N_P
c	Credential size
d	Depth of each tree
ℓ	Number of trees (d and ℓ satisfy $d \cdot \ell = n$)
$t_{\max}$	Threshold of the round-trip time
h_K	Keyed-hash function whose output size is a bit string of size at least $c + \ell \cdot (2^{d+1} - 2)$

Protocol. The prover starts the protocol by picking a fresh (large) nonce N_P. The prover then commits (using for example a hash) on N_P and its identity. This commitment is not keyed. The prover now activates its distance-bounding hardware and set the output channel according to the opposite of the first bit of the nonce N_P. From this moment on, any signal that the prover receives on channel C_0 will be reflected on the output channel that is set. However, the prover does not start switching between output channels yet.

Upon receiving the commitment, the verifier picks a fresh (large) nonce N_V and prepares to initiate the distance-bounding phase, in which it will measure the distance bound to the prover. The verifier starts a high precision clock to measure the (round trip) time-of-flight of the signal, Δt, and begins to transmit his nonce N_V on channel C_0. From this point on, the verifier also listens on the two reply channels C_1 and C_2 and keeps listening on the two channels until he either receives the expected response from the prover or until he detects an error and aborts the protocol.

As soon as the prover receives (and, in parallel demodulates) the first bit of N_V on C_0, he starts switching reply channels according to the bits of his nonce N_P. When the first few bits are being demodulated, the prover is still reflecting the input (challenge) bits and the switching of the channels is not started yet (i.e., the prover does not start sending back N_P yet). This function, used by the prover to form its reply to the verifier, is called "Challenge Reflection with Channel Selection" (CRCS). The demodulation of the bits is not done within the distance-bounding hardware (called the distance-bounding extension), but is done in the prover's regular radio. A possible implementation of the distance-bounding extension (i.e., of CRCS) using analog mixers is described in [61]. It is not important how long it takes for the prover's radio to demodulate the first bits since the prover does not need to begin to switch the output channels within any predefined time as long as the prover keeps track of the delay n. The delay represents the time taken by the prover to react to the incoming signal, i.e., to switch its circuit to transmit the first bit of its answer. The switching starts within the duration of N_V, and allows the transmission of N_P. The first part of N_V could even be known and constitute a public and fixed-length preamble,

upon the detection of which the prover would start switching the channels (i.e., would start sending N_P).

When the prover starts sending N_P, he sends the bits of N_P with a fixed frequency (e.g., every $100ms$) by switching channels depending on the value of the current bit. In each interval, the prover reflects back several bits of N_V and a single bit of N_P. The bit of N_P is encoded in the choice of the reply channel. The prover also receives in parallel the verifier's challenge nonce (i.e., N_V) on channel C_0 using his regular radio.

When the verifier has sent all the bits of his nonce, he waits for the prover to complete the reflection of the signal and then both the prover and verifier disable their distance-bounding extensions. The verifier can then use an auto-correlation detector like the ones used in GPS receivers to determine the exact time of flight, Δt, of the reflected signal. This can also be done during the distance-bounding phase, i.e., in parallel to the analog distance-bounding circuit. Finally, the prover sends a signed message compounded by the commitment sent during the first slow phase, the delay n, his nonce N_P, and the verifier's identity and nonce.

Final Decision. The verifier accepts the prover's identity only if the bits of N_P were sent within the same time duration, these bits match with those he received in the final message of the prover, the reflection of N_V through the channel switch was correct, the signature in the final message is correct, the delay n' he computed match with the prover's one n (including in the final message), and finally that the round-trip time is below the time threshold $t_{\max}$.

Table 17. Parameters and functions (Algorithm 8)

δ_V	Size of the verifier's challenge nonce N_V
δ_P	Size of the prover's nonce N_P
σ	Lower bound for the size of the commitment and the signature
$t_{\max}$	Threshold of the round trip time
Commit	Secure commitment function that outputs σ bits.
Sign	Signature function whose output size is σ

17.9 Trujillo-Rasua, Martin and Avoine's Protocol (2010)

Poulidor, the graph-based distance-bounding protocol (Algorithm 9) designed by Trujillo-Rasua, Martin, and Avoine [69], uses specific node and edge dependencies in the tree of the AT protocol [7] – which then can alternatively be represented by an acyclic graph. Poulidor benefits from a lower memory requirement compared to the AT protocol. Security is also reduced.

Initialization. Prior the protocol execution, the legitimate prover and the verifier agree on the security parameters and functions described in Table 18, and a common secret K.

