RFID Distance Bounding Protocols with Mixed Challenges

Chong Hee Kim, Gildas Avoine

***Abstract*—RFID systems suffer from different location-based attacks such as distance fraud, mafia fraud, and terrorist fraud. Among them mafia fraud is the most serious one as it can be mounted without the awareness of neither the reader nor the tag. In such an attack, the adversary acts as a man-in-the-middle who relays the signal between the two entities, possibly without knowing the specifications of the protocol used on the channel. Recently, distance bounding protocols measuring the round-trip times of messages exchanged between the reader and the tag have been designed to prevent this attack. Almost all the existing proposals are based on binary challenges, with no final signature, and provide a mafia fraud success probability equal to $(3/4)^n$, where n is the number of rounds in the protocol, or require too much memory. In this article, we introduce new distance bounding protocols, based on binary mixed challenges, that converge toward the expected and optimal $(1/2)^n$ bound and which only require little memory.**

***Index Terms*—RFID, authentication, distance bounding protocol, relay attack, distance fraud.**

I. Introduction

RADIO-Frequency Identification (RFID) devices, which include tags and contactless smartcards, are usually passive, namely they operate without any internal battery and receive power from the reader. They offer a long lifetime and a reduced cost but suffer from limited computational and storage capabilities.

RFID-based systems are vulnerable to different location-based attacks, especially the mafia fraud [1]. Such an attack consists in an adversary making a reader believe that it is communicating with a legitimate tag, and vice versa, while this is not the case. The adversary acts as a man-in-the-middle who relays the signal between the two entities, possibly without knowing the specifications of the protocol used on the channel. Consequently, the mafia fraud cannot be prevented using a cryptographic protocol that operates only in the application layer.

One solution to mitigate this problem consists in providing to the devices a means to obtain their global location, e.g., with a GPS module. Unfortunately, the technical and cost requirements of this approach do not fit the RFID constraints. The way that is considered today to thwart the mafia fraud in RFID systems is based on the use of *distance bounding protocols*, which measure the signal strength or the communication time. Measuring the signal strength is not secure as the adversary can easily amplify the signal. Therefore, in this article, we consider distance bounding protocols based in the measurement of the round-trip time of authenticated messages exchanged between two RFID devices, namely a tag and a reader.

A. Frauds

The *mafia fraud* has been introduced by Desmedt et al. [9], [10], then extended by Bengio et al. [3]. In this attack scenario, both the reader (R) and the tag (T) are honest, but an adversary performs a man-in-the-middle attack between them, using a fraudulent tag ($\overline{T}$) and a fraudulent reader ($\overline{R}$). The fraudulent tag $\overline{T}$ interacts with the honest reader R and the fraudulent reader $\overline{R}$ interacts with the honest tag T. The devices $\overline{T}$ and $\overline{R}$ cooperate together and communicate through a wired or wireless dedicated channel. This architecture enables $\overline{T}$ to convince R of a statement related to the secret information of the honest tag T, without actually knowing anything about this secret information.

Following the mafia fraud, two other attacks have been suggested: the *distance fraud* and the *terrorist fraud* (see [1] for a comprehensive and historical overview). In the distance fraud, the adversary is no longer a man-in-the-middle but a dishonest tag that claims to be closer than it really is. The terrorist fraud is an extension of mafia fraud where the tag T is no longer honest and collaborates with the fraudulent tag $\overline{T}$. The dishonest tag T uses $\overline{T}$ to convince the reader that it is closer than it really is, but T does not want to provide to $\overline{T}$ the ability to perform itself the fraud afterward.

Among these attacks, the mafia fraud is definitely the most serious one since it can be mounted without the awareness of the honest tag. Many works about distance bounding have been published recently [2], [4], [5], [6], [7], [11], [12], [13], [15], [16], [17], [19]. None of them has succeeded in preventing mafia, distance, and terrorist fraud attacks simultaneously. Defeating three attacks simultaneously is quite a difficult challenge and an ongoing research topic. Up to our knowledge, there is no existing scheme yet that resists to the three frauds with a significant probability. Therefore we only focus on mafia and distance fraud attacks in this article.

B. Distance bounding protocols

In 1993, Brands and Chaum presented the first distance bounding protocol [4]. The basic mechanism is as follows. The protocol includes a *fast-bit exchange* phase where the reader sends out one bit and starts a timer. Then the tag responds to the reader with one bit that stops the timer. The reader measures the round-trip time and extracts the propagation time.

C. H. Kim and G. Avoine are with the Information Security Group, Université Catholique de Louvain, Belgium (e-mail:{chonghee.kim,gildas.avoine}@uclouvain.be).

After a series of n rounds (n is a security parameter), the reader decides whether the tag is within the authorized area.

The processing time spent during the exchanges should be minimized to reduce the uncertainty of the distance-bounding process. Single-bit exchanges provide the highest time (and therefore distance) resolution, as it depends only on propagation time, pulse width, and processing delay. In contrast, some authors suggest multi-bit exchanges [7], but this affects the resolution.

The choice of the communication medium and the transmission format should be optimized as well. In [8], the authors propose some principles: 1) use a communication medium with a propagation speed as close as possible to the physical limit for propagating information through space-time, 2) use a communication format in which only a single bit is transmitted and recipient can instantly react on its reception, 3) minimize the length of the symbol used to represent this single bit.

