An Efficient Distance Bounding RFID Authentication Protocol: Balancing False-Acceptance Rate and Memory Requirement

Gildas Avoine[1] and Aslan Tchamkerten[2]

[1] Université catholique de Louvain, Louvain-la-Neuve, Belgium
[2] Telecom ParisTech, France

Abstract. The Mafia fraud consists in an adversary transparently relaying the physical layer signal during an authentication process between a verifier and a remote legitimate prover. This attack is a major concern for certain RFID systems, especially for payment related applications.
Previously proposed protocols that thwart the Mafia fraud treat relaying and non-relaying types of attacks equally: whether or not signal relaying is performed, the same probability of false-acceptance is achieved. Naturally, one would expect that non-relay type of attacks achieves a lower probability of false-acceptance.
We propose a low complexity authentication protocol that achieves a probability of false-acceptance essentially equal to the best possible false-acceptance probability in the presence of Mafia frauds. This performance is achieved without degrading the performance of the protocol in the non-relay setting. As an additional feature, the verifier can make a rational decision to accept or to reject a proof of identity even if the protocol gets unexpectedly interrupted.

Key words: authentication, false-acceptance rate, proximity check, mafia fraud, memory, relay attack, RFID

1 Introduction

Radio Frequency Identification (RFID) allows to identify and authenticate objects or subjects wirelessly, using transponders — micro-circuits with an antenna — queried by readers through a radio frequency channel. This technology is one of the most promising of this decade and is already widely used in practice (e.g., access cards, public transportation passes, payment cards, passports). This success is partly due to the steadily decrease in both size and cost of passive transponders called *tags*. The characteristics of this technology — ubiquity, low-resource, wireless — open a security breach that is seriously considered by the US National Institute of Standards and Technology, which recently published guidelines on how to securely develop RFID systems [16].

In 1987 Desmedt *et al.* [6] introduced the *Mafia fraud*[3] that defeated any authentication protocol. In this attack, the adversary successfully passes the authentication by relaying the messages between the verifier and a remote legitimate prover. When it was

[3] Sometimes referred to as 'relay attack.'

introduced, the Mafia fraud appeared somewhat unrealistic since the prover is supposed unaware of the manoeuvre.

Nowadays, the Mafia fraud is a major issue of concern for RFID systems. We illustrate this in the following example. Consider an RFID-based ticket machine in a theater. To buy a ticket, the customer needs to be close enough to the machine (RFID reader) such that his pass (RFID tag) is in the field of the machine. The pass can be kept in the customer's pocket during the transaction. A ticket is delivered by the machine if the pass is able to prove its authenticity. Assume there is a line of customers waiting for a ticket, including Alice the victim. Bob and Charlie are the adversaries: Bob is far in the queue close to Alice, while Charlie faces the machine. When the machine initiates the transaction with Charlie's card, Charlie forwards the received signal to Bob who transmits it to Alice. The victim's tag automatically answers since a passive RFID tag — commonly used for such applications — responds without requiring the agreement of its holder. The answer is then transmitted back from Alice to the machine through Bob and Charlie who act as relays. The whole communication is transparently relayed and the attack eventually succeeds: Alice pays Charlie's ticket. Note that Bob must be close to the victim in order to query her tag. In such an application, the communication distance is either a few centimeters (when the tag is ISO 14443-compliant [13]) or a few decimeters (when the tag is ISO 15693-compliant [14]). This is more than enough to enable an adversary to illegitimately query the tag of a passerby. In 2005, Hancke [9] successfully performed a Mafia fraud against an RFID system where the two colluders where 50 meters apart and connected through a radio-channel.

