The Poulidor Distance-Bounding Protocol

Rolando Trujillo-Rasua[1], Benjamin Martin[2], and Gildas Avoine[2]

[1] Universitat Rovira i Virgili
Department of Computer Engineering and Mathematics
Catalonia, Spain
rolando.trujillo@urv.cat
[2] Université catholique de Louvain
Information Security Group
B-1348 Louvain-la-Neuve, Belgium
{benjamin.martin, gildas.avoine}@uclouvain.be

Abstract. RFID authentication protocols are susceptible to different types of relay attacks such as mafia and distance frauds. A countermeasure against these types of attacks are the well-known *distance-bounding protocols*. These protocols are usually designed to resist to only one of these frauds, though, behave poorly when both are considered. In this paper (i) we extend the analysis of mafia and distance frauds in recently released protocols. (ii) We introduce the concept of distance-bounding protocols based on graphs while previous proposals rely on linear registers or binary trees. (iii) We propose an instance of the graph-based protocol that resists to both mafia and distance frauds without sacrificing memory. To the best of our knowledge, this protocol achieves the best trade-off between these two frauds.

Key words: RFID, authentication, distance-bounding protocol, mafia fraud, distance fraud, graph.

1 Introduction

Radio Frequency IDentification (RFID) is a contactless technology that is becoming the solution for everyday identification/authentication applications, such as access control, passport, public transportation, payment, ticketing, etc. The main purpose of RFID is to allow *readers* to communicate wirelessly with *tags* implanted into objects. While identification does not involve heavy computation capabilities for tags, authentication process, such as the ISO/IEC 9798 [2] or ISO/IEC 11770 [1] standards, requires more powerful tags performing strong cryptographic algorithms.

The most widespread and low-cost tags are *passive*, meaning that they do not have their own power source, and are supplied by the electromagnetic field of a reader. Although capacities of such tags are quite limited, some of them benefit from cryptographic building blocks and secure authentication protocols. They are typically used in the above-mentioned applications. Nevertheless, Desmedt,

Goutier and Bengio [5] presented in 1987, an attack that defeated any authentication protocol. In this attack, called *Mafia Fraud*, the adversary passes through the authentication process by simply relaying the messages between a legitimate reader (the verifier) and a legitimate tag (the prover). Thus she does not need to modify or decrypt any exchanged data. Later in 1993, Brands and Chaum [4] proposed a countermeasure that prevents from such an attack by estimating the distance between the reader and the tag to authenticate: the *distance-bounding protocol*. They also introduced in [4] a new kind of attack, named *Distance Fraud*, where a dishonest prover claims to be closer to the verifier than it really is.

Since then, many distance-bounding protocols have been proposed to thwart these attacks. In 2005, Hancke and Kuhn [6] proposed the first distance-bounding protocol dedicated to RFID. It is split in two phases: a *slow phase*, in which reader and tag exchange two nonces, and carry on resource-consuming operations; followed by a *fast phase* divided into n rounds where, in each one, the reader measures the time taken by a single bit challenge/response. Based on these exchanges, the reader is able to bound the distance between itself and the tag. These communications also provide the identity proof of the tag. Unfortunately, the adversary success probability regarding mafia and distance frauds is $(3/4)^n$ while one may expect $(1/2)^n$. Therefore, others protocols [3, 7, 8, 10–12] attempt to fix the Hancke and Kuhn's proposal.

There exist distance-bounding protocols structured differently than the one proposed by Hancke and Kuhn. For example, the protocols [4, 8, 9] perform a third additional phase in which the tag signs the exchanged bits. However, in practice this final phase represents an additional delay. As stated in [3], as the authentication entirely relies on this phase, if the latter is interrupted or not reached, then the whole process is lost. Therefore, protocols without this final slow phase are more flexible and faster. In the sequel we only focus on such protocols.

Kim and Avoine's protocol [7] and Avoine and Tchamkerten's protocol [3] are built in the same manner as Hancke and Kuhn's one. To the best of our knowledge, they have the best resistance considering only mafia fraud. However, Kim and Avoine's protocol [7] severely sacrifices the distance fraud security, whereas Avoine and Tchamkerten's one [3] requires an exponential amount of memory ($2^{n+1} - 2$ in its standard configuration) to achieve such a high mafia fraud resistance. Either Hancke and Kuhn nor the two latter protocols achieve a good balance between memory, mafia fraud resistance and distance fraud resistance.

The first contribution of this paper is the mafia and distance fraud detailed analysis of the protocols [3] and [7]. Then, we introduce the concept of distance-bounding protocols based on graphs, and we propose a new distance-bounding protocol based on a particular graph. Our goal is not to provide the best protocol in terms of mafia fraud or distance fraud, but to design a protocol that ensures a good trade-off between these concerns, while still using a linear memory. So, our protocol is never the best one when considering only one property, but is undeniably a good option when considering the three properties all together. That is why we name our protocol *Poulidor* as a famous French bicycle racer

known as *The Eternal Second* : never the best in any race, but definitively the best in average.

The paper is organized as follows. In Section 2, we describe in detail Hancke and Kuhn's protocol [6], Kim and Avoine's protocol [7] and Avoine and Tchamkerten's protocol [3]. Section 3 presents our graph-based protocol. In Section 4, we formally define the adversary strategies for mafia and distance frauds, and give a security analysis of the graph-based protocol regarding these two strategies. We show in Section 5 that our protocol has the best trade-off between mafia fraud resistance, distance fraud resistance and memory consumption. Finally, Section 6 discusses the obtained results, and raises some open problems to the scientific community.

2 State of the Art

2.1 Hancke and Kuhn's Protocol

Hancke and Kuhn's protocol (HKP) [6], depicted in Figure 1, is a key-reference protocol in terms of distance bounding devoted to RFID systems. HKP is a simple and fast protocol, but it suffers from a high adversary success probability.

Initialization The prover (P) and the verifier (V) share a secret x and agree on (i) a security parameter n, (ii) a public hash function H whose output size is $2n$, and (iii) a given timing bound $\Delta t_{\max}$.

Protocol HKP consists of two phases: a slow one followed by a fast one. During the slow phase V generates a random nonce N_V and sends it to P. Reciprocally, P generates N_P and sends it to V. V and P then both compute $H^{2n} := H(x, N_P, N_V)$. In what follows, H_i $(1 \leq i \leq 2n)$ denotes the i-th bit of H^{2n}, and $H_i \dots H_j$ $(1 \leq i < j \leq 2n)$ denotes the concatenation of the bits from H_i to H_j. Then V and P split H^{2n} into two registers of length n: $R^0 := H_1 \dots H_n$ and $R^1 := H_{n+1} \dots H_{2n}$. The fast phase then consists of n rounds. In each of them, V picks a random bit c_i (the challenge) and sends it to P. The latter immediately answers $r_i := R^{c_i}_i$, the i-th bit of the register R^{c_i}.