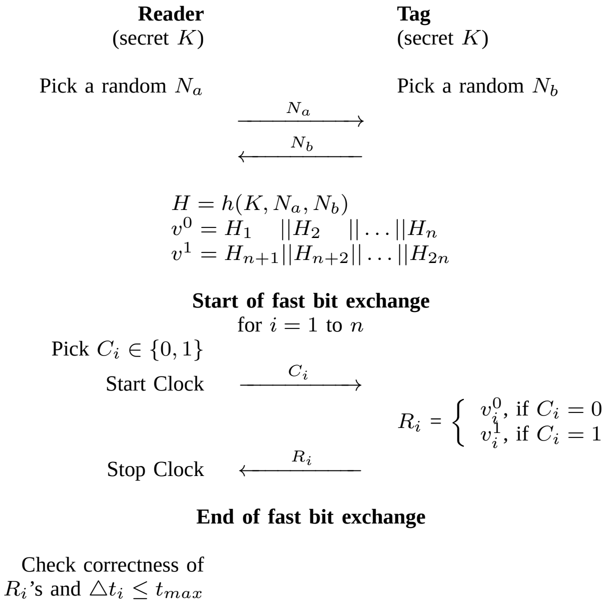

Reader
(secret K)
Tag
(secret K)
Pick a random N_a
Pick a random N_b
$\xrightarrow{N_a}$
$\xleftarrow{N_b}$
$H = h(K, N_a, N_b)$
$v^0 = H_1 \| H_2 \| \ldots \| H_n$
$v^1 = H_{n+1} \| H_{n+2} \| \ldots \| H_{2n}$
Start of fast bit exchange
for $i = 1$ to n
Pick $C_i \in \{0, 1\}$
Start Clock
$\xrightarrow{C_i}$
$R_i = \begin{cases} v_i^0, \text{ if } C_i = 0 \\ v_i^1, \text{ if } C_i = 1 \end{cases}$
Stop Clock
$\xleftarrow{R_i}$
End of fast bit exchange
Check correctness of
R_i's and $\triangle t_i \le t_{max}$

Fig. 1. Hancke and Kuhn's protocol

Hancke and Kuhn propose in [12] a distance bounding protocol (HK) that undeniably became a key-reference in this domain. As depicted in Fig. 1, the protocol is carried out as follows. After the exchange of random nonces N_a and N_b, the reader and the tag compute two n-bit sequences, v^0 and v^1, using a pseudorandom function (in practice a MAC algorithm or a hash function). Then the reader sends a random bit, n times. Upon receiving a bit, the tag sends back a bit R_i from v^0 (resp. v^1) if the received bit C_i is equal to 0 (resp. 1). After n rounds, the reader checks the correctness of the values R_i and the measured round-trip times.

An adversary who tries to impersonate a tag needs to correctly answer to the n challenges. In each round, the probability that she sends a correct response is *a priori* $\frac{1}{2}$. However she can query the tag in advance with some arbitrary C_i''s, between the nonces' exchange phase and the fast bit exchange phase. In other words, the adversary can use a *pre-ask strategy*, defined in [1]. Doing so, the adversary obtains n bits from the registers. For example, if the adversary queries the tag with n zeroes, she entirely gets v^0. With the probability of $\frac{1}{2}$, the adversary has the correct guess, that is $C_i' = C_i$, and therefore has in advance the correct values R_i's that are needed to satisfy the reader. With the probability of $\frac{1}{2}$, the adversary can only reply with a random bit, which is correct with probability $\frac{1}{2}$. Therefore, the adversary can reply correctly to the reader with probability $\frac{3}{4}$.

One of the solutions to reduce this probability consists in including a final signed message [4], [17], [19]. However this approach introduces an overhead and requires an additional message to be transmitted, slowing down the protocol execution.

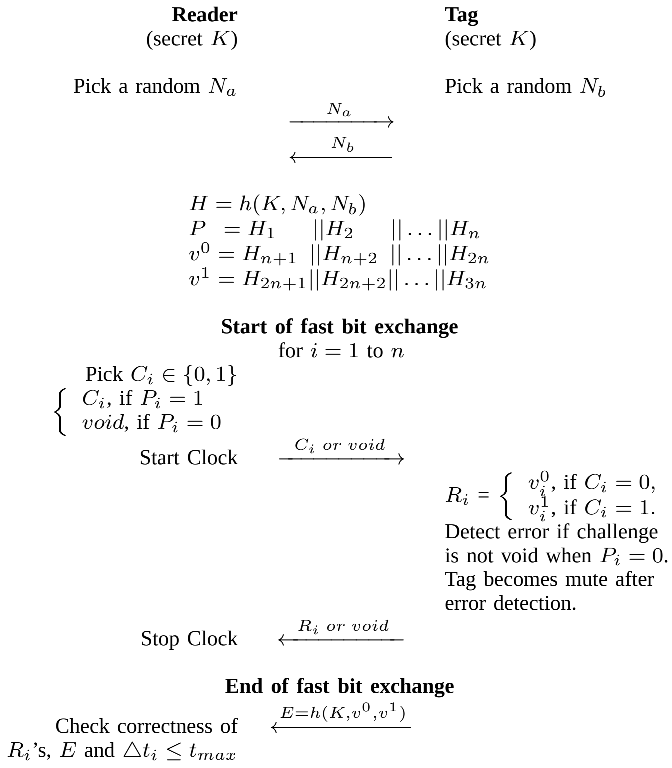

Reader
(secret K)
Tag
(secret K)
Pick a random N_a
Pick a random N_b
$\xrightarrow{N_a}$
$\xleftarrow{N_b}$
$H = h(K, N_a, N_b)$
$P = H_1 \| H_2 \| \ldots \| H_n$
$v^0 = H_{n+1} \| H_{n+2} \| \ldots \| H_{2n}$
$v^1 = H_{2n+1} \| H_{2n+2} \| \ldots \| H_{3n}$
Start of fast bit exchange
for $i = 1$ to n
Pick $C_i \in \{0, 1\}$
$\begin{cases} C_i, \text{ if } P_i = 1 \\ void, \text{ if } P_i = 0 \end{cases}$
Start Clock
$\xrightarrow{C_i \ or \ void}$
$R_i = \begin{cases} v_i^0, \text{ if } C_i = 0, \\ v_i^1, \text{ if } C_i = 1. \end{cases}$
Detect error if challenge
is not void when $P_i = 0$.
Tag becomes mute after
error detection.
Stop Clock
$\xleftarrow{R_i \ or \ void}$
End of fast bit exchange
Check correctness of
R_i's, E and $\triangle t_i \le t_{max}$
$\xleftarrow{E = h(K, v^0, v^1)}$

Fig. 2. Munilla and Peinado's protocol

Munilla and Peinado, in collaboration with Ortiz, propose in [15], [16] a modified version of Hancke and Kuhn's protocol by applying "void challenges" in order to reduce the success probability of the adversary. As shown in Fig. 2, the challenges from the reader are divided into two categories, *full challenges* and *void challenges*. After the exchange of random nonces (N_a and N_b), the reader and the tag compute a $3n$-bit sequence, $P \| v^0 \| v^1$, using a pseudorandom function. The vector P is used to decide when sending void challenges: if $P_i = 1$ the reader sends a random challenge, while if $P_i = 0$ it sends a void challenge, i.e., it does not send anything. The void challenges allow the tag to detect if an adversary is currently trying to obtain the responses in advance. If the tag detects such a behavior, it stops sending responses. Otherwise, the protocol eventually ends with a message to verify that no adversary has been detected.