In 2007, Halváč and Rosa [8] noticed that the standard ISO 14443 [13] for proximity cards and widely deployed in secure applications, can easily be abused by a Mafia fraud due to the untight timeouts in the communication. Indeed, ISO 14443 specifies a *frame waiting time* (FWT) such that the reader is allowed to retransmit or give up the communication if the queried tag remains unresponsive while the FWT is over. The FWT is equal to FWT $= (256 \times 16/\text{fc}) \times 2^{\text{FWI}}$, where fc is the frequency carrier (13.56 MHz in almost all secure RFID applications), and where FWI is the Frame Waiting time Integer, a value chosen between 0 and 14. By default FWI = 4, which means that FWT = 4.8 ms. However, when the tag needs more time to process the information it receives, it can impose the reader to increase the FWI up to 14, which corresponds to FWT = 4949 ms. (This feature is used for example by electronic passports that implement active authentication [11]. Passports are not able to compute an RSA or ECC signature on the fly within 4.8 ms and so require a larger FWT.) During a Mafia fraud the adversary can request the reader to increase its timeout up to 4949 ms, which gives her enough time to perform the attack over a long distance using for instance Internet.

2 State of the Art and Contributions

In 1990 Brands and Chaum [2] proposed a protocol that thwarts the Mafia fraud and which is based on the idea of a *proximity check* introduced in [1]. The protocol, depicted in Figure 1, consists of a *fast phase* followed by a *slow phase*. During the fast phase, the verifier and the prover exchange random one-bit messages and the verifier measures the round trip time (RTT) of the exchanges. After n rounds, where n is a security parameter,

the slow phase is engaged. The verifier asks the prover to sign the received and sent bits, and, upon reception of the signature, and given the measured RTT, the verifier decides whether or not to accept the proof of identity. The probability that a Mafia fraud succeeds is then $(1/2)^n$.

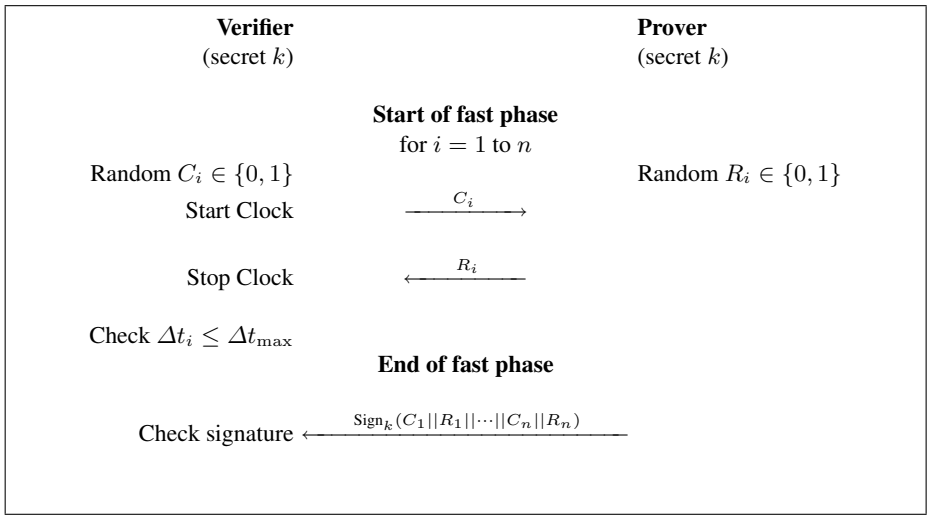

Fig. 1. Brands and Chaum's protocol

It is only in 2005, after Hancke put into practice a mafia fraud [9] that proximity check protocols[4] came back under the spotlights. The same year, Hancke and Kuhn [10] published a new distance bounding protocol that is today a key reference. Depicted in Figure 2, their protocol consists of a slow phase followed by a fast phase. In the slow phase, the verifier and the prover first exchange random nonces, then, based on the nonces and the secret key, they compute two secret registers in the form of n-bit strings V and W. The fast phase consists of n rounds. During the ith round, the verifier sends a random bit and the prover answers the ith bit V_i of V if the challenge is 0, and the ith bit W_i of W if the challenge is 1.

As explained in [10], the false-acceptance rate (FAR) is $(3/4)^n$ instead of $(1/2)^n$, as in Brands and Chaum's protocol, because an adversary can query the prover between the slow phase and the fast phase in order to obtain one full register. However, the protocol has interesting properties such as the absence of a signature at the final stage which allows the verifier to make a 'rational' decision on whether to accept or to reject a proof of identity even in cases where the protocol gets unexpectedly interrupted. In practice, one could imagine the situation where the verifier accepts a proof of identity provided that a minimal number of correct fast phase replies are given, so that to allow some flexibility in the event of an interrupted authentication. In contrast, with the Brands and Chaum protocol, if the protocol does not end properly, i.e., if the final signature is not

[4] In the literature often referred to as 'distance bounding protocols.'