Verification At the end of the fast phase, the verifier checks that the answers received from the prover are correct and that $\Delta t_i \leq \Delta t_{\max}$ $(1 \leq i \leq n)$.

2.2 Kim and Avoine's Protocol

Kim and Avoine's protocol (KAP) [7], represented in Figure 2, basically relies on *predefined* challenges. Predefined challenges allow the prover to detect that an attack occurs as follows: the prover and the verifier agree on some predefined 1-bit challenges; if the adversary sends in advance a challenge to the prover that is different from the expected predefined challenge, then the prover detects the attack and until the end of the protocol execution, sends random responses to the adversary. The complete description of KAP protocol is provided below.

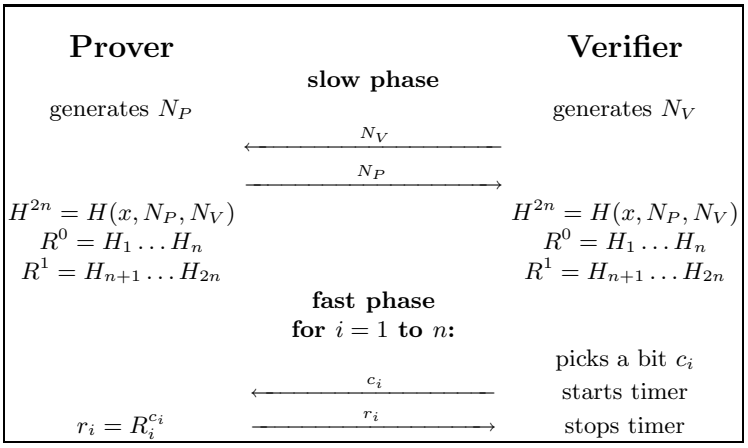

Fig. 1. Hancke and Kuhn's protocol

Initialization The prover (P) and the verifier (V) share a secret x and agree on (i) a security parameter n, (ii) a public hash function H whose output size is $4n$, and (iii) a given timing bound $\Delta t_{\max}$.

Protocol As previously, V and P exchange nonces N_V and N_P. From these values they compute $H^{4n} = H(x, N_P, N_V)$, and split it in four registers. $R^0 := H_1 \dots H_n$ and $R^1 := H_{n+1} \dots H_{2n}$ are the potential responses. The register $D := H_{3n+1} \dots H_{4n}$ constitutes the potential predefined challenges. Finally, the register $T := H_{2n+1} \dots H_{3n}$ allows the verifier (resp. prover) to decide whether a predefined challenge should be sent (resp. received): in round i, if $T_i = 1$ then a random challenge is sent; if $T_i = 0$ then the predefined challenge D_i is sent instead of a random one.

Verification At the end of the fast phase, the verifier checks that the answers received from the prover are correct and that $\Delta t_i \leq \Delta t_{\max}$ $(1 \leq i \leq n)$.

2.3 Avoine and Tchamkerten's Protocol

The Avoine and Tchamkerten's protocol (ATP) [3] is slightly different from the other existing distance bounding protocols. This protocol is also based on single bit challenge/response exchanges. However, the authors propose to use a decision tree to set up the fast phase. Figure 3 depicts the protocol detailed below.

Initialization The prover and the verifier share a secret x, agree on (i) two security parameters $n = \alpha k$ and m, (ii) a pseudo-random function PRF whose output size is at least $m + \alpha(2^{k+1} - 2)$ bits, (iii) a timing bound $\Delta t_{\max}$.

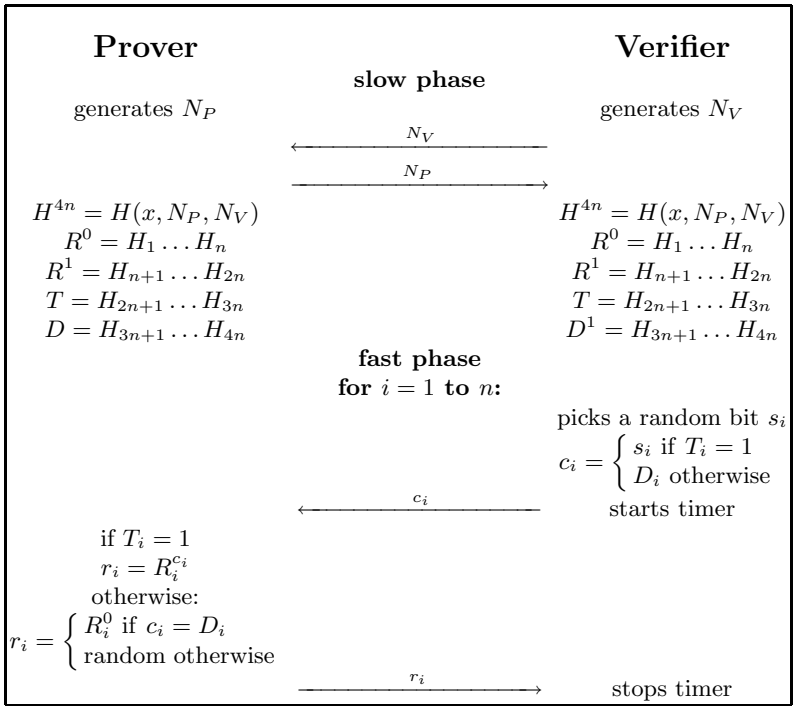

Fig. 2. Kim and Avoine's protocol

Protocol The prover P and the verifier V both generate a nonce, N_P for P and N_V for V. The verifier sends his nonce to P. Upon reception, the latter computes $PRF(x, N_P, N_V)$. He then sends $[PRF(x, N_P, N_V)]_1^m$, the first m bits of $PRF(x, N_P, N_V)$, and his nonce. These bits are used for the authentication.

P and V use the remaining $\alpha(2^{k+1} - 2)$ bits to label the nodes of α binary decision trees of depth k. Each node of the trees[3] is labeled by one bit from $[PRF(x, N_P, N_V)]_{m+1}^{m+\alpha(2^{k+1}-2)}$ (the remaining bits) in a one-to-one way. These labels represent the prover's responses during the fast phase. The challenges are symbolized by the edges of the trees, the left and right edges are labeled with 0 and 1 respectively.

Afterwards, the fast phase begins, for $1 \leq i \leq \alpha$, and $1 \leq j \leq k$, V picks a bit c_j^i at random, starts a timer and sends c_j^i to P. The latter immediately answers a bit $r_j^i = \text{node}(c_1^i, \dots c_j^i)$, the value in the i-th tree of the node relied to the root by the edges labeled $c_1^i, \dots, c_j^i$. Once V receives P's response, he stops his timer and computes Δt_j^i.