The adversary can choose between two main strategies: (a) querying the tag in advance, taking the risk to be uncovered by the reader (*pre-ask strategy*), and (b) answering randomly to the reader and then querying the tag in order to get the valid final signature (*post-ask strategy*). The mafia fraud success probability depends on p_f, the probability of occurrence of a full challenge, and can be calculated as follows:

$$\Pr_{MP} = \begin{cases} (1-\frac{p_f}{2})^n, \text{if } \; p_f \le \frac{4}{5} \text{ (post-ask strategy)}, \\ (p_f \cdot \frac{3}{4})^n, \text{if } \; p_f > \frac{4}{5} \text{ (pre-ask strategy)}. \end{cases} \qquad (1)$$

The smallest mafia fraud success probability is obtained when $p_f = 4/5$, but it is not easy to generate a bit string P with such a p_f. However, the value $p_f = 3/4$ is close to $4/5$ and is much easier to generate. By generating a random $2n$-bit P and letting '00', '01', or '10' as $P_i = 1$ and '11' as $P_i = 0$, we can get $p_f = 3/4$. If the responses of the tag are taken out from one edge of the bit-string (LSB, the least significant bit) or from the other one (MSB, the most significant bit) depending on the challenge, $n+1$ bits are enough to generate $v^0||v^1$. Therefore $3n+1$ bits only ($2n$ bits for P, $n+1$ bits for the responses) needs to be stored. The success probability of the adversary is $(\frac{5}{8})^n$ if the string P is random [16], which is less than $(\frac{3}{4})^n$.

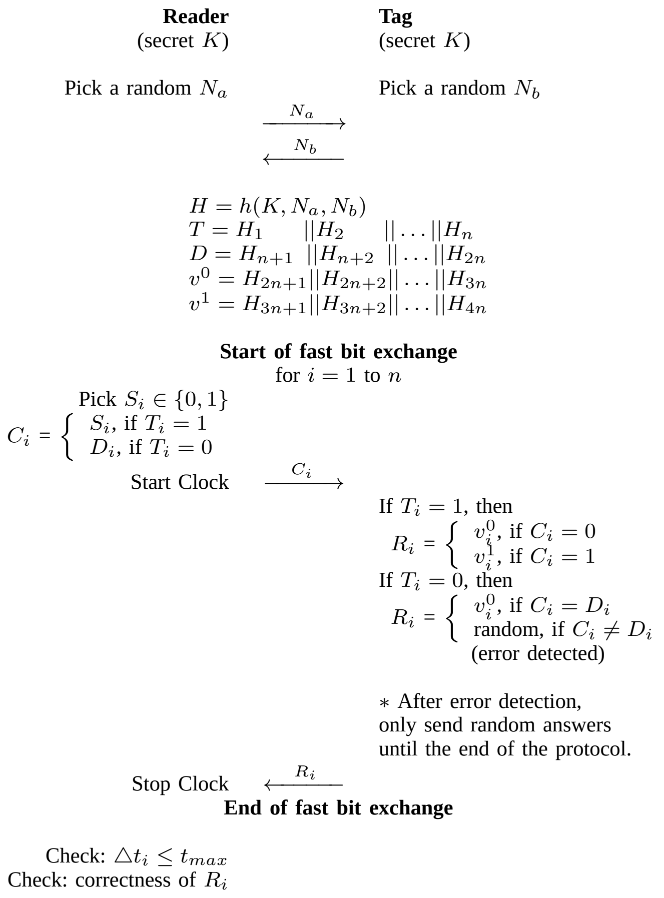
Note that the final confirmation message $h(K, v^0, v^1)$ does not take any C_i input. So it can be pre-computed before the fast bit exchange starts. Unfortunately, the disadvantages of this protocol is that (a) it requires three (physical) states: 0, 1, and *void*, which may be difficult to implement (b) the success probability of the adversary is higher than $(\frac{1}{2})^n$.

Avoine and Tchamkerten proposed in [2] a distance bounding protocol using a decision tree to set up the fast phase. Their protocol shows a good security against the mafia fraud: the adversary success probability is $(\frac{1}{2})^n(\frac{n}{2}+1)$. However its memory requirement exponentially increases as n becomes larger, that is, $2^{n+1}-2$ bits should be stored in the tag. To overcome this huge memory requirement, they propose a variant using several smaller trees that reduces the storage requirement but that increases the adversary's probability of success. With α small trees of depth k, i.e., $n = \alpha k$, the adversary's probability of success is $((\frac{1}{2})^k(\frac{k}{2}+1))^\alpha$ and the number of bits to store is $\alpha(2^{k+1}-2)$.

Trujillo-Rasua, Martin, and Avoine proposed in [18] a new distance bounding protocol, called Poulidor, based on graphs. Their goal is not to provide the best security against mafia fraud or distance fraud, but to design a protocol that ensures a reasonable trade-off between these concerns, while still using a linear memory.

C. Contributions

We provide new distance bounding protocols, KA1, KA1$^+$, and KA2, that use *mixed challenges*. Compared to MP [15], [16], our protocols do not require neither three physical states nor a confirmation message, which improves their efficiency. KA1 was originally proposed in the extended abstract [14]. KA1$^+$ is a slight modification of KA1 to decrease the adversary success probability when considering a distance fraud. KA2 reduces the adversary success probability, considering both mafia and distance frauds, and the memory consumption of the tag as well, without sacrificing any valuable property compared to KA1.