Protocol. During the slow-phase, both the verifier and the prover build a directed graph G. The proposed graph requires $2n$ nodes $\{q_0, q_1, \ldots, q_{2n-1}\}$, and $4n$ edges $\{s_0, s_1, \cdots, s_{2n-1}, \ell_0, \ell_1, \cdots, \ell_{2n-1}\}$ such that, s_i $(0 \leq i \leq 2n-1)$

Table 18. Parameters and functions (Algorithm 9)

n	Number of iterations in the fast phase
κ	Size of the secret key K
δ_P and δ_V	Size of nonces N_P and N_V respectively
$t_{\max}$	Threshold of the round-trip time
H	Hash function whose output size is $2n$

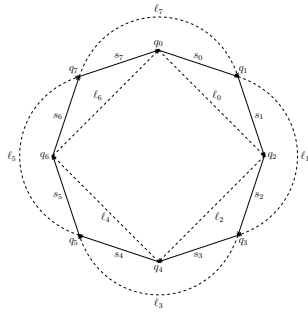


Fig. 7. Graph when $n = 4$

is an edge from q_i to $q_{(i+1) \mod 2n}$, and ℓ_i $(0 \leq i \leq 2n-1)$ is an edge from q_i to $q_{(i+2) \mod 2n}$. Figure 7 depicts the graph when $n = 4$.

In order to build G, the verifier sends a nonce N_V to the prover, and the latter sends a nonce N_P to the verifier. From these values, and the secret K, they compute $H = h(K, N_P, N_V)$ and set up a graph G as follows: the first $2n$ bits are used to value the nodes while the remaining bits are used to value the edges s_i $(0 \leq i \leq 2n-1)$, and finally $\ell_i = s_i \oplus 1$ $(0 \leq i \leq 2n-1)$. After agreeing on the graph, the fast phase begins. This phase consists of n stateful rounds numbered from 0 to $n-1$. Initially $q_{p_0} = q_{v_0} = q_0$, but in the i-th round P's state and V's state are represented by the nodes q_{p_i} and q_{v_i} respectively. Upon reception of the i-th challenge c_i, P moves from the node q_{p_i} to $q_{p_{i+1}}$ in the following way: $q_{p_{i+1}} = q_{(p_i+1) \mod 2n}$ if s_i is labeled with c_i, otherwise $q_{p_{i+1}} = q_{(p_i+2) \mod 2n}$. Finally, the prover sends as response r_i the bit-value of the node $q_{p_{i+1}}$. Upon reception of the prover's answer r_i, the verifier stops his timer, and computes Δt_i, i.e., the round trip time spent for this exchange. Besides this, V moves to the node $q_{v_{i+1}}$ using the challenge c_i (as the prover did but from the node q_{v_i}) and checks if $q_{v_{i+1}} = r_i$.

Final Decision. The verifier accepts the prover's identity only if n responses of the fast phase are correct and the time constraint $\Delta t_i \leq t_{\max}$, $i \in \{1, 2, \ldots, n\}$, for some threshold $t_{\max} > 0$.

17.10 Kim and Avoine's Protocol (KA2) (2011)

Kim and Avoine introduced in 2009 a distance-bounding protocol with mixed challenges [48], namely challenges known and challenges unknown in advance by the prover. Challenges known in advance allow the prover to help the verifier to detect an attack, but these challenges also allow the prover to perform a distance fraud. Kim and Avoine improved their protocol in 2011, yielding a new variant known as KA2 [49], which is analyzed in this section (Algorithm 10).

Initialization. Prior to the protocol execution, the legitimate prover and the verifier agree on the security parameters and functions described in Table 19, along with a common secret key K.

Protocol. The verifier sends the prover a nonce N_V and the prover sends the verifier a nonce N_P. They then use the pseudo-random function h and the secret key K to generate a $2n$-bit sequence $D||R^0||R^1$.

During the first α rounds, the verifier sends predefined 1-bit challenges c_i. In every round, the prover sends a 1-bit response that is R_i^0 if $c_i = D_i$. Otherwise, he sends random answers until the end of the fast phase.

During the remaining $n - \alpha$ rounds, the verifier sends random 1-bit challenges c_i. In every round, the prover sends a 1-bit response that is $R_i^{c_i}$, or he sends random answers until the end of the fast phase if a problem ($c_i \neq D_i$) was detected during the first α rounds.