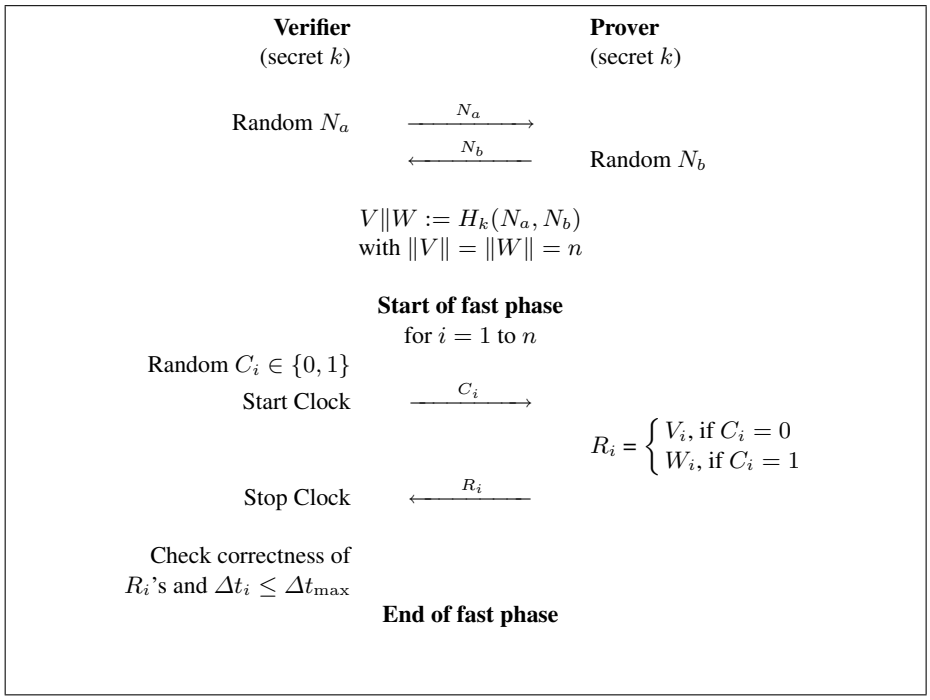

Fig. 2. Hancke and Kuhn's protocol

received by the verifier, it is difficult for the verifier to infer about the validity of the proof of identity.

Since 2005, several protocols have been proposed. Either they are based on the approach of Brands and Chaum, require a final signature, and target $(1/2)^n$ as FAR ([2–5, 17–21, 24, 25]), or, they follow the Hancke and Kuhn approach, have no final signature, and target $(3/4)^n$ as FAR ([10, 23]).[5]

Note that for both families of protocols the security is solely based on the number of fast phase rounds, which in practice cannot be made very large.[6] Moreover, for both families a FAR of $(1/2)^n$ (or $(3/4)^n$) can be achieved even without carrying a mafia fraud. In other words, these protocols do not distinguish between an attacker that relay signals from an attacker that does not relay signals. As a consequence, because n can't be made large, these protocols are not suitable for applications where a high level of security is demanded, yet mafia frauds are hard to perform.

Below we provide a new low complexity distance bounding protocol that, in particular, combines the advantages of the BC and HK families. It does not require a final signature, it achieves a FAR essentially equal to $(1/2)^n$ in the presence of Mafia frauds, and achieves the same level of security with respect to non-Mafia type of attacks as common challenge-response authentication protocols (e.g., compliant with ISO 9798 [15]).

[5] A comparison of most of these protocols is given in [17].

[6] To the best of our knowledge, distance bounding protocol haven't been implemented yet.