Reader
(secret K)
Tag
(secret K)
Pick a random N_a
Pick a random N_b
N_a
N_b
$H = h(K, N_a, N_b)$
$T = H_1 \; ||H_2 \; ||\ldots||H_n$
$D = H_{n+1} \; ||H_{n+2} \; ||\ldots||H_{2n}$
$v^0 = H_{2n+1}||H_{2n+2}||\ldots||H_{3n}$
$v^1 = H_{3n+1}||H_{3n+2}||\ldots||H_{4n}$
Start of fast bit exchange
for $i = 1$ to n
Pick $S_i \in \{0,1\}$
$C_i = \begin{cases} S_i, \text{ if } T_i = 1 \\ D_i, \text{ if } T_i = 0 \end{cases}$
Start Clock
C_i
If $T_i = 1$, then
$R_i = \begin{cases} v_i^0, \text{ if } C_i = 0 \\ v_i^1, \text{ if } C_i = 1 \end{cases}$
If $T_i = 0$, then
$R_i = \begin{cases} v_i^0, \text{ if } C_i = D_i \\ \text{random, if } C_i \neq D_i \\ \text{(error detected)} \end{cases}$
$*$ After error detection,
only send random answers
until the end of the protocol.
Stop Clock
R_i
End of fast bit exchange
Check: $\triangle t_i \le t_{max}$
Check: correctness of R_i

Fig. 3. Distance bounding protocol using mixed challenges: KA1 protocol

We also provide a comparison of KA1, KA2, and other existing key-reference protocols: HK (Hancke and Kuhn) [12], MP (Munilla and Peinado) [15], AT (Avoine and Tchamkerten) [2], and Poulidor (Trujillo-Rasua, Martin, and Avoine) [18]. Finally we provide an analysis in a noisy channel.

II. Distance bounding protocols using mixed challenges: KA1 and KA1$^+$

A. Description of KA1

To overcome the disadvantage of MP, we present an enhancing technique based on *mixed challenges*: the challenges from the reader to the tag in the fast-bit exchange are divided into two categories, the *random challenges* and the *predefined challenges*. The earlier are random bits chosen by the reader and the latter are predefined bits known in advance by both the reader and the tag.

As shown in Fig. 3, the reader and the tag compute a random $4n$-bit sequence $T||D||v^0||v^1$, after the exchange of random nonces (N_a and N_b). The vector T is used to decide whether a random or a predefined challenge should be sent: if $T_i = 1$ the reader sends a random challenge $S_i \in \{0,1\}$, while if $T_i = 0$ it sends a predefined challenge D_i to the tag. From the adversary viewpoint, all C_i's look like random. Therefore she cannot distinguish random challenges from predefined challenges. However, with the predefined challenges, the tag can detect an adversary who early queries it with wrong challenges. Upon reception of a challenge C_i, the tag sends back the bit $v_i^{C_i}$ if $T_i = 1$ (random challenge). When $T_i = 0$ (predefined challenge), the tag sends back the bit v_i^0 if $C_i = D_i$ or a random bit if $C_i \neq D_i$ (it detects an error). Once the tag has detected an error, it always replies a random value (See Section II-B) to all the subsequent challenges sent by the reader. By doing this, both the reader and the tag fight the adversary.

We point out that we do not use any confirmation message after the end of the fast bit exchange phase, which improves the efficiency in terms of computation and communication compared to MP [16].

B. Discussion about the tag's behavior after an error is detected

In our protocol, the tag always replies with a random bit after the detection of an error. This is a conservative behavior but some other ones are also possible.

Interrupt the protocol. One may think that the tag can simply interrupt the protocol when an error is detected. However, the reader may simply concludes in such a case that the protocol failed, while in practice it could be interesting for the reader to be able to distinguish a failure from an attack; it could so react accordingly.

Complementary bits. Another variant is for the tag to send the complementary bits of the right answers once an error is detected. In this way, the tag helps the reader to detect earlier that an attack occurs. Indeed, if the strategy of the adversary consists exactly in sending to the reader what she previously received from the tag, her probability of success is 0 once she sent a wrong challenge. However, with such a variant, a better strategy for the adversary is to expect an early wrong challenge, and then to flip all the subsequent responses from the tag. His probability of success becomes 1 once she sent a wrong challenge.

Half-time complementary bits. To thwart an attack based on the "flip strategy", one way consists in flipping only half of the responses. Thus, after an error is detected, the tag sends a right response when $T_i = 0$ but sends a complementary one when $T_i = 1$. As the adversary cannot distinguish between $T_i = 0$ and $T_i = 1$, she cannot decide when she delivers the response as it is or not. Consequently, after an error, the probability for the adversary to get the right response is 1 if $T_i = 0$, and 0 if $T_i = 1$.

Use the obsolete D_is. Instead of using the complementary approach or generating new random bits, the tag may reply with the remaining D_i's after an attack is detected. Indeed, after a wrong challenge is received, the D_i's become useless. This approach has two advantages: (a) to avoid generating new random values; (b) to help the reader to detect earlier an attack (the reader checks if the answers match the D_i's). However, as the D_i's are still used for the reader's challenges when $T_i = 0$, this variant gives to the adversary the ability to observe that she has been detected by the tag. She may so interrupt the protocol, expecting the reader to conclude that a failure occurs instead of an attack.

Use the obsolete T_is. As in the previous variant, after an error is detected, the tag uses an already generated random register that is no longer in use. So, T is used instead of D because T does not reveal that the adversary is detected. This variant presents the same two advantages than the previous one without revealing the attack detection.

A detailed analysis of the success probability of the adversary follows in the next section. We consider in this analysis the basic version of the protocol, where the tag replies with random bits after an error is detected.

C. Security analysis

We define p_d as the probability that a challenge is a *predefined challenge*. Similarly p_r is defined as the probability that a challenge is a *random challenge*. Therefore we have $p_d + p_r = 1$.