Final Decision. The verifier accepts the prover's identity only if n responses of the fast phase are correct, while also meeting the time constraint $\Delta t_i \leq t_{\max}$, $i \in \{1, 2, \ldots, n\}$, for a threshold $t_{\max} > 0$.

Table 19. Parameters and functions (Algorithm 10)

n	Number of iterations in the fast phase
κ	Size of the secret key K
δ	Size of nonces N_V and N_P
$t_{\max}$	Threshold of the round-trip time
α	Number of predefined rounds
h	Pseudo-random function whose output size is $2n$

17.11 Yum, Kim, Hong and Lee's Protocol (2010)

Yum, Kim, Hon and Lee created a distance-bounding protocol with mutual authentication [76].

Initialization. Prior to the protocol execution the users A and B agree on the security parameters and functions described in Table 20, in addition to a common secret key K.

Protocol. The protocol consists of a slow phase where two nonces (N_A and N_B) are exchanged, and a fast phase where challenge bits c_i and response bits r_i are exchanged. In the slow phase, the users compute three n-bit sequences, D, R^0, and R^1 using a pseudo-random function applied to N_A and N_B. In the i-th round of the fast phase, each user acts as a prover or a verifier according to the "direction bit" D_i. When $D_i = 0$, A sends a random challenge bit c_i and B answers with ${R_i}^{c_i}$, i.e., the i-th bit of the register R^{c_i}. When $D_i = 1$, B sends a challenge and A responds. If the received response bit is incorrect, the recipient moves to a "protection mode": he sends random bits for all subsequent rounds. Each user also checks that no collision occurred in the round, that is, the two users did not talk or remain silent simultaneously.

Final Decision. A accepts B as legitimate only if the responses of the fast phase are correct and meet the time constraint $\Delta t_i \leq t_{\max}$ for Case I, for some threshold $t_{\max} > 0$. So does B for Case II.

Table 20. Parameters and functions (Algorithm 11)

n	Number of iterations in the fast phase
κ	Size of the secret key K
δ	Size of nonces N_A and N_B
$t_{\max}$	Threshold of the round-trip time
h	Pseudo-random function whose output size is $3n$

17.12 SKI Protocols (2013)

In [14–16], the authors introduced a series of protocols called *SKI*. These protocols (presented in Algorithm 12) are described as follows.

Initialization. Prior to the protocol execution, the legitimate prover and the verifier agree on the security parameters and functions described in Table 21, and a common secret K.

Table 21. Parameters and functions (Algorithm 12)

n	Number of iterations in the fast phase
t	Size of the challenges domain
t'	Security parameter
$\mathfrak{q}$	Power of a prime number
κ	Size of the secret key K
δ	Size of the nonces N_P and N_V
$t_{\max}$	Threshold of the round-trip time
f	Pseudo random function whose output size is $t'n$ elements of $\mathbb{F}_{\mathfrak{q}}$
x	Maximum number of incorrect rounds

Protocol. During the slow phase, the prover first generates a nonce N_P, and sends it to the verifier. The verifier then generates its nonce N_V along with $a = (a_1, \ldots, a_{t'})$ ($a_i \in \mathbb{F}_{\mathfrak{q}}^n$ where $\mathbb{F}_{\mathfrak{q}}$ is the finite fields of order $\mathfrak{q}$ and the authors of [14] employ in concrete examples $\mathfrak{q} = 2$) and a mapping $L \in \mathcal{L}$, where $\mathcal{L}$ is defined below. Using its nonce and the prover's nonce, he computes $f_K(N_P, N_V, L)$ and XORs it with a, in order to obtain the mask M. Finally, the verifier sends N_V, L, and M to the prover. Using these two values and its nonce, the prover computes a and $K' = L(K)$.

Then the n-round fast phase begins. In each round, the verifier picks a challenge $c_i \in \{1, \ldots, t\}$ at random. Then, he starts a timer and sends c_i to the prover. Upon reception of the challenge, the prover first checks whether c_i belongs to $\{1, \ldots, t\}$. If $c_i \notin \{1, \ldots, t\}$ the protocol stops. If $c_i \in \{1, \ldots, t\}$, the prover computes its answer, $r_i = F(c_i, a_i, K'_i)$, where the function F is presented in more details below. The prover then sends its answer back to the verifier. Once the verifier received r_i, he stops its timer and stores Δt_i, the round trip time of the round i, as well as r_i.