Verification The verifier authenticates the prover if the m bits, sent during the slow phase, are those he expected. The prover succeeds the distance-bounding

[3] Except the roots.

stage, if all his responses are correct and if for all $1 \leq i \leq \alpha$ and $1 \leq j \leq k$, $\Delta t_j^i \leq \Delta t_{\max}$.

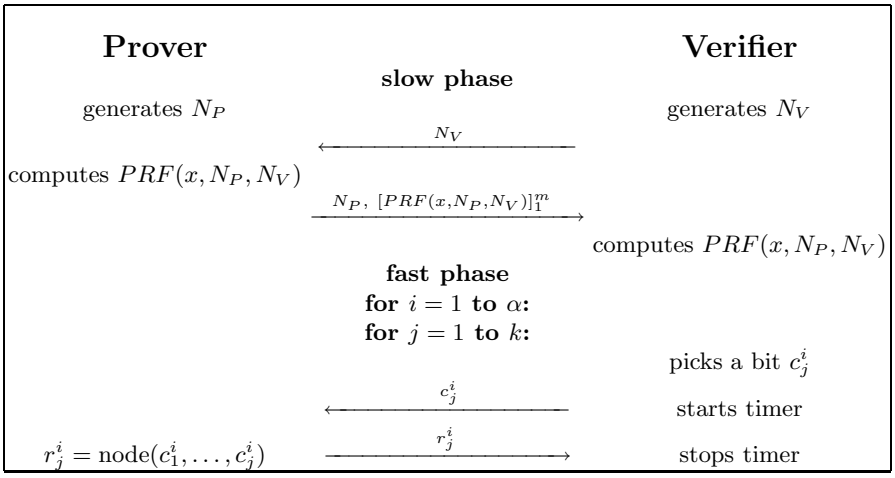

Fig. 3. Avoine and Tchamkerten protocol

3 Graph-Based Distance-Bounding Protocol

The ATP protocol [3] in its standard configuration ($\alpha = 1$) relies on a binary tree. The amount of memory needed to build this binary tree is exponential regarding to the number of rounds. Although the authors in [3] proposed to split the binary tree in order to reduce the memory requirements, they point out that this leads to a significant decrease in the security level of the protocol. We intend to go a step forward by proposing protocols based on graphs rather than trees. The graph-based protocols, as presented below, provide a greater design flexibility, a high security level and a low memory consumption.

3.1 Initialization

Parameters The prover P and the verifier V agree on four public parameters: (i) a security parameter n that represents the number of rounds in the protocol, (ii) a timing bound $\Delta t_{\max}$, (iii) a pseudo random function PRF whose output size is $4n$ bits, and (iv) a directed graph G whose characteristics are discussed below. They also agree on a shared secret x.

Graph To achieve n rounds, 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 $(0 \leq i \leq 2n-1)$ 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 4 depicts the graph when $n = 4$.

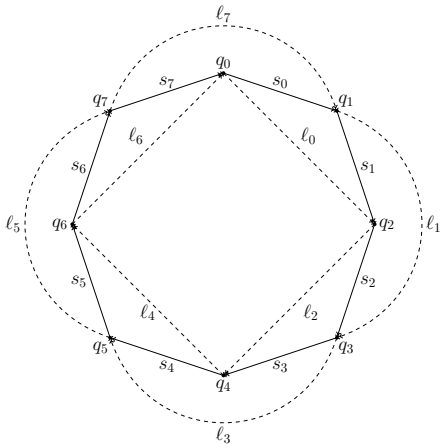

Fig. 4. Graph when $n = 4$

3.2 Exchanges

As described below, the protocol is divided in two phases, a slow phase followed by a fast one. Figure 5 summarizes the protocol.

Slow phase – P and V generate nonces N_P and N_V respectively, and exchange them. From these values and the secret x, they compute $H_1||\ldots||H_{4n} = PRF(x, N_P, N_V)$ where H_i denotes the i-th bit of the output of $PRF(x, N_P, N_V)$. The bits $H_1, \ldots, H_{4n}$ set up the 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 $(0 \leq i \leq 2n-1)$, finally $\ell_i = s_i \oplus 1$ $(0 \leq i \leq 2n-1)$.

Fast phase – This phase consists of n stateful rounds numbered from 0 to $n-1$. 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: initially $q_{p_0} = q_{v_0} = q_0$. Upon reception of the i-th challenge c_i, P moves to 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 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 $q_{v_{i+1}} = r_i$.

3.3 Verification

The authentication succeeds if all the responses are correct, and each round is completed within the time bound $\Delta t_{\max}$.

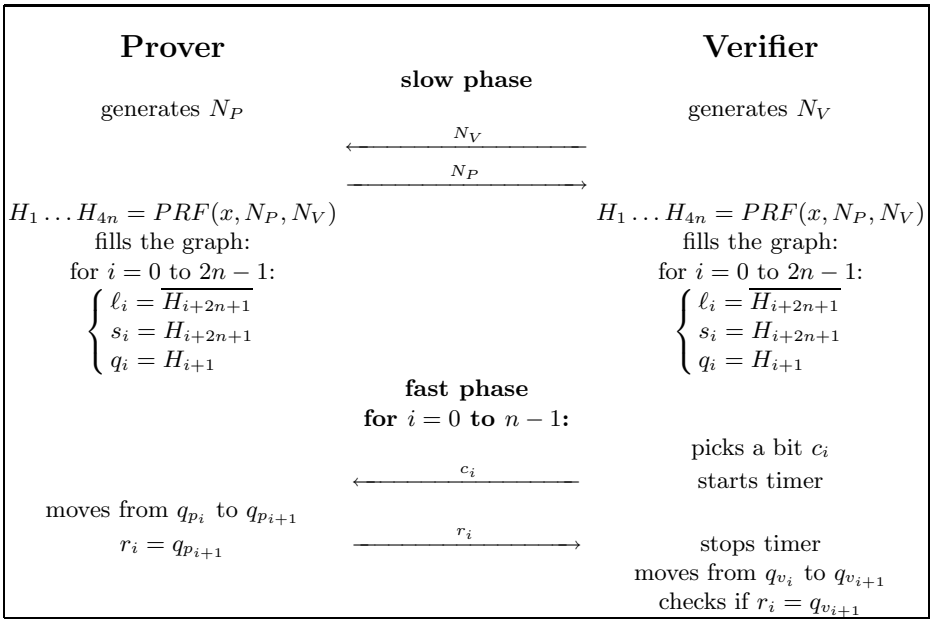

Fig. 5. Our proposal

4 Security Analysis of the Graph-Based Protocol

As stated in the introduction, mafia fraud and distance fraud are the two main security concerns when considering distance bounding protocols. We analyze in this section the graph-based protocol with respect to these frauds.

4.1 Mafia Fraud

To analyze the mafia fraud we consider the adversary abilities complying with the models provided in [3], [6] and [7]. Below, we define the *head node* and rephrase the well-known pre-ask strategy (see for example [9]) with our terminology.