3 Protocol

3.1 Protocol requirements and assumptions

In the presence of a legitimate prover, the authentication protocol must guarantee that the verifier always accepts his proof of identity. The protocol must also prevent an adversary of being falsely identified, assuming she can participate either passively or actively in protocol executions with either or both the prover and the verifier. This means that the adversary can both eavesdrop protocol executions between the legitimate prover and the verifier (passive attack), and be involved in protocol executions with the verifier and the legitimate prover separately or simultaneously (active attack). We assume that neither the prover nor the verifier colludes with the adversary, i.e., the only information the adversary can obtain from the prover or the verifier is through protocol executions.

3.2 Protocol description and initialization

The protocol we describe in this section may, for certain RFID applications, require too much memory. Nevertheless, to simplify the exposition, we present and analyze this version of the protocol and later (Section 5) provide a twist that allows to drastically reduce the memory requirement while not affecting the security of the protocol.

The protocol consists of a 'slow' authentication phase followed by a 'fast' proximity check phase. Both phases have their own security parameters: m (credential size) for the authentication and n (number of rounds) for the proximity check.

Initialization. Prior to the protocol execution, the legitimate prover and the verifier agree on the security parameters m and n and a common secret key k.

Authentication. The verifier first sends a random nonce N_a to the prover, in the form of a bit string. The prover then generates a random nonce N_b and, based on N_a and N_b, computes a keyed-hash value $H_k(N_a, N_b)$ whose output size is at least $m + 2^{n+1} - 2$ bits. The prover sends to the verifier both N_b and $[H_k(N_a, N_b)]_1^m$, which denotes the first m bits of $H_k(N_a, N_b)$. (The length of the bit strings N_a and N_b is discussed in Section 4.)

Proximity check. Using the subsequent $2^{n+1} - 2$ bits of the hash value $H_k(N_a, N_b)$, denoted $[H_k(N_a, N_b)]_{m+1}^{m+2^{n+1}-2}$, the prover and the verifier label a full binary tree of of depth n as follows (see Figure 4 for an example). The left and the right edges are labeled 0 and 1, respectively, and each node (except the root) is associated with the value of a particular bit in $[H_k(N_a, N_b)]_{m+1}^{m+2^{n+1}-2}$ in a one-to-one fashion.[7]

An n-round fast bit exchange between the verifier and the prover proceeds using the tree: the edge and the node values represent the verifier's challenges and the prover's

[7] To do this one can sequentially assign the bit values of $[H_k(N_a, N_b)]_{m+1}^{m+2^{n+1}-2}$ to all the nodes of the tree by starting with the lowest level nodes, moving left to right, and moving up in the tree after assigning all the nodes of the current level.

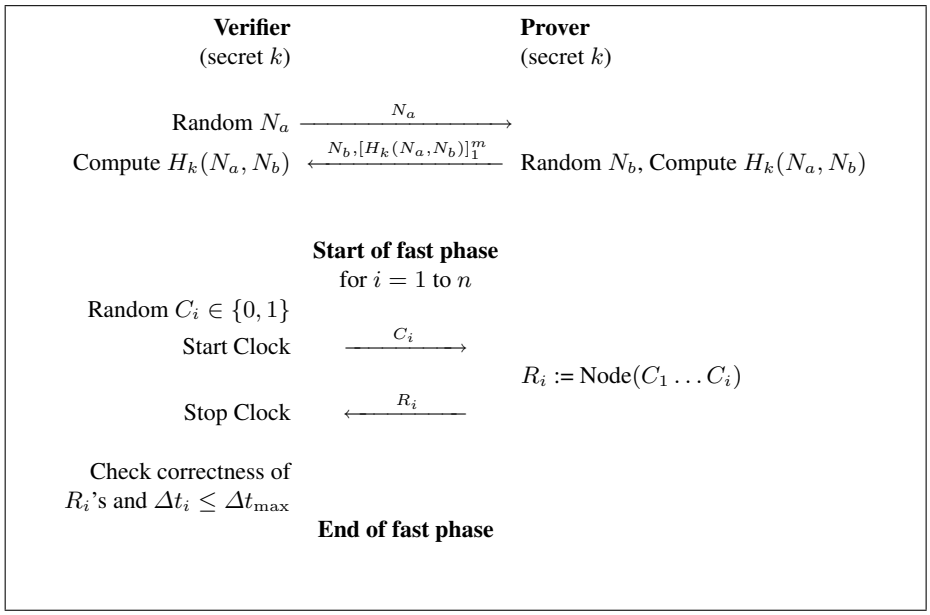

Fig. 3. Tree-based RFID distance bounding protocol.