As discussed below, the **SKI** protocol is specified with another set $\mathcal{L} = \mathcal{L}_{\mathsf{bit}}$ containing all functions L_μ, for $\mu \in \mathbb{F}_{\mathfrak{q}}^\kappa$ defined by $L_\mu(K) = (\mu \cdot K, \ldots, \mu \cdot K)$

i.e., $L_\mu(K)$ is the n-bit vector in which all bits are set to the dot product of μ and K.

Final Decision. The protocol succeeds if there are at least $n - x$ rounds i for which r_i is correct and $\Delta t_i \leq t_{\max}$. The verifier then outputs a message Out_V, denoting the success or failure of the protocol.

REMARK 10. *With respect to the mapping in $\mathcal{L}$ introduced along with **SKI**, note that usual distance-bounding protocols would employ $\mathcal{L} = \mathcal{L}_{\mathsf{classic}}$ i.e., the set containing a single function L which is the identity function. Thus, in those case, $L(K) = K$ (imposing further that $\kappa = n$). The value of x also introduced along with **SKI** is used to tolerate some level of noise in the time-critical exchanges. However, introducing this tolerance brings a new type of terrorist fraud, as it will be discussed in Section 18.5. The purpose of $\mathcal{L} = \mathcal{L}_{\mathsf{bit}}$ is precisely to defeat this attack. But, to compare with other protocols, our analyses below assume $x = 0$ and $\mathcal{L} = \mathcal{L}_{\mathsf{classic}}$.*

REMARK 11. *The function F is essential for the **SKI** protocols. Using a different function leads to different protocol security achievements. Specifically, the authors mainly refer to the efficient cases of $\mathfrak{q} = 2$, $t' = 2$, and $F(1, a_i, K'_i) = (a_i)_1$, $F(2, a_i, K'_i) = (a_i)_2$, and $F(3, a_i, K'_i) = K'_i + (a_i)_1 + (a_i)_2$, where $K'_i \in GF(2)$, $(a_i)_j \in GF(2)$, $j = 1, 2$. Generally speaking, this response function, denoted F_{xor}, can be given as follows: $F_{xor}(c_i, a_i, K'_i) = K'_i 1_{c_i=t} + (a_i)_1 1_{c_i \in \{t,1\}} + \ldots + (a_i)_{t-1} 1_{c_i \in \{t,t-1\}}$, where $c_i \in \{1, \ldots, t\}$, $K'_i \in GF(\mathfrak{q})$, $\mathfrak{q} \geq 2$, $(a_i)_j \in GF(\mathfrak{q})$, $j \in \{1, \ldots, t-1\}$, and 1_R is 1 if R is true and 0 otherwise.*

The authors actually consider two variants SKI_{pro} with $t = 3$ and SKI_{lite} with $t = 2$, namely SKI_{lite} never uses the $c_i = 3$ challenge. Other cases (treated separately) are summarized as follows:

- *SKI_4: defined by the response-function F_{xor} above, with $\mathfrak{q} = 2$, $t = 4$, $t' = 3$, i.e., $F(c_i, a_i, x_i) = (a_i)_{c_i}$ for $c_i \in \{1, 2, 3\}$ and $F(4, a_i, K'_i) = K'_i + (a_i)_1 + (a_i)_2 + (a_i)_3$, with $(a_i)_1, (a_i)_2, (a_i)_3, K'_i \in GF(2)$;*
- *SKI_{shamir}: defined by a variant of response-function based on the Shamir secret sharing, with $\mathfrak{q} = 4$, $t = 3$, $t' = 2$, i.e., $F(c_i, a_i, K'_i) = K'_i + (a_i)_1 \bar{c}_i + (a_i)_2 \bar{c}_i^2$ for $\bar{c}_i \in GF(4)^*$, with $(a_i)_1, (a_i)_2 \in GF(4)$. Here, $c \mapsto \bar{c}$ denotes a one-to-one mapping from $\{1, 2, 3\}$ to $GF(4)^*$.*

While SKI_{pro} can be presented as a variant of the TDB protocol proposed in [5] and SKI_{lite} is very similar to the Hancke and Kuhn protocol [42], other variants of F can be suggested, yielding different **SKI** *protocols. These functions have to respect the requirements provided in [14]. These are informally summarized in Remark 12.*

REMARK 12 (REQUIREMENTS FOR THE FUNCTION F AND THE SET $\mathcal{L}$; (SEE [14] FOR DETAILS)). *The F function must comply to the following conditions, in order to ensure security, as stated in Section 1.6.*