Definition 1 (Head node). *Given a sequence of challenges* $\{c_1, c_2, \cdots, c_i\}$ $(1 \leq i \leq n)$*, the head node is the node that should be use by the prover to sends the response to the verifier according to this sequence of challenges. The head node is denoted as* $\Omega(c_1, c_2, \cdots, c_i)$*.*

Definition 2 (Pre-ask strategy). *The pre-ask strategy begins at the end of the slow-phase and before the beginning of the fast phase. First, the adversary sends a sequence of challenges* $\{\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_n}\}$ *to the prover and receives a sequence of responses* $\{\Omega(\tilde{c_1}), \Omega(\tilde{c_1}, \tilde{c_2}), \cdots, \Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_n})\}$*.*
Later, during the fast phase, the adversary tries to use the information obtained from the prover in the best way. Let consider $\{c_1, c_2, \cdots c_i\}$ *the challenges sent by the verifier until the* i*-th round during the fast phase. If* $\forall j$ *s.t.* $1 \leq j \leq i$*, we*

have $c_j = \tilde{c_j}$ then the adversary sends as response $\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_i})$. Otherwise she sends as response the value $\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_j})$ where j is selected according to some rule that will be defined later.

Remark 1. Sending a combination of two or more values as response is completely useless for the adversary because the nodes' values in the graph are independent from each other. Furthermore in the graph-based protocol, one node is never used twice to send a response. Therefore, the adversary can neither obtain nor infer more information than the one obtained from the prover. Finally, note that in the security analysis of previous protocols [3], [6] and [7], the best adversary strategy is to pick $j = i$ for every round, i.e. the adversary sends exactly what she received from the prover in the i-th round. However, as we explain below, in the graph-based protocol it makes sense to send a value received in a different round.

While the challenges sent by the adversary match with the challenges sent by the verifier, then the adversary is able to send the correct response. However, after the first *incorrect* adversary challenge, she can no longer be convinced about the correctness of her response. Consequently, we analyze below the adversary success probability when the adversary sends at least an *incorrect* challenge to the prover during the pre-ask strategy.

Theorem 1. *Let $(c_1, c_2, \cdots, c_i)$ be the sequence of verifier challenges until the i-th round, and let $(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_n})$ be the sequence of adversary challenges in the pre-ask strategy. Let F be the random variable representing the first round in which $c_t \neq \tilde{c_t}$ $(1 \leq t \leq n)$. Given, $\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_j})$, the adversary response in the i-th round for some $(1 \leq j \leq n)$, we have:*

$$\Pr(\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_j}) = \Omega(c_1, c_2, \cdots, c_i) | F = t) = \begin{cases} 1 & \text{if } i < t \text{ and } i = j, \\ \frac{1}{2} & \text{if } i < t \text{ and } i \neq j, \\ \frac{1}{2} & \text{if } i \geq t \text{ and } j < t, \\ p(t) & \text{if } i \geq t \text{ and } j \geq t, \end{cases}$$

where $p(t) = \frac{1}{2} + \frac{1}{2^{i+j-2t+2}} \sum_{k=0}^{k=2n-1} \left(A^{i-t}[1,k] A^{j-t}[2,k] + A^{i-t}[2,k] A^{j-t}[1,k] \right)$, and A is the adjacency matrix of the graph which represents the graph-based protocol.

Proof. We analyze the problem by cases:

Case 1 ($i < t$ and $i = j$). As $i < t$ then $\forall 1 \leq k \leq i$, $\tilde{c_k} = c_k$, therefore $\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_j}) = \Omega(c_1, c_2, \cdots, c_i)$.

Case 2 ($i < t$ and $i \neq j$). As $i < t$ then $\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_i}) = q_{v_i} = \Omega(c_1, c_2, \cdots, c_i)$. On the other hand, as $i \neq j$ then q_{v_i} and $\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_j})$ are not the same node in the graph. As the node values in the graph are independent, we conclude that, $\Pr(\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_j}) = \Omega(c_1, c_2, \cdots, c_i)) = \frac{1}{2}$.

Case 3 ($i \geq t$ and $j < t$). This case is analog to Case 2.

Case 4 ($i \geq t$ and $j \geq t$). Let be $q_{v_i} = \Omega(c_1, c_2, \cdots, c_i)$ and $q_{a_j} = \Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_j})$, so:

$$\Pr(\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_j}) = \Omega(c_1, c_2, \cdots, c_i)) = \Pr(q_{v_i} = q_{a_j}) \; . \tag{1}$$

Now, $\Pr(q_{v_i} = q_{a_j}) = \Pr(q_{v_i} = q_{a_j} | v_i = a_j) \Pr(v_i = a_j) + \Pr(q_{v_i} = q_{a_j} | v_i \neq a_j) \Pr(v_i \neq a_j)$ where $\Pr(q_{v_i} = q_{a_j} | v_i = a_j) = 1$ by definition of the graph-based protocol. On the other hand, $\Pr(q_{v_i} = q_{a_j} | v_i \neq a_j) = \frac{1}{2}$ because the node values are selected at random in the protocol, then:

$$\Pr(q_{v_i} = q_{a_j}) = \frac{1}{2} + \frac{\Pr(v_i = a_j)}{2} \; . \tag{2}$$

As $0 \leq v_i, a_j \leq 2n - 1$ then:

$$\Pr(v_i = a_j) = \sum_{k=0}^{k=2n-1} \Pr(v_i = k) \Pr(a_j = k) \; . \tag{3}$$

As $c_t \neq \tilde{c_t}$ for the first time, then two equally probable cases occur: 1) $\Omega(c_1, \cdots, c_t) = q_x$ and $\Omega(\tilde{c_1}, \cdots, \tilde{c_t}) = q_{x+1}$, 2) $\Omega(c_1, \cdots, c_t) = q_{x+1}$ and $\Omega(\tilde{c_1}, \cdots, \tilde{c_t}) = q_x$, where $(0 \leq x \leq 2n - 1)$ and $\forall x$, $x + 1 = (x + 1) \mod 2n$. Using these two events in the equation 3 we obtain:

$$\Pr(v_i = a_j) = \frac{1}{2} \left(\sum_{k=0}^{k=2n-1} \Pr(v_i = k | \Omega(c_1, \cdots, c_t) = q_x) \Pr(a_j = k | \Omega(c_1, \cdots, c_t) = q_x) \right.$$

$$\left. + \sum_{k=0}^{k=2n-1} \Pr(v_i = k | \Omega(c_1, \cdots, c_t) = q_{x+1}) \Pr(a_j = k | \Omega(c_1, \cdots, c_t) = q_{x+1}) \right) . \tag{4}$$