The adversary can choose one out of two attack strategies: classical impersonation or pre-ask strategy. The post-ask strategy is useless since the protocol does not have any final message [1]. The adversary's probabilities of success are respectively denoted $P_{impersonation}$ and $P_{pre-ask}$. The probability $P_{impersonation}$ is always $(1/2)^n$ and smaller than $P_{pre-ask}$. Therefore it is better for the adversary to use the pre-ask strategy. From now on we only consider $P_{pre-ask}$.

1) Mafia fraud success probability: To compute $P_{pre-ask}$, we assume that the adversary queries the tag in advance. If the challenge, C_i^*, that the adversary sends in advance to the tag is the same than the challenge C_i sent by the reader to the tag, she sends the response received from the tag to the reader. If $C_i^* \neq C_i$, then she sends a random bit to the reader.

The probability of not being detected by the reader until the i-th round, $P(i)$, depends whether the attack is detected by the tag in the previous rounds or not. We define the following events:

- $\bar{a}_i$: the event that the attack is *not* detected at the i-th round by the reader,
- b_i: the event that the attack is detected at the i-th round by the tag,
- $\bar{b}_i$: the event that the attack is *not* detected at the i-th round by the tag,
- $\bar{A}_i$: the event that the attack is *not* detected **until the i-th round** by the reader,
- B_i: the event that the attack is detected at the i-th round by the tag **for the first time**,
- $\bar{B}_i$: the event that the attack is *not* detected **until the i-th round** by the tag.

Therefore,

$$P(i) = \Pr(\bar{A}_i|\bar{B}_i)\Pr(\bar{B}_i) + \sum_{k=1}^{i} \Pr(\bar{A}_i|B_k)\Pr(B_k). \quad (2)$$

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

$$\Pr(B_i) = (1 - \frac{p_d}{2})^{i-1} \cdot \frac{p_d}{2}, \quad (3)$$

and the probability of not being detected by the tag until i-th round is:

$$\Pr(\bar{B_i}) = (1-\frac{p_d}{2})^i. \tag{4}$$

We can compute

$$\Pr(\bar{A_i}|B_k) = \prod_{j=1}^{k-1}\Pr(\bar{a_j}|\bar{b_j}) \cdot \prod_{j=k}^{i}\Pr(\bar{a_j}|b_k), \tag{5}$$

where $\Pr(\bar{a_j}|b_k)=\frac{1}{2}$, $k \le j$ and

$$\Pr(\bar{a_j}|\bar{b_j}) = \frac{\Pr(\bar{a_j}\cap\bar{b_j})}{\Pr(\bar{b_j})}.$$

The probability $\Pr(\bar{a_j}\cap\bar{b_j}) = \Pr(\bar{a_j}\cap\bar{b_j}|p_d)p_d + \Pr(\bar{a_j}\cap\bar{b_j}|p_r)p_r$. And $\Pr(\bar{a_j}\cap\bar{b_j}|p_d) = \frac{1}{2}$ as the adversary should send the correct challenge. $\Pr(\bar{a_j}\cap\bar{b_j}|p_r) = \frac{3}{4}$ as this is the same as in Hancke and Kuhn's protocol. This yields:

$$\Pr(\bar{a_j}|\bar{b_j}) = \frac{\frac{1}{2}p_d + \frac{3}{4}p_r}{1-\frac{p_d}{2}} = \frac{2p_d+3p_r}{4-2p_d}. \tag{6}$$

From Equations (5) and (6), we have

$$\begin{aligned}
\Pr(\bar{A_i}|B_k) &= \prod_{j=1}^{k-1}\Pr(\bar{a_j}|\bar{b_j}) \cdot \prod_{j=k}^{i}\Pr(\bar{a_j}|b_k) \\
&= \prod_{j=1}^{k-1}\frac{2p_d+3p_r}{4-2p_d} \cdot \prod_{j=k}^{i}\frac{1}{2} \\
&= (\frac{2p_d+3p_r}{4-2p_d})^{k-1} \cdot (\frac{1}{2})^{i-k+1}. \qquad (7)
\end{aligned}$$

Similarly

$$\Pr(\bar{A_i}|\bar{B_i}) = \prod_{j=1}^{i}\Pr(\bar{a_j}|\bar{b_j}) = (\frac{2p_d+3p_r}{4-2p_d})^i. \tag{8}$$

From Equations (2), (3), (4), (7), and (8), we can finally compute the probability of not being detected by the reader until the i-th round as follows:

$$\begin{aligned}
P(i) &= \Pr(\bar{A_i}|\bar{B_i})\Pr(\bar{B_i}) + \sum_{k=1}^{i}\Pr(\bar{A_i}|B_k)\Pr(B_k) \\
&= (\frac{2p_d+3p_r}{4-2p_d})^i(1-\frac{p_d}{2})^i + \\
&\quad \sum_{k=1}^{i}\left\{(\frac{2p_d+3p_r}{4-2p_d})^{k-1}(\frac{1}{2})^{i-k+1}\}\{(1-\frac{p_d}{2})^{k-1}\frac{p_d}{2}\right\} \\
&= (\frac{3-p_d}{4})^i + \frac{p_d}{2}\sum_{k=1}^{i}(\frac{3-p_d}{4})^{k-1}(\frac{1}{2})^{i-k+1}.
\end{aligned}$$

When $p_d = 0$ (always random challenges),

$$P(i) = (\frac{3}{4})^i,$$

and when $p_d = 1$ (always predefined challenges),

$$\begin{aligned}
P(i) &= (\frac{1}{2})^i + \frac{1}{2}\sum_{k=1}^{i}(\frac{1}{2})^{k-1}(\frac{1}{2})^{i-k+1} \\
&= (\frac{1}{2})^i(\frac{i}{2}+1). \qquad (9)
\end{aligned}$$

2) Distance fraud success probability: Until now we assumed that the tag was honest and the adversary tried to perform a mafia fraud. In this section, we consider the case of a dishonest tag. The latter knows the predefined challenges before the fast-bit exchange phase starts and may use this knowledge to deceive the reader.