(1) *For any c_i, $F(c_i, \cdot, \cdot)$ must be $GF(\mathfrak{q})$-linear and non-degenerate in the a_i part.*
(2) *For any two values c_i and c'_i of the i-th challenge and for any a_i, $F(c_i, a_i, K'_i)$ and $F(c'_i, a_i, K'_i)$ give no information about K'_i.*

(3) *For any a_i, one can compute K'_i from the table of the map $c_i \mapsto F(c_i, a_i, K'_i)$.*
(4) *For any K'_i, the largest preimage of $c_i \mapsto F(c_i, a_i, K'_i)$ must be small, on average over a_i.*

The third requirement above is used for resistance to terrorist fraud. Note that SKI_{lite} does not satisfy it, so it does not resist to terrorist fraud. The requirement on $\mathcal{L}$ is that given a source generating some $(L, L(K) + e)$ for $L \in \mathcal{L}$ uniformly distributed and e of "small" Hamming weight, and arbitrary distribution, then K can be reconstructed.

18 SUPPLEMENTARY MATERIALS: APPENDIX B

This section presents generic improvements that can be applied on distance-bounding protocols.

18.1 MUltiState Enhancement: MUSE

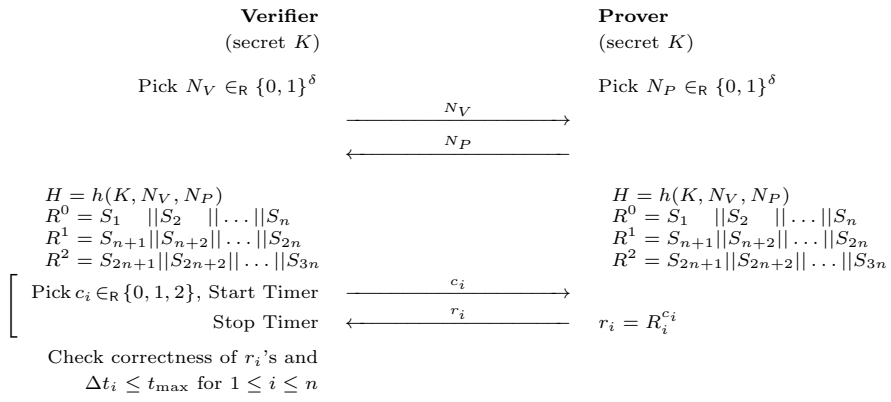
Although location-based authentication services that measure the round trip time of entire data packets have been proposed [75], most of the distance-bounding protocols are based on the measurement of the round trip time of 1-bit messages. Munilla and Peinado [54, 55] initiated a new family of protocols that use an additional third state during the fast phase. Although binary data are still exchanged during that phase, Munilla and Peinado suggest to use void challenges. These void challenges, which means that no challenge is sent, are used to authenticate the verifier, reducing thus the success probability of a pre-ask strategy.

MUSE is a generalization of this idea proposed by Avoine, Floerkemeier, and Martin [3], where the number of possible states used during the fast phase can be still larger: the authors indeed extend the concept of void challenges to p-symbols where $p \geq 2$. Using p-symbols is a generic technique that reduces the number of rounds during the fast phase. Algorithm 13 describes MUSE-3 HK, which is the 3-symbol variant of HK. In MUSE-3 HK, $H = h(K, N_V, N_P)$ is used to fill up three registers R^j $(j = 0, 1, 2)$ that each contains n 3-symbols $\{S_{jn+1}, ..., S_{jn+n}\}$. When considering the mafia fraud against MUSE-3 HK, the success probability is $\Pr_{\mathrm{MF|pre}} = \left(\frac{5}{9}\right)^n$, which is better than the 3-symbol protocol of Munilla and Peinado [54, 55]. Note that to be able to easily generate and store p-symbols $(p > 2)$ on prover side the authors suggested to encode challenges and responses on $\lfloor \log_2(p) \rfloor$ bits.