As $A^y[x, k]$ represents the number of walks of size y between the nodes x and k, then $\Pr(v_i = k | \Omega(c_1, \cdots, c_t) = q_x) = \frac{A^{i-t}[x,k]}{2^{i-t}}$ and $\Pr(v_i = k | \Omega(c_1, \cdots, c_t) = q_{x+1}) = \frac{A^{i-t}[x+1,k]}{2^{i-t}}$, in the same way $\Pr(a_j = k | \Omega(c_1, \cdots, c_t) = q_x) = \frac{A^{j-t}[x,k]}{2^{j-t}}$ and $\Pr(a_j = k | \Omega(c_1, \cdots, c_t) = q_{x+1}) = \frac{A^{j-t}[x+1,k]}{2^{j-t}}$. Then using Equation 4:

$$\Pr(v_i = a_j) = \frac{1}{2^{i+j-2t+2}} \sum_{k=0}^{k=2n-1} \left(A^{i-t}[x, k] A^{j-t}[x + 1, k] + A^{i-t}[x + 1, k] A^{j-t}[x, k] \right) \; . \tag{5}$$

Given the graph characteristics, we have $A^y[x, k] = A^y[(x - z) \mod 2n, (k - z) \mod 2n]$ for any $z \in \mathbb{N}$. Therefore, $A^{i-t}[x, k] = A^{i-t}[1, (k - x + 1) \mod 2n]$ and $A^{i-t}[x + 1, k] = A^{i-t}[2, (k - x + 1) \mod 2n]$, in the same way, $A^{j-t}[x, k] = A^{j-t}[1, (k - x + 1) \mod 2n]$ and $A^{j-t}[x + 1, k] = A^{j-t}[2, (k - x + 1) \mod 2n]$. So:

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

$$\sum_{k=0}^{2n-1} \left(A^{i-t}[1,k]A^{j-t}[2,k] + A^{i-t}[2,k]A^{j-t}[1,k]\right). \quad (6)$$

Equations 1, 2, 5, and 6 yield the expected result.

□

Remark 2. Using Theorem 1, assuming $c_1 \neq \tilde{c_1}$, then for $i = 1$ we obtain that $\Pr(\Omega(\tilde{c_1}, \tilde{c_2}) = \Omega(c_1)) = \frac{5}{8} > \Pr(\Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_j}) = \Omega(c_1))$ for every $j \neq 2$. It means that in this case it is better for the adversary to send the second response of the prover $(\Omega(\tilde{c_1}, \tilde{c_2}))$. These results only reinforce the ideas expressed in the Remark 1, that is the best adversary strategy is not always to pick $j = i$ in the graph-based protocol.

Corollary 1. *Given $r_i = \Omega(\tilde{c_1}, \tilde{c_2}, \cdots, \tilde{c_i})$ and $c'_i = \Omega(c_1, c_2, \cdots, c_i)$ for every $1 \leq i \leq n$, the best adversary success probability in the mafia fraud is:*

$$\sum_{t=1}^{t=n} \frac{1}{2^t} \left(\prod_{i=t}^{i=n} \max(\Pr(r_1 = c'_i | F = t), \cdots, \Pr(r_n = c'_i | F = t)) \right) + \frac{1}{2^n}$$

where $\Pr(r_j = c'_i | F = t)$ is defined in Theorem 1.

Proof. The adversary success probability in the mafia fraud is:

$$\sum_{t=1}^{t=n} (\Pr(\text{success} | F = t) \Pr(F = t)) + \Pr(c_1 = \tilde{c_1}, c_2 = \tilde{c_2}, \cdots, c_n = \tilde{c_n}). \quad (7)$$

As the challenges are selected at random, then:

$$\begin{array}{l} \Pr(F = t) = \frac{1}{2^t}. \\ \Pr(c_1 = \tilde{c_1}, c_2 = \tilde{c_2}, \cdots, c_n = \tilde{c_n}) = \frac{1}{2^n}. \end{array} \quad (8)$$

Considering the pre-ask attack strategy in Definition 2:

$$\Pr(\text{success} | F = t) = \prod_{i=t}^{i=n} \max(\Pr(r_1 = c'_i | F = t), \cdots, \Pr(r_n = c'_i | F = t)). \quad (9)$$

Equations 7, 8, and 9 yield the expected result.

□

4.2 Distance Fraud

The distance fraud analysis for most of the distance-bounding protocols is not a hard task. However, for the ATP [3] protocol, to the best of our knowledge, nobody has found the distance fraud success probability. Unfortunately, in the graph-based protocol which has some similarities with the ATP protocol, distance fraud analysis is also not trivial. Then, in this paper we provide an upper bound of the distance fraud for a sub-family of the distance-bounding protocols, which will be useful for the ATP protocol, and of course, for the graph-based protocol too.

Definition 3 (Distance-bounding protocol sub-family). *Let consider $\mathcal{P}$, a distance bounding protocol. $\mathcal{P}$ belongs to the distance-bounding protocol sub-family if it fulfills the following requirements:*

- *During the fast phase, in each round the verifier sends a bit as challenge and the prover answers with a bit alike.*
- *There is no final phase.*
- *After the slow-phase, it should be possible to build a function $f : \{0,1\}^n \rightarrow \{0,1\}^n$ such that, given any sequence of challenge $\{c_1, c_2, \cdots, c_n\}$, then $f(c_1, c_2, \cdots, c_n)$ is the correct response sequence for the verifier. Since now on, we are going to call this function as "prover function".*

Definition 4 (Prover function pre-image). *For a sequence $y \in \{0,1\}^n$ and a prover function f, the prover function pre-image is the set $I_y = \{x \in \{0,1\}^n | f(x) = y\}$.*

We now define the adversary capability in the distance fraud:

Definition 5 (Adversary capability in the distance fraud). *The adversary capability in the distance fraud is twofold:*

1. *The adversary has access to the prover function.*
2. *The adversary can send in advance a sequence $y \in \{0,1\}^n$ to the verifier, trying to maximize $\Pr(f(c_1, c_2, \cdots, c_n) = y)$ where $\{c_1, c_2, \cdots, c_n\}$ is a random sequence of challenges.*

Proposition 1. *Let y be the sequence sent by the adversary in advance, then the success probability in the distance fraud is $\frac{|I_y|}{2^n}$.*

So, the adversary strategy is pretty clear, she must find and send a sequence $y \in \{0,1\}^n$, such that for any sequence $x \in \{0,1\}^n$ it holds that $|I_y| \geq |I_x|$.