In the extended abstract [14], we assume that each bit of T is generated according to the distribution of p_d (or p_r). The probability of success of the distance fraud by the dishonest tag for one round is:

$$\begin{aligned}
P_{\text{distance fraud}} &= P_{\text{random ch. \& deceive}} + P_{\text{predefined ch. \& deceive}} \\
&= p_r(P_{v_i^0=v_i^1 \text{ \& deceive}} + P_{v_i^0\neq v_i^1 \text{ \& deceive}}) + p_d \\
&= p_r \cdot (\frac{1}{2}\cdot 1 + \frac{1}{2}\cdot\frac{1}{2}) + p_d \\
&= \frac{3}{4}p_r + p_d \\
&= 1 - \frac{1}{4}p_r.
\end{aligned}$$

If a challenge is random ($T_i = 1$) and the i-th bits of v^0 and v^1 are equal, then the tag can send its response early. If the i-th bits of v^0 and v^1 are not equal and $T_i = 1$, the tag chooses the response randomly. If a challenge is predefined ($T_i = 0$), she can send its response early. Hence, the overall probability for n rounds is $(1-\frac{1}{4}p_r)^n$.

D. Description of KA1+

In this article, we propose a new method, called KA1$^+$, to generate T, such that the tag generates $n - \lceil n \cdot p_d \rceil$ 1's and $\lceil n \cdot p_d \rceil$ 0's and randomly mixes them according to the output of $h(K, N_a, N_b)$. There are so exactly $n - \lceil n \cdot p_d \rceil$ random challenges. The success probability of the distance fraud becomes $(\frac{3}{4})^{n-\lceil n \cdot p_d \rceil}$, which is better than the one achieve with KA1.

We note that the success probability of the distance fraud decreases as p_r gets closer to 1. However that of the mafia fraud increases as p_r becomes higher. Therefore the trade-off between these two attacks should be considered according to the applications.

III. Improved scheme: KA2

A. Description of KA2

In KA1, the tag can ckeck whether an attack occurs only when a challenge is predefined. The adversary's probability of success in one round then becomes $\frac{1}{2}$ after an error is detected by the tag. In KA2, the predefined challenges are placed in the first α rounds of the fast bit exchange phase, which allows to detect the attack earlier, hence decreasing the probability of success of the adversary. The random challenges are sent in the remaining $\beta = n - \alpha$ rounds, hence making $p_d = \frac{\alpha}{n}$. By doing so, KA2, which is depicted in Fig. 4, provides a better security against both the mafia fraud and the distance fraud than KA1. Furthermore KA2 requires less memory.

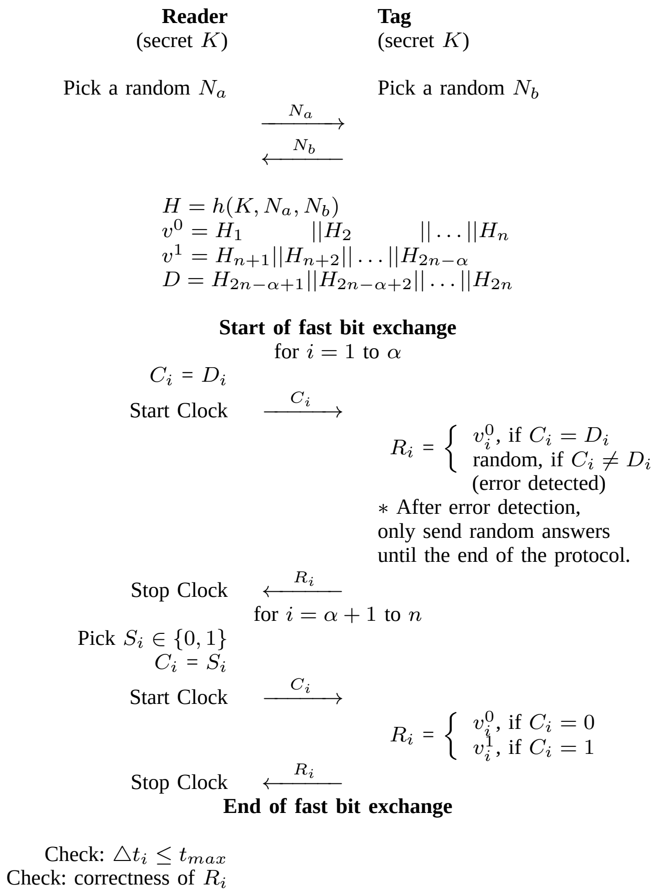

Reader
(secret K)
Tag
(secret K)
Pick a random N_a
Pick a random N_b
N_a
N_b
$H = h(K, N_a, N_b)$
$v^0 = H_1 \quad ||H_2 \quad ||\ldots||H_n$
$v^1 = H_{n+1}||H_{n+2}||\ldots||H_{2n-\alpha}$
$D = H_{2n-\alpha+1}||H_{2n-\alpha+2}||\ldots||H_{2n}$
Start of fast bit exchange
for $i = 1$ to α
$C_i = D_i$
Start Clock
C_i
$R_i = \begin{cases} v_i^0, \text{ if } C_i = D_i \\ \text{random, if } C_i \neq D_i \\ \text{(error detected)} \end{cases}$
$*$ After error detection,
only send random answers
until the end of the protocol.
Stop Clock
R_i
for $i = \alpha + 1$ to n
Pick $S_i \in \{0, 1\}$
$C_i = S_i$
Start Clock
C_i
$R_i = \begin{cases} v_i^0, \text{ if } C_i = 0 \\ v_i^1, \text{ if } C_i = 1 \end{cases}$
Stop Clock
R_i
End of fast bit exchange
Check: $\triangle t_i \leq t_{max}$
Check: correctness of R_i

Fig. 4. Distance bounding protocol using mixed challenges: KA2 protocol

B. Mafia fraud success probability

The probability of not being detected by the reader until the i-th round, $P(i)$, is

$$P(i) = \Pr(\bar{A}_i|\bar{B}_i)\Pr(\bar{B}_i) + \sum_{k=1}^{i}\Pr(\bar{A}_i|B_k)\Pr(B_k).$$

To compute $P(i)$, we have to consider two cases: $i \leq \alpha$ and $i > \alpha$. When $i \leq \alpha$, only predefined challenges are sent. When $i > \alpha$, random challenges are sent after predefined challenges.