replies, respectively. At each step $i \in \{1, 2, \dots, n\}$ the verifier generates a challenge in the form of a random bit C_i and sends it to the prover. The prover replies $R_i =$ Node$(C_1 \dots C_i)$, the value of the node in the tree whose edge path from the root is $C_1, C_2, \dots C_i$.

In the example illustrated by Figure 4, the verifier always replies 0 in the second round unless the first and the second challenges are equal to 1 in which case the verifier replies 1, i.e., Node$(00) =$ Node$(01) =$ Node$(10) = 0$ and Node$(11) = 1$.

Finally, for all $i \in \{1, 2, \dots, n\}$, the verifier measures the time interval Δt_i between the instant C_i is sent until R_i is received.

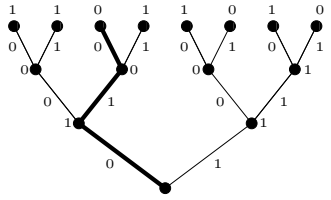

Fig. 4. Decision tree with $n = 3$. The thick line path in the tree corresponds to the verifier's challenges $0, 1, 0$ and the prover's replies $1, 0, 0$.

Final decision. The verifier accepts the prover's identity only if the m authentication bits are correct and if the n replies of the fast phase are correct while meeting the time constraint of the form $\Delta t_i \leq \Delta t_{\max}$, $i \in \{1, 2, \ldots, n\}$. A typical threshold value for $\Delta t_{\max}$ is $2d/c$, where d denotes the distance from the verifier to the expected position of the prover and where c denotes the speed of light.

4 Security Analysis

Protocols belonging to the HK family do not distinguish authentication from proximity check, which means that the security level of the proximity check is as high as the authentication one, in other words the credential parameter m is equal to the number of fast phase rounds n. While $m = 64$ is a realistic assumption,[8] $n = 64$ seems to be unpracticable due to the limited transaction time and because a proximity check over many bits seems already a practical challenge. In our protocol, authentication and proximity check are distinct. We can keep $m = 64$ while choosing a smaller n. A conservative value for the nonces' lengths is $|N_a| = |N_b| = m = 64$ bits.

We analyze our protocol by considering two cases, depending on whether or not the legitimate prover is reachable during the attack.

4.1 Attack in the absence of a legitimate prover

The case where the legitimate prover is unreachable right during the attack is similar to the classical cryptographic model. To succeed the adversary must pass both the authentication and the proximity check, without knowing the secret key. Since the hash function H_k is supposed to be cryptographically secure, we can consider that $[H_k(N_a, N_b)]_1^m$ provides no information about $[H_k(N_a, N_b)]_{m+1}^{m+2^{n+1}-2}$, i.e., the authentication reveals nothing about the proximity check and vice versa. The protocol thus achieves the same security level as any challenge-response protocol whose credential size is $m + n$ bits.

4.2 Attack in the presence of a legitimate prover

When the legitimate prover is reachable during the attack, the adversary can execute a Mafia fraud in order to successfully pass the authentication step. The FAR is then computed as follows.

Due to the time constraint, the adversary cannot usefully relay information between the verifier and the prover during the fast phase without being detected; the adversary's reply at time i must be independent of the verifier's challenge at time i, for any $i \in \{1, 2, \ldots, n\}$. However, there is no time measure before the fast phase, which allows the adversary to query the legitimate prover with one sequence of challenges $\tilde{C}^n \triangleq \tilde{C}_1 \ldots \tilde{C}_n$, hoping these will correspond to the challenges $C^n \triangleq C_1 \ldots C_n$ provided by the verifier during the fast phase.

[8] Note that attacks cannot be performed off-line.