18.2 PUF-based protocols

Kardaş, Kiraz, Bingöl, and Demirci introduce in [46] two novel distance-bounding protocols based on Physically Unclonable Functions (PUFs). A PUF is defined as an unclonable function embedded in a physical structure that is easy to implement but practically impossible to duplicate, even given the exact manufacturing process definitions. The output of the function is obtained as a result of inherent physical properties such as delays of gates and wires in a circuit, variations in the temperature and supply voltage. The unclonability of the function is guaranteed by these physical processes, and some mechanisms (e.g., Fuzzy Extractors) are used to ensure the determinism. Since PUFs behave as a random function (if one assumes that all the physical properties cannot be predicted), without

Algorithm 13: Hancke and Kuhn's Protocol with MUSE-3



having the actual PUF circuit it is hard to predict the outputs as given the inputs. Moreover, their intrinsic structure yields resistance against tampering since physically tampering will most likely change its physical structure.

The authors define a strong adversary model in which the adversary has access to volatile memory of the prover, namely an RFID tag. PUF functions are used to prevent an adversary from obtaining the long-term secrets and clone the tags. The main idea is that long-term secrets are not stored in the memory of the prover but they are reconstructed from pre-secrets using a PUF circuit during each protocol execution.

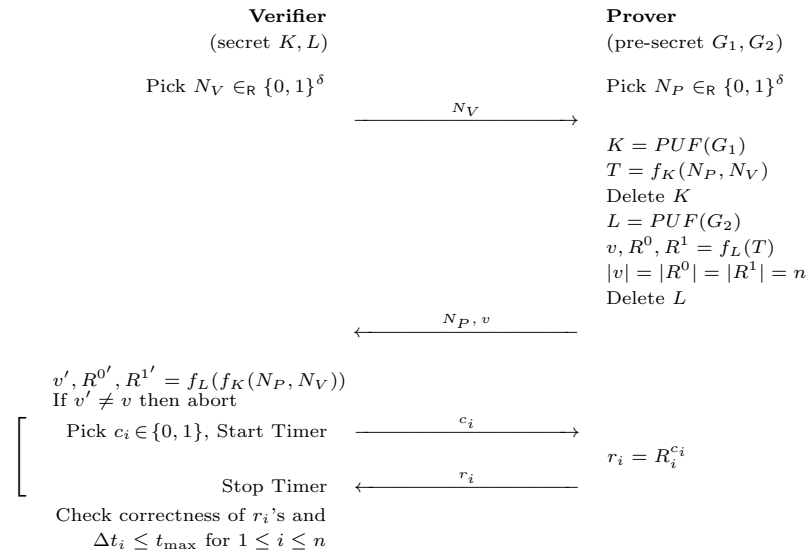
The first protocol proposed by Kardaş et al. is described in Algorithm 14. They use two different long-term keys K and L which are consecutively generated as outputs of the PUF function. Note that K and L never appear in the volatile memory at the same time. First, K is constructed by using PUF, and then completely deleted from the memory after being used as a key of PRF function. Then similarly, L is generated and deleted after generation of registers. Hence, whenever an adversary tampers the tag she can only obtain one of the keys, under the assumption that the structure of the PUF circuit has been destroyed after the attack thus PUF cannot be re-evaluated anymore. The authors state that since the adversary cannot retrieve all the long-term keys, she can only perform the attack in black-box model.

Given that the success probability of mafia and terrorist frauds remains high, namely $(3/4)^n$, the authors introduce an extended protocol with a final signature that reaches $(1/2)^n$ against these frauds.

18.3 Threshold Distance-Bounding Protocol to Defeat Terrorist Fraud

Many distance-bounding protocols are subject to terrorist fraud as the long-term key cannot be retrieved in practice from the information needed to successfully pass the protocol. Avoine, Lauradoux, and Martin in [5] suggest that a secret-sharing scheme, possibly based on threshold cryptography can be used to thwart terrorist fraud. In their proposal, the authentication material consists of p shares

Algorithm 14: Kardaş et al.'s protocol based on PUF without final signature



of a (p, k) threshold scheme: if the prover reveals any combination of k shares to the adversary, the long-term secret leaks. By contrast, gathering strictly less than k shares reveals no information about the secret.