Theorem 2. *Given $x, y \in \{0,1\}^n$ two random sequences, and a prover function f, then, for any sequence $z \in \{0,1\}^n$ such that $I_z \neq \emptyset$ we have:*

$$\Pr(x \in I_z) \leq \frac{\frac{1}{2^n} + \sqrt{\frac{1}{2^{2n}} - \frac{4}{2^n} + 4\Pr(f(x) = f(y))}}{2}$$

Proof. Given that $I_z \neq \emptyset$, we have:

$$\begin{aligned}\Pr(f(x) = f(y)) = \Pr(f(x) = f(y)|y \in I_z)\Pr(y \in I_z) \\ + \Pr(f(x) = f(y)|y \notin I_z)\Pr(y \notin I_z)\end{aligned} \tag{10}$$

But, $\Pr(f(x) = f(y)|y \in I_z) = \Pr(x \in I_z) = \Pr(y \in I_z)$ because x and y are random sequences. On the other hand, $\Pr(f(x) = f(y)|y \notin I_z) \geq \frac{1}{2^n}$ because of the "prover function" definition. Therefore, using these results in Equation 10:

$$\Pr(f(x) = f(y)) \geq \Pr(x \in I_z)^2 + \frac{1}{2^n}(1 - \Pr(x \in I_z)) \,. \tag{11}$$

Calculating the discriminant of this quadratic inequality, and obtaining its solutions, we conclude the proof. Note that, this quadratic inequality has real solutions because $\Pr(f(x) = f(y)) \geq \frac{1}{2^n}$, and in this case, the discriminant value is always positive. □

Corollary 2. *For every distance-bounding protocol that complies with Definition 3, the adversary success probability in the distance fraud is upper bounded by:*

$$\frac{\frac{1}{2^n} + \sqrt{\frac{1}{2^{2n}} - \frac{4}{2^n} + 4\Pr(f(x) = f(y))}}{2} \,.$$

With this last result, we are giving a way to compute an upper bound of a sub-family of the distance-bounding protocols. We show below how it is possible to apply this result to the graph-based protocol, and later we apply the same result for the ATP protocol.

Theorem 3. *The distance fraud success probability for the graph-based protocol is upper bounded by:*

$$\frac{\frac{1}{2^n} + \sqrt{\frac{1}{2^{2n}} - \frac{4}{2^n} + 4p}}{2} \,.$$

where

$$p = \prod_{i=1}^{i=n} \left(\frac{1}{2} + \frac{1}{2^{2i+1}} \sum_{k=0}^{k=2n-1} (A^i[0,k])^2 \right) \,.$$

Proof. Let considered two random sequences $x = \{x_1, x_2, \cdots, x_n\}$ and $y = \{y_1, y_2, \cdots, y_n\}$, then by the definition of the graph-based protocol and the definition of "Prover Function":

$$\Pr(f(x) = f(y)) = \prod_{i=1}^{i=n} \Pr(\Omega(x_1, \cdots, x_i) = \Omega(y_1, \cdots, y_i)) \,. \tag{12}$$

Let be $q_{x_i} = \Omega(x_1, \cdots, x_i)$ and $q_{y_i} = \Omega(y_1, \cdots, y_i)$, then, like in Theorem1, we can obtain that:

$$\Pr(q_{x_i} = q_{y_i}) = \frac{1}{2} + \frac{\Pr(x_i = y_i)}{2} \ . \tag{13}$$

and

$$\Pr(x_i = y_i) = \sum_{k=0}^{k=2n-1} \Pr(x_i = k) \Pr(y_i = k) \ . \tag{14}$$

Once again, as $A^i[j,k]$ represents the number of walks of size i between the nodes j and k, where A is the adjacency matrix of the graph, then $\Pr(x_i = k) = \frac{A^i[0,k]}{2^i} = \Pr(y_i = k)$. Therefore, using Equation 14:

$$\Pr(x_i = y_i) = \sum_{k=0}^{k=2n-1} \left(\frac{A^i[0,k]}{2^i} \right)^2 \ . \tag{15}$$

Equations 12, 13, and 15, yield to:

$$\Pr(f(x) = f(y)) = \prod_{i=1}^{i=n} \left(\frac{1}{2} + \frac{1}{2^{2i+1}} \sum_{k=0}^{k=2n-1} (A^i[0,k])^2 \right) \ . \tag{16}$$

Applying Equation 16 to Corollary 2, considering that $p = \Pr(f(x) = f(y))$, we conclude the proof of this theorem. □

5 Comparison

In this paper we are analyzing three parameters: mafia fraud, distance fraud and memory consumption. Therefore, we need these values for each of the previous considered protocols. Unfortunately, the computation of the mafia fraud success probability for KAP protocol [7] is not correct, but in Appendix A we provide a correct calculation. On the other hand, as we previously said, ATP distance fraud success probability was not presented in [3], nevertheless, in Appendix B we give a distance fraud upper bound for this protocol exactly as we did with the graph-based protocol.

Since we consider memory consumption as a main concern in distance-bounding protocols, we relax the ATP protocol, as its authors propose, to fit with linear memory. Nevertheless, reducing the memory in ATP protocol, increases the adversary success probability for both type of fraud. Hence, we pick $\alpha = \frac{n}{3}$ in which case the memory consumption equals to $\frac{14n}{3} \approx 5n$ whereas the security is still ensured. Note that this memory consumption is in the range of the other studied protocol. This instance of the ATP protocol is named "ATP3".

Table 1 depicts the values of the three parameters for each protocols that we are considering. In terms of memory the Hancke and Kuhn protocol is, undoubtedly, the best protocol. As can be seen in Figure 6, when considering only mafia fraud resistance KAP and ATP protocols are the best ones. And only in

Table 1. This table depicts the values of the three parameters (memory, mafia fraud success probability and distance fraud success probability), for the HKP protocol, the KAP protocol, the ATP protocols (ATP and ATP3), and the graph-based protocol (GRAPH).

	Memory	Mafia Fraud	Distance Fraud
HKP	$2n$ [6]	$\left(\frac{3}{4}\right)^n$ [6]	$\left(\frac{3}{4}\right)^{n\ 2}$
KAP	$4n$ [7]	Appendix A	$\left(\frac{3}{4}+\frac{p_d}{4}\right)^n$ [7]
ATP	$2^{n+1}-2$ [3]	$\left(\frac{1}{2}\right)^n\left(\frac{n}{2}+1\right)$ [3]	Appendix B
ATP3	$\frac{14n}{3}$ [3]	$\left(\frac{1}{2}\right)^n\left(\frac{5}{2}\right)^{\frac{n}{3}}$ [3]	$(0.3999)^{\frac{n}{3}\ 3}$
GRAPH	$4n$	Corollary 1	Theorem 3

terms of distance fraud, the lowest adversary success probability is reached by the ATP protocol (see Figure 7).