Since the probability of false acceptance is the same given any $\tilde{C}^n$, without loss of generality we assume that the adversary queries the prover with the all-zero sequence, i.e., $\tilde{C}^n = 0^n$. The adversary is then successful only if $\tilde{R}_i = R_i$ for all $i \in \{1, 2, \ldots, n\}$, where $\tilde{R}_i$ denotes the adversary's reply at time i.

Letting t be the first time $i \geq 1$ when $C_i = 1$, we have that $\tilde{R}_i = R_i$ for $i \in \{1, 2, \ldots, t-1\}$, and $\tilde{R}_i = R_i$ with probability $1/2$ for $i \in \{t, t+1, \ldots, n\}$, because the adversary can still try her chance by sending random replies once $C_i = 1$ is observed. Therefore, letting $R^n = R_1, R_2, \ldots, R_n$, the probability of a successful attack over one particular protocol execution can be computed as

$$\begin{aligned}
\Pr(\tilde{R}^n = R^n) &= \sum_{i=1}^{n} \Pr(\tilde{R}^n = R^n | t = i) \Pr(t = i) \\
&\quad + \Pr(\tilde{R}^n = R^n | C^n = 0^n) \Pr(C^n = 0^n) \\
&= \sum_{i=1}^{n} 2^{-(n-i+1)} 2^{-i} + 2^{-n} \\
&= 2^{-n}(n/2 + 1) \, .
\end{aligned}$$

5 Multiple Trees: Balancing FAR and Memory Requirement

The second phase of the protocol is memory consuming; for n fast phase rounds we need to store

$$2^{n+1} - 2$$

bits. We now provide a means to drastically reduce this memory requirement by means of multiple trees.

Consider a fast phase based on α small trees of depth k, rather than based on a single large tree of depth $n = \alpha k$. The fast phase proceeds in the same way than described in Section 3.2 except that now the verifier accepts a proof of identity only if the k replies of each of the α trees are correct. Using multiple trees requires to store

$$\alpha(2^{k+1} - 2) \tag{1}$$

bits for the fast phase and the FAR guaranteed by the proximity check equals to

$$\left(2^{-k} \, (k/2 + 1)\right)^{\alpha} . \tag{2}$$

It is easily seen that the use of multiple trees in place of a single tree reduces the storage requirements at the expense of the false-acceptance rate. In general, among all pairs (α, k) that achieve a targeted probability of false-authentication in the presence of active attacks, one may want to pick the pair for which α is maximal so that to reduce the storage requirement. When $\alpha = 1$ and $k = n$ (single tree case), the storage requirement is maximal and the probability of false-acceptance is minimal. At the other extreme, when $\alpha = n$ and $k = 1$, the fast phase of our protocol corresponds to the Hancke and Kuhn protocol [10]. The storage requirement is minimal, equal to $2n$, and

the probability of false-acceptance is maximal, i.e., $(3/4)^n$. Finally note that, in order for the FAR of the proximity check to decay as $(1/2)^n$ instead of $(3/4)^n$, it is necessary and sufficient that k is a growing function of n.[9] Letting, for instance, $k = \log_2 n$ and $\alpha = n/\log_2 n$, the storage requirement becomes

$$\frac{2n^2}{\log_2 n}(1 - 1/n)$$

which is already a huge improvement compared to the single tree case $(2^{n+1} - 2)$ for $n \geq 2$.

The key in reducing the FAR from $(3/4)^n$, given by the Hancke and Kuhn protocol, to $(1/2)^n$ lies in the dependencies of the answers provided by the prover. In the Hancke and Kuhn protocol, the reply at time i is only a function of the ith challenge. When using trees, the ith reply potentially also depends on challenges that are posterior to the ith challenge, making it less likely for an adversary to succeed. Interestingly, the past dependency for each reply need only be 'mild': to achieve 2^{-n} it is sufficient to consider many trees each of small depth $\log_2(n)$, i.e., each reply depends at most on the last $\log_2(n)$ challenges. As a consequence, the storage requirement can be maintained low; the storage requirement grows quadratically with n instead of exponentially as in the single tree case. The bottom line is that the use of multiple trees allows to drastically reduce the storage requirement without penalizing the false-acceptance rate.