To illustrate this, the authors describe a variant of HK, which they call TDB (Threshold Distance-Bounding), where the responses to the challenges are generated using a threshold scheme. This protocol differs from HK in the way the registers are generated during the slow phase: after the nonce exchange, verifier and prover use their shared secret K to compute a $p \times n$ matrix $\mathcal{R}$ over a group G. The matrix $\mathcal{R}$ is used to respond to the challenges as follows. The verifier requires the prover during the i-th round the value $r_{c_i,i}$ in $\mathcal{R}$ (c_i-th row and i-th column). The challenges consequently consist of $\lceil \log_2 p \rceil$ bits and the responses of $\lceil \log_2 |G| \rceil$ bits. The calculation of $\mathcal{R}$ is such that the knowledge of any combination of k elements of a given column reveals a coordinate of the key.

$$\mathcal{R} = \begin{pmatrix} r_{1,1} & \dots & r_{1,n} \\ \vdots & \ddots & \vdots \\ r_{p,1} & \dots & r_{p,n} \end{pmatrix}$$

The authors introduce in [5] three classes of adversaries: i.) `BD-ADV` or blind-adversary, who does not learn whether the protocol succeeds, ii.) `RE-ADV` or result-adversary, who can observe if the protocol succeeds and, iii.) `RD-ADV` or round-adversary, who has the capability of observing the result of each round. They then analyze the resistance of their approach when facing each of these adversaries, according to the parameters p and k. The parameter p is actually critical regarding mafia fraud, while k impacts the probability of a successful terrorist fraud.

For BD-ADV, the maximum number of elements of a column of $\mathcal{R}$ which can be safely given to the adversary is $k-1$. As a result, and for this adversary, TDB implemented with $(p,2)$ threshold scheme is secure against terrorist fraud (this probability coincides with that for the mafia fraud) for any $p \geq 2$.

When the other adversaries are considered, the post-ask strategy must be analyzed. These adversaries can learn two elements of each column of $\mathcal{R}$ for each protocol round, modifying all the challenges c_i received from the verifier and sending the modified versions $\widehat{c_i}$ to the prover; i.e., $\forall i\ \widehat{c_i} \neq c_i$. If a round succeeds, then $\widehat{r_{c_i}} = r_{c_i}$. The RD-ADV can do this on all rounds in parallel, while RE-ADV is limited to a single round per attack. So, TDB should be used with $k \geq 3$ if we want to protect the key against those stronger adversaries. On the other hand, the prover should give to the adversary at most $k-2$ shares at each round (and not $k-1$ as when BD-ADV was analyzed). Thus, in the context of RE-ADV and RD-ADV, to be secure against terrorist fraud attack, schemes $(p,3)$ for any $p \geq 3$ should be used.

The authors also describe a variant, called TTDB, that reduces the number of systems of shares computed. Whereas a column of $\mathcal{R}$ is used only once in TDB, the same column is used q times in TTDB. TTDB actually differs from TDB on three points: i.) The size of prover's answers; TTDB works on vectors of q coordinates in G, and therefore the responses of the prover are elements in G^q. ii.) The matrix computation; each distinct column is repeated q times in the matrix. The overall number of rounds is kept constant n, and consequently there are only n/q distinct columns in $\mathcal{R}$. The resulting $p \times n$ matrix $\mathcal{R}$ over G^q is defined by:

$$\begin{pmatrix} \overbrace{r_{1,1}...r_{1,1}}^{q \text{ times}} & \dots & \overbrace{r_{1,n/q}...r_{1,n/q}}^{q \text{ times}} \\ \vdots & \ddots & \vdots \\ r_{p,1}...r_{p,1} & \dots & r_{p,n/q}...r_{p,n/q} \end{pmatrix}$$

Finally: iii.) when working on a given distinct column of $\mathcal{R}$, the challenges c_i are not allowed to be repeated.

The results show that TTDB is a generalization of TDB for the terrorist fraud. For BD-ADV, TTDB is secure when $q = k-1$. Stronger adversaries, with the post-ask strategy, can recover at most $2q$ shares for round. Therefore $(p, 2q+1)$ threshold schemes should be used, and the prover, when colluding with the adversary, should only reveal q shares. For these values, TTDB is also secure against terrorist fraud.

18.4 Previous-Challenge Dependent Protocols

Previous-challenge dependent distance-bounding protocols are analyzed by Kara, Kardaş, Bingöl, and Avoine in [45]. They focus on the low-cost distance-bounding protocols having bitwise fast phases and no final signature. As for the classification, they introduce the notion of k-previous challenge dependent (k-PCD) protocols where each response bit depends on the current and the k previous challenges. First, the authors analyze the case $k = 0$, that is when each response bit depends on the current challenge only, and the case $k = 1$. They show that

the latter provides a better security than the former one and propose a natural extension to transform 0-PCD protocols into 1-PCD protocols. This modification consists in a simple polynomial arithmetic operation to compute the responses.