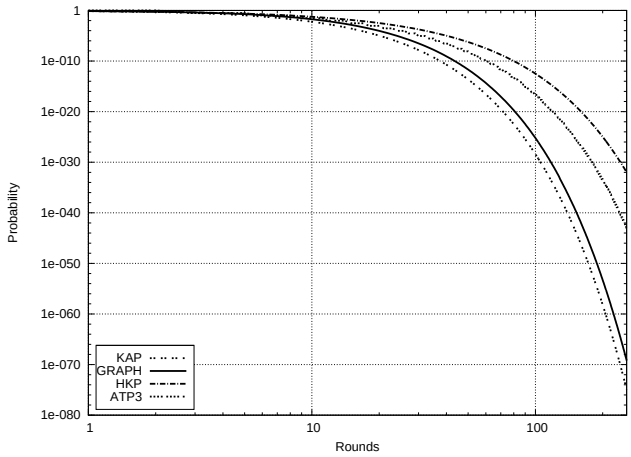

Fig. 6. In this figure we show the mafia fraud probability achieved by the GRAPH protocol, HKP protocol , and ATP3 protocol. The ATP protocol in its standard configuration is not presented in this chart because it has the same mafia fraud probability than the KAP protocol.

[2] The distance fraud probability for the HKP protocol is computed using the distance fraud probability in the KAP protocol. Note that, the KAP protocol with $p_d = 0$ and the HKP protocol are the same.

[3] The distance fraud probability for the ATP3 protocol is the accurate value and not an upper bound like in ATP or GRAPH protocols. It was computed by brute force, i.e. for a given instance, we computed the adversary success probability. Then, considering all the possible instance we deduce the probability in the average case.

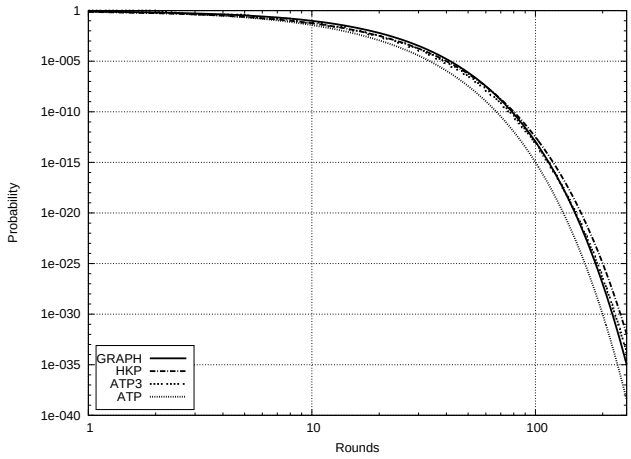

Fig. 7. In this figure we show the distance fraud probability achieved by the GRAPH protocol, HKP protocol, and the ATP protocols (ATP and ATP3). The KAP protocol was not presented in this chart because in the best case has the same distance fraud probability than the HKP protocol.

However, our interest is finding the best protocol given a security level in terms of mafia fraud and distance fraud. Therefore, Figure 8 depicts for each configuration (mafia and distance), the protocol needing a lower number of rounds to reach these security values. As it can be seen in Figure 8, the graph-based protocol is, in general, the best protocol when considering memory consumption, distance, and mafia fraud at the same time. In particular, if one requires low success probabilities for both mafia and distance fraud, we stress out the particularly good behavior of the graph-based protocol. Note that in some cases more than one protocol is optimal in terms of number of rounds, in this case the best in terms of memory is chosen.

6 Conclusions and Remarks

In this paper we take a step forward in the parameters (mafia fraud, distance fraud, and memory) for the distance-bounding protocols. In particular, we provide a way to compute an upper bound on the distance-fraud probability, which is useful for analyzing previous protocols and designing future ones. In addition, we propose a new distance-bounding protocol, and we show that the achieved security level is better than all previously published papers when considering the three parameters at the same time.

This paper do not only provide a simple, fast, and flexible protocol, but it also introduces the graph-based protocol concept and new open questions along with. First of all, an interesting question is to know if there are graph-based protocols that behave still better than the one presented here. In particular,

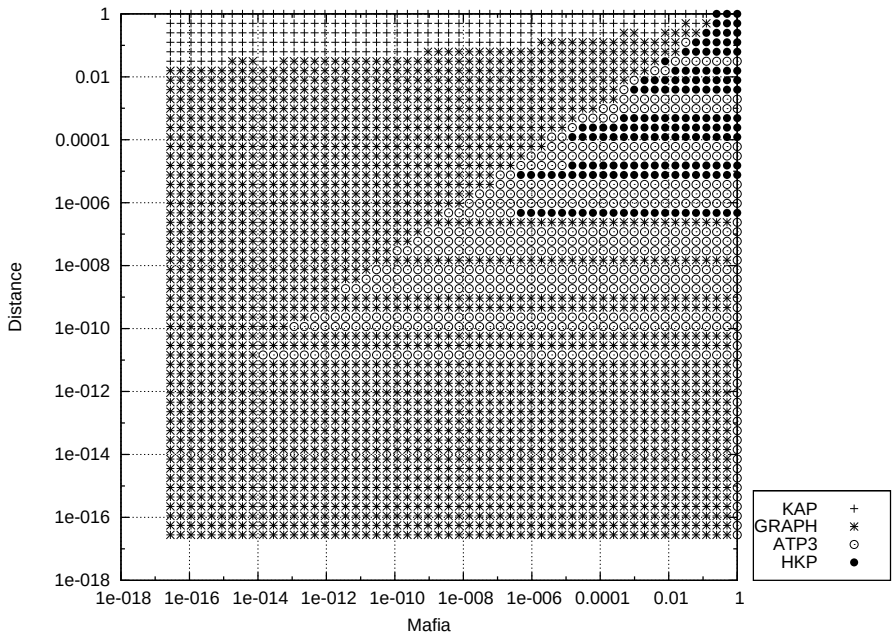

Fig. 8. In this figure we show the best protocol in terms of number of rounds given different values of mafia fraud probability and distance fraud probability. The considered protocols are: the graph-based protocol (GRAPH), the Hancke and Kuhn's protocol (HKP), the Kim and Avoine's protocol (KAP), and the Avoine and Tchamkerten's protocol (ATP3). The ATP protocol in its standard configuration is not considered in this chart because we are comparing only protocols with linear memory consumption.

if the number of rounds is not a critical parameter, prover and verifier may be allowed to increase the number of rounds while keeping a 2n-node graph. This means that some nodes may be used twice. In such a case, the security analysis provided in this paper must be refined. On the other hand, although a bound on the distance fraud success probability is provided, calculating the exact probability of success is still cumbersome.

Acknowledgments. This work is partially funded by the Spanish Government through projects TSI2007-65406-C03-01 "E-AEGIS" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", by the Government of Catalonia under grant 2009 SGR 1135, and by the Walloon Region Marshall plan through the SPW DG06 Project TRASILUX.

The authors thank to Chong Hee Kim for his support in the computation of the adversary mafia fraud probability for the Kim and Avoine's protocol [7],

Tania Martin for her precious help, Pierre François and Juan A. Rodríguez for the interesting discussions about graphs.