As a numerical example, to achieve a FAR of 0.01% in the presence of Mafia frauds, the Hancke and Kuhn protocol requires 32 rounds, the Brands and Chaum 14 rounds, and ours 17 rounds. With these parameters, our protocol allows to reduce the FAR down to $0.01\% \cdot 2^{-m}$ (m is typically equal to 64 or 128.) with respect to non-Mafia types of attacks, in contrast with the Hancke and Kuhn and the Brands and Chaum protocols.

For our protocol, the use of a single tree of depth 17 necessitates 32 Kbytes of memory, but a FAR of 0.01% in the presence of Mafia frauds can also be obtained by using two trees each of depth 9. This decreases the needed memory down to 256 bytes (0.25 Kbytes). For comparison, a typical chip for ePassports contains roughly 40Kbytes of EEPROM and 6Kbytes of RAM.

6 Computation

Note that only one step of the protocol involves computation, the hash value. In particular, the labeling of the nodes involves no computation, and selectors can efficiently be implemented in wired logic to directly access these values.

Tags that include a microprocessor usually embed a hash function — this is for example mandatory for tags compliant with DOC 9303 [11] which imposes SHA-1.

[9] More precisely, when k is a growing function of n, the exponential rate at which the FAR decreases with respect to n approaches one as n grows, i.e.,

$$-\frac{1}{n}\log_2(\text{FAR})$$

tends to 1 as n tends to infinity.

Note that some tags, e.g., Oberthur ID-One EPass 64 [22], implement even the SHA-256 hash function.

Tags without microprocessor usually do not implement a standardized hash function. Instead, a symmetric cipher is available, which can be either a stream cipher or a block cipher. The cipher can then be the building block of a hash function [12]. We note that in 2004, Feldhofer, Dominikus, and Wolkerstorfer [7] proposed a lightweight implementation of AES in less than $4\,000$ logic gates, enabling its implementation with wired logic only. We are not aware of commercial products using this implementation, though.

7 Concluding Remarks

The contribution of this paper consists in a low complexity tree-based RFID distance bounding protocol that combines the advantages of the protocols belonging to the Brands and Chaum's family with the advantages of the protocols belonging to the Hancke and Kuhn's family. In particular, it essentially achieves the optimal false-acceptance probability in the presence of Mafia frauds and it allows the verifier to make a rational decision even if the protocol does not end properly. In contrast with previously proposed distance bounding protocols, the security of the present protocol when the adversary can perform relay attacks does not come at the expense of the security of the protocol when the adversary cannot perform relay attacks. Our protocol achieves the same level of security with respect to non-Mafia type of attacks as common challenge-response authentication protocols.

Our protocol is suited, in terms of memory and computation, to current RFID tags designed for secure applications. It is so a solid candidate for environments where on-the-fly authentication is needed while dealing with Mafia type of frauds, e.g., in e-payment and public transportation.

References

1. Thomas Beth and Yvo Desmedt. Identification tokens – or: Solving the chess grandmaster problem. In Alfred Menezes and Scott Vanstone, editors, *Advances in Cryptology – CRYPTO'90*, volume 537 of *Lecture Notes in Computer Science*, pages 169–176, Santa Barbara, California, USA, August 1990. IACR, Springer-Verlag.
2. Stefan Brands and David Chaum. Distance-bounding protocols. In Tor Helleseth, editor, *Advances in Cryptology – EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359, Lofthus, Norway, May 1993. IACR, Springer-Verlag.
3. Laurent Bussard and Walid Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In Sasaki Ryoichi, Qing Sihan, and Okamoto Eiji, editors, *Security and Privacy in the Age of Ubiquitous Computing*, volume 181 of *IFIP International Federation for Information Processing*, pages 223–238, Chiba, Japan, May-June 2005. Springer-Verlag.
4. Laurent Bussard and Yves Roudier. Embedding distance-bounding protocols within intuitive interactions. In Dieter Hutter, Günter Müller, Werner Stephan, and Markus Ullmann, editors, *Security in Pervasive Computing – SPC*, volume 2802 of *Lecture Notes in Computer Science*, pages 119–142, Boppard, Germany, March 2003. Springer-Verlag.