The authors show that mafia fraud and distance fraud are correlated by providing trade-off curves between the security levels of these two attacks. They give the theoretical security bounds for two classes: 0-PCD and 1-PCD. The authors thus claim that protocols can be designed to enforce the mafia or distance fraud resistance, but not both at the same time, without increasing the memory needs. For $k = 0$ they find that $Pr_{\mathrm{MF}}(R) + Pr_{\mathrm{DF}}(R) \geq 3/2$, where $Pr_{\mathrm{MF}}(R)$ and $Pr_{\mathrm{DF}}(R)$ are the maximum probabilities for an adversary of correctly guessing one bit response for mafia fraud and distance fraud respectively. As a consequence of this result, one can conclude that protocols with $k = 0$ cannot attain the ideal security against distance fraud, i.e., $Pr_{\mathrm{DF}}(R) = 1/2$, without being totally vulnerable against mafia fraud; and also that the security of mafia fraud cannot be better than $3/4$.

The optimal security limit for mafia fraud and the trade-off curve for protocols with $k = 1$ turn out to be $Pr_{\mathrm{MF}}(R) \geq 5/8$ and $Pr_{\mathrm{MF}}(R) + Pr_{\mathrm{DF}}(R) \geq 5/4$ respectively, and therefore it lies below that the previous one for $k = 0$. Thus, the ideal security level against distance fraud can be reached with $Pr_{\mathrm{MF}}(R) \geq 3/4$.

Finally, the authors apply the natural extension to HK for improving distance fraud resistance in one case, and for improving mafia fraud resistance in the other case[6].

The authors leave as an open question to construct trade-off curves for $k \geq 2$, but they conjecture that the security should be enhanced when k is increased.

18.5 Distance bounding over noisy channels

Distance-bounding protocols are conducted over noisy wireless *ad hoc* channels. The fast phase consists, for the most part, of single bits sent between the prover and the verifier. Due to the unreliability of the channel, the communicating parties might receive erroneous bits during this phase. Being robust to relatively high bit-error rates is a desirable property for a distance-bounding protocol.

There are two main approaches in the literature to make distance-bounding protocols noise-resilient, both requiring to increase the number of rounds during the fast phase.

The first and easiest approach to deal with noise is to allow up to x incorrect responses during the fast phase: the distance-bounding protocol succeeds if at least $(n - x)$ bit-responses sent by the prover are correct. This technique can be easily applied when the correctness of each of the n responses can be verified independently, which is the case for most distance-bounding protocols.

The second approach consists of using an error correcting code. It can be applied on many protocols but it is particularly useful in protocols where one single bit error does not allow the verifier to check the correctness of the other rounds (e.g., in BC protocol). The idea is to apply an (n, k) error correcting code on a bitstring of length k, which is used by the prover to compute the responses

[6]Note that there is a typo in [45], where it should be $y^i_{c_i} \oplus y^{i-1}_{\overline{c}_{i-1}}$ instead of $y^i_{c_i} \oplus y^i_{\overline{c}_{i-1}}$.

in the fast phase (e.g., to compute a XOR of the i-th bit of this bitstring and the challenge). The error correcting code is constructed in such a way that it can correct at least x bit errors. By applying this code to the bitstring, its length increases to n bits. These n bits are then used in the fast bit exchange phase. After this phase, the verifier applies the error correcting code to compute and verify the original bitstring of k bits. Note that only the parameter k has an influence on the security, in contrast to n.

Note that most distance-bounding protocols can be easily made noise-resilient by applying one of the two approaches. The second approach can be used by BC and MAD protocols, while the first approach can be easily applied on most other distance-bounding protocols. However, it seems harder to make DBPK-Log, Tree-based, Poulidor, and RC protocols noise-resilient.

When implementing a noise-resilient distance-bounding protocol, it is of the utmost importance to accurately estimate the bit error rate expected during the fast phase. If the estimation on the number x of expected bit errors is lower than the actual bit error rate then the *false rejection ratio* is significant, meaning that some honest provers are not accepted by the verifier. However, a high x affects the security level of the protocol in a negative way: an attacker can guess some responses wrongly, and blame it on the noise. Consequently, when analyzing the security properties of a noise-resilient distance-bounding protocol, it is typically assumed that no noise is present during the fast phase, but the verifier allows up to x bit errors. This is the worst case scenario. The success probability of an attacker depends on x. This often makes the analysis more complex and the comparison of various distance-bounding protocols difficult. Noise resilience has consequently not been considered in the analyses of the protocols provided in Sections 3 to 14.