1) Probability when $i \leq \alpha$: When $i \leq \alpha$, there are only predefined challenges. Therefore $p_d = 1$ and $P(i)$ is computed from Equation (9):

$$P(i) = (\frac{1}{2})^i(\frac{i}{2} + 1), \quad i \leq \alpha. \tag{10}$$

2) Probability when $i > \alpha$:

$$P(i) = \Pr(\bar{A}_i|\bar{B}_i)\Pr(\bar{B}_i) + \sum_{k=1}^{i}\Pr(\bar{A}_i|B_k)\Pr(B_k).$$

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

$$\Pr(B_i) = \begin{cases} (\frac{1}{2})^{i-1} \cdot \frac{1}{2} = (\frac{1}{2})^i, & i \leq \alpha, \\ 0, & i > \alpha. \end{cases} \tag{11}$$

and the probability of not being detected by the tag until the i-th round is:

$$\Pr(\bar{B}_i) = \begin{cases} (\frac{1}{2})^i, & i \leq \alpha, \\ (\frac{1}{2})^\alpha, & i > \alpha. \end{cases} \tag{12}$$

From Equations (11) and (12), we have:

$$\begin{aligned} P(i) &= \Pr(\bar{A}_i|\bar{B}_i)\Pr(\bar{B}_i) + \sum_{k=1}^{i}\Pr(\bar{A}_i|B_k)\Pr(B_k), \\ &= \Pr(\bar{A}_i|\bar{B}_i)(\frac{1}{2})^\alpha + \sum_{k=1}^{\alpha}\Pr(\bar{A}_i|B_k)(\frac{1}{2})^k, \end{aligned} \tag{13}$$

where $i > \alpha$. We can compute $\Pr(\bar{A}_i|B_k)$ for $i > \alpha$ and $1 \leq k \leq \alpha$:

$$\Pr(\bar{A}_i|B_k) = \prod_{j=1}^{k-1}\Pr(\bar{a_j}|\bar{b_j}) \cdot \prod_{j=k}^{i}\Pr(\bar{a_j}|b_k), \tag{14}$$

where $\Pr(\bar{a_j}|b_k) = \frac{1}{2}$, $k \leq j$ and

$$\Pr(\bar{a_j}|\bar{b_j}) = \frac{\Pr(\bar{a_j} \cap \bar{b_j})}{\Pr(\bar{b_j})}.$$

The probability $\Pr(\bar{a_j} \cap \bar{b_j}) = \frac{1}{2}$, as the adversary should send the correct challenge. Therefore:

$$\Pr(\bar{a_j}|\bar{b_j}) = \frac{\frac{1}{2}}{\frac{1}{2}} = 1.$$

From Equation (14), we have

$$\begin{aligned} \Pr(\bar{A}_i|B_k) &= \prod_{j=1}^{k-1}\Pr(\bar{a_j}|\bar{b_j}) \cdot \prod_{j=k}^{i}\Pr(\bar{a_j}|b_k) \\ &= \prod_{j=1}^{k-1} 1 \cdot \prod_{j=k}^{i}\frac{1}{2}, \\ &= (\frac{1}{2})^{i-k+1}, \end{aligned} \tag{15}$$

and

$$\begin{aligned} \Pr(\bar{A}_i|\bar{B}_i) &= \prod_{j=1}^{\alpha}\Pr(\bar{a_j}|\bar{b_j}) \cdot \prod_{j=\alpha+1}^{i}\Pr(\bar{a_j}|\bar{b_j}) \\ &= \prod_{j=1}^{\alpha} 1 \cdot \prod_{j=\alpha+1}^{i}(\frac{3}{4}) \\ &= (\frac{3}{4})^{i-\alpha}. \end{aligned} \tag{16}$$

From Equations, (13), (15) and (16), we can finally compute the probability of not being detected by the reader until the i-th round, $P(i)$, for $i > \alpha$ as follows:

$$\begin{aligned} P(i) &= \Pr(\bar{A}_i|\bar{B}_i)\Pr(\bar{B}_i) + \sum_{k=1}^{i}\Pr(\bar{A}_i|B_k)\Pr(B_k), \\ &= \Pr(\bar{A}_i|\bar{B}_i) \cdot (\frac{1}{2})^\alpha + \sum_{k=1}^{\alpha}\Pr(\bar{A}_i|B_k) \cdot (\frac{1}{2})^k, \\ &= (\frac{3}{4})^{i-\alpha} \cdot (\frac{1}{2})^\alpha + \sum_{k=1}^{\alpha}(\frac{1}{2})^{i-k+1} \cdot (\frac{1}{2})^k, \\ &= (\frac{3}{4})^{i-\alpha} \cdot (\frac{1}{2})^\alpha + \alpha(\frac{1}{2})^{i+1}, \quad i > \alpha. \end{aligned} \tag{17}$$

From Equations (10) and (17), we finally have:

$$P(i) = \begin{cases} (\frac{1}{2})^i(\frac{i}{2}+1), & i \le \alpha, \\ (\frac{3}{4})^{i-\alpha}\cdot(\frac{1}{2})^\alpha + \alpha(\frac{1}{2})^{i+1}, & i > \alpha. \end{cases} \qquad (18)$$

For n rounds, we have $P(n) = (\frac{3}{4})^{n-\alpha}\cdot(\frac{1}{2})^\alpha + \alpha(\frac{1}{2})^{n+1}$.

C. Distance fraud success probability

During the first α iterations, the dishonest tag can correctly answer to the challenges. Therefore the success probability for the first α rounds is 1. For the remaining $n-\alpha$ rounds, if the i-th bits of v^0 and v^1 are the same, then the tag can definitely send a correct response without waiting for the query. If the i-th bits of v^0 and v^1 are not equal then the tag has to choose the response randomly. So the probability of success is $\frac{3}{4}$. The overall probability of success is $(\frac{3}{4})^{n-\alpha}$.

IV. Analysis with noise

In practice, channel noise introduces some errors in the communication between the tag and the reader. Therefore, the reader must accept a tag as valid, even if, out of n received responses, at most m are incorrect. However, the adversary can get benefits from this threshold. Consequently, we analyze the success probability of the adversary in the noisy case. We define the following new event and notions:

- C_i: the event that at most m errors are detected until the i-th round by the reader,
- m_1: the number of errors detected by the reader before the tag detects an error,
- m_2: the number of errors detected by the reader after the tag detects an error, $m = m_1 + m_2$.

A. Analysis of KA1 with a noisy channel

The success probability of the adversary for KA1 protocol is:

$$P(n) = \Pr(C_n|\bar{B_n})\Pr(\bar{B_n}) + \sum_{k=1}^{n}\Pr(C_n|B_k)\Pr(B_k), \quad (19)$$

where

$$\begin{aligned} \Pr(C_n|\bar{B_n}) &= \sum_{i=0}^{m}\binom{n}{i}(1-\Pr(\bar{a}_i|\bar{b}_i))^i(\Pr(\bar{a}_i|\bar{b}_i))^{n-i}, \\ &= \sum_{i=0}^{m}\binom{n}{i}(1-\frac{2p_d+3p_r}{4-2p_d})^i(\frac{2p_d+3p_r}{4-2p_d})^{n-i}, \end{aligned} \qquad (20)$$

and

$$\begin{aligned} &\Pr(C_n|B_k) \\ = &\sum_{i=0}^{m_1}\binom{k-1}{i}(1-\Pr(\bar{a}_i|\bar{b}_i))^i(\Pr(\bar{a}_i|\bar{b}_i))^{k-1-i}, \\ &\cdot\sum_{i=0}^{m_2}\binom{n-k+1}{i}(1-\Pr(\bar{a}_i|b_k))^i\Pr(\bar{a}_i|b_k)^{n-k+1-i}, \\ = &\sum_{i=0}^{m_1}\binom{k-1}{i}(1-\frac{2p_d+3p_r}{4-2p_d})^i(\frac{2p_d+3p_r}{4-2p_d})^{n-i}, \\ &\cdot\sum_{i=0}^{m_2}\binom{n-k+1}{i}(\frac{1}{2})^{n-k+1}. \end{aligned} \qquad (21)$$

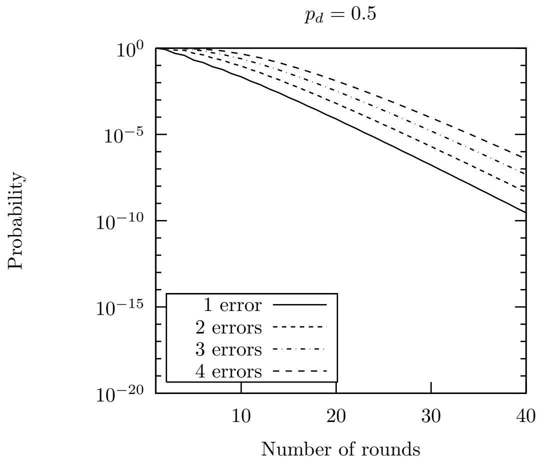


Fig. 5. Mafia fraud success probabilities of KA2 protocol in the noisy case.

From Equations (3) and (4), we have:

$$\begin{aligned} \Pr(B_k) &= (1-\frac{p_d}{2})^{i-k}\cdot\frac{p_d}{2}, & (22) \\ \Pr(\bar{B_n}) &= (1-\frac{p_d}{2})^n. & (23) \end{aligned}$$

Therefore, from Equations (19), (20), (21), (22), and (23) we can compute the success probability of the adversary in a noisy channel.

B. Analysis of KA2 with a noisy channel

When the channel is noisy, the success probability of the adversary with KA2 is:

$$P(n) = \Pr(C_n|\bar{B_n})\Pr(\bar{B_n}) + \sum_{k=1}^{\alpha}\Pr(C_n|B_k)\Pr(B_k), \quad (24)$$

where

$$\Pr(C_n|\bar{B_n}) = \sum_{i=0}^{m}\binom{n}{i}(1-(\frac{3}{4})^{n-\alpha}))^i((\frac{3}{4})^{n-\alpha})^{n-i}, \quad (25)$$

and

$$\Pr(C_n|B_k) = \sum_{i=0}^{m_2}\binom{n-k+1}{i}(\frac{1}{2})^{n-k+1}. \quad (26)$$

As $k \le \alpha$ and $\Pr(\bar{a}_i|\bar{b}_i) = 1$ for $i \le k$, we have always $m_1 = 0$.

From Equation (11) and (12), we have

$$\begin{aligned} \Pr(B_k) &= (\frac{1}{2})^k, & (27) \\ \Pr(\bar{B_n}) &= (\frac{1}{2})^\alpha. & (28) \end{aligned}$$

The adversary's success probabilities of KA2 protocol in the noisy case are depicted in Fig. 5.

V. Comparisons

We compare our protocols with HK (Hancke and Kuhn) [12], MP (Munilla and Peinado) [15], AT (Avoine and Tchamkerten) [2], and Poulidor (Trujillo-Rasua, Martin, and Avoine) [18] in terms of mafia fraud success probability

TABLE I
STORAGE REQUIREMENTS

Protocol	Memory requirement in Tag
HK	$2n$
MP	$3n+1$
AT	$2^{n+1}-2$
AT3	$\frac{14n}{3}$
Poulidor	$4n$
KA1	$4n$
KA2	$2n$

and storage requirements[1]. We also compare them with the multiple-tree variant of AT, named "AT3," that has a linear memory and a smaller mafia fraud success probability. We choose the number of trees, $\alpha = \frac{n}{3}$. Then the required memory is $\frac{14n}{3} \simeq 5