5. Srdjan Capkun, Levente Buttyan, and Jean-Pierre Hubaux. SECTOR: secure tracking of node encounters in multi-hop wireless networks. In *1st ACM Workshop on Security of Ad Hoc and Sensor Networks – SASN'03*, pages 21–32, 2003.
6. Yvo Desmedt, Claude Goutier, and Samy Bengio. Special uses and abuses of the fiat-shamir passport protocol. In Carl Pomerance, editor, *Advances in Cryptology – CRYPTO'87*, volume 293 of *Lecture Notes in Computer Science*, pages 21–39, Santa Barbara, California, USA, August 1988. IACR, Springer-Verlag.
7. Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. In Marc Joye and Jean-Jacques Quisquater, editors, *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370, Boston, Massachusetts, USA, August 2004. IACR, Springer-Verlag.
8. Martin Halváč and Tomáč Rosa. A Note on the Relay Attacks on e-Passports: The Case of Czech e-Passports. Cryptology ePrint Archive, Report 2007/244, 2007.
9. Gerhard Hancke. A practical relay attack on ISO 14443 proximity cards. Manuscript, February 2005.
10. Gerhard Hancke and Markus Kuhn. An RFID distance bounding protocol. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*, Athens, Greece, September 2005. IEEE.
11. ICAO DOC–9303. Machine readable travel documents, part 1, volume 2, November 2004.
12. ISO/IEC 10118-2. Information technology – security techniques – hash-functions – part 2: Hash-functions using an n-bit block cipher.
13. ISO/IEC 14443. Identification cards – contactless integrated circuit(s) cards – proximity cards.
14. ISO/IEC 15693. Identification cards – contactless integrated circuit(s) cards – vicinity integrated circuit(s) card.
15. ISO/IEC 9798. Information technology – security techniques – entity authentication.
16. Tom Karygiannis, Bernard Eydt, Greg Barber, Lynn Bunn, and Ted Phillips. Guidelines for securing radio frequency identification (RFID) systems – special publication 800-98. Recommandations of the National Institute of Standards and Technology, April 2007.
17. Chong Hee Kim, Gildas Avoine, François Koeune, François-Xavier Standaert, and Olivier Pereira. The Swiss-Knife RFID Distance Bounding Protocol. In *International Conference on Information Security and Cryptology – ICISC*, Lecture Notes in Computer Science, Seoul, Korea, December 2008. Springer-Verlag.
18. Catherine Meadows, Radha Poovendran, Dusko Pavlovic, LiWu Chang, and Paul Syverson. *Distance Bounding Protocols: Authentication Logic Analysis and Collusion Attacks*, volume 30 of *Advances in Information Security series, Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, chapter 2, pages 279–298. Springer-Verlag, 2007.
19. Jorge Munilla, Andres Ortiz, and Alberto Peinado. Distance Bounding Protocols with Void-Challenges for RFID. Printed handout of Workshop on RFID Security – RFIDSec 06, July 2006.
20. Jorge Munilla and Alberto Peinado. Attacks on Singelee and Preneel's protocol. Cryptology ePrint Archive, Report 2008/283, June 2008.
21. Ventzislav Nikov and Marc Vauclair. Yet Another Secure Distance-Bounding Protocol. Cryptology ePrint Archive, Report 2008/319, 2008. http://eprint.iacr.org/.
22. Oberthur Card Systems. Id-one epass.
23. Jason Reid, Juan Gonzalez Neito, Tee Tang, and Bouchra Senadji. Detecting relay attacks with timing based protocols. In Feng Bao and Steven Miller, editors, *ACM symposium on Information, computer and communications security – ASIACCS*, pages 204–213, Singapore, March 2007. ACM.

24. Dave Singelée and Bart Preneel. Distance bounding in noisy environments. In *Security and Privacy in Ad-hoc and Sensor Networks – ESAS 2007*, volume 4572 of *Lecture Notes in Computer Science*, pages 101–115, 2007.
25. Yu-Ju Tu and Selwyn Piramuthu. RFID Distance Bounding Protocols. In *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.