References

1. ISO/IEC 11770: Information technology – security techniques – key management.
2. ISO/IEC 9798: Information technology – security techniques – entity authentication.
3. Gildas Avoine and Aslan Tchamkerten. An efficient distance bounding RFID authentication protocol: balancing false-acceptance rate and memory requirement. In *Information Security Conference – ISC'09*, volume 5735 of *Lecture Notes in Computer Science*, Pisa, Italy, September 2009.
4. Stefan Brands and David Chaum. Distance-bounding protocols. In *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 344–359, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.
5. Yvo Desmedt, Claude Goutier, and Samy Bengio. Special uses and abuses of the fiat-shamir passport protocol. In *CRYPTO*, pages 21–39, 1987.
6. Gerhard Hancke and Markus Kuhn. An RFID Distance Bounding Protocol. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*, pages 67–73, Athens, Greece, September 2005. IEEE, IEEE Computer Society.
7. Chong Hee Kim and Gildas Avoine. RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks. In *8th International Conference on Cryptology And Network Security – CANS'09*, Kanazawa, Ishikawa, Japan, December 2009. Springer.
8. Chong Hee Kim, Gildas Avoine, François Koeune, François-Xavier Standaert, and Olivier Pereira. The Swiss-Knife RFID Distance Bounding Protocol. In P.J. Lee and J.H. Cheon, editors, *International Conference on Information Security and Cryptology – ICISC*, volume 5461 of *Lecture Notes in Computer Science*, pages 98–115, Seoul, Korea, December 2008. Springer-Verlag.
9. Jorge Munilla, Andres Ortiz, and Alberto Peinado. Distance Bounding Protocols with Void-Challenges for RFID. In *Workshop on RFID Security – RFIDSec'06*, Graz, Austria, July 2006. Ecrypt.
10. Jorge Munilla and Alberto Peinado. Security Analysis of Tu and Piramuthu's Protocol. In *New Technologies, Mobility and Security – NTMS'08*, pages 1–5, Tangier, Morocco, November 2008. IEEE.
11. Jason Reid, Juan Gonzalez Neito, Tee Tang, and Bouchra Senadji. Detecting relay attacks with timing based protocols. In Feng Bao and Steven Miller, editors, *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security – ASIACCS '07*, pages 204–213, Singapore, Republic of Singapore, March 2007. ACM.
12. Yu-Ju Tu and Selwyn Piramuthu. RFID Distance Bounding Protocols. In *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

Appendix

A Mafia fraud success probability for KAP [7]

In the Kim and Avoine protocol the adversary success probability in the mafia fraud depends on the predefined challenges probability (p_d). Let:

- L_i be the event that the adversary win the i-th round.
- D_i be the event that the adversary is detected in the i-th round by the tag for the first time.
- N_i be the event that the adversary is detected by the tag in the i-th round, and N the event that the adversary is never detected.

Remark 3. The notation $\bar{A}$ represents the complement of the event A.

By the law of total probability:

$$P(\text{success}) = \sum_{i=1}^{i=n} \Pr(\text{success}|D_i)\Pr(D_i) + \Pr(\text{success}|N)\Pr(N)\ . \tag{17}$$

As $\Pr(N_i) = \frac{p_d}{2}$, then:

$$\Pr(N) = (1 - \frac{p_d}{2})^n\ . \tag{18}$$

The probability of being detected in the i-th round for the first time is:

$$\Pr(D_i) = \prod_{j=1}^{j=i-1} \Pr(\bar{N_j})\Pr(N_i) = \left(\frac{2-p_d}{2}\right)^{i-1}\left(\frac{p_d}{2}\right)\ . \tag{19}$$

On the other hand:

$$\Pr(\text{success}|D_i) = \prod_{j=1}^{j=i-1} \Pr(L_j|\bar{N_j}) \prod_{j=i}^{j=n} \Pr(L_j|N_j) \tag{20}$$

where $\Pr(L_j|N_j) = \frac{1}{2}$ and:

$$\Pr(L_j|\bar{N_j}) = \frac{\Pr(L_j \cap \bar{N_j})}{\Pr(\bar{N_j})}\ . \tag{21}$$

where $\Pr(L_j \cap \bar{N_j}) = \Pr(L_j \cap \bar{N_j}|p_d)p_d + \Pr(L_j \cap \bar{N_j}|p_r)p_r$. But, $\Pr(L_j \cap \bar{N_j}|p_d) = \frac{1}{2}$ because the adversary must send the correct challenges c_j in this round. And, $\Pr(L_j \cap \bar{N_j}|p_r) = \frac{3}{4}$ because this is the same case as in Hancke and Kuhn protocol. Therefore, $\Pr(L_j \cap \bar{N_j}) = \frac{1}{2}p_d + \frac{3}{4}p_r = \frac{3-p_d}{4}$. Using this result in Equation 21:

$$\Pr(L_j|\bar{N_j}) = \frac{3-p_d}{4-2p_d}\ . \tag{22}$$

using Equation 20, and 22:

$$\Pr(\text{success}|D_i) = \left(\frac{3-p_d}{4-2p_d}\right)^{i-1}\left(\frac{1}{2}\right)^{n-i+1}, \tag{23}$$

and

$$\Pr(\text{success}|N) = \left(\frac{3-p_d}{4-2p_d}\right)^{n}. \tag{24}$$

Using the equations 17, 18, 19, 23 and 24 we obtain the adversary success probability for the mafia fraud in the Kim and Avoine protocol:

$$P(\text{success}) = \frac{p_d}{2}\sum_{i=1}^{i=n}\left(\frac{3-p_d}{4}\right)^{i-1}\left(\frac{1}{2}\right)^{n-i+1} + \left(\frac{3-p_d}{4}\right)^{n}. \tag{25}$$

B Distance Fraud Success Probability for ATP [3]

To find an upper bound of the adversary success probability in the distance fraud for the ATP protocol, we use the result of the Theorem 3. Indeed, this protocol has the same behavior than the graph-based protocol. The only difference between them is that the ATP protocol create a full tree as graph. Therefore, in ATP protocol the distance fraud success probability is upper bounded by:

$$\frac{\frac{1}{2^n} + \sqrt{\frac{1}{2^{2n}} - \frac{4}{2^n} + 4p}}{2},$$

where

$$p = \prod_{i=1}^{i=n}\left(\frac{1}{2} + \frac{1}{2^{2i+1}}\sum_{k=0}^{k=2n-1}(A^i[0,k])^2\right).$$

To give a complete equation, we define $A^i[0,k]$ for a tree. For this purpose, we consider that the nodes in the tree are labeled between 0 and 2^n-1 using a breadth-first algorithm, then:

$$A^i[0,k] = \begin{cases} 1 & \text{if} \quad 2^i - 1 \leq k < 2^{i+1} - 1, \\ \\ 0 & \text{otherwise.} \end{cases}$$

Finally we obtain:

$$p = \prod_{i=1}^{i=n}\left(\frac{1}{2} + \frac{1}{2^{i+1}}\right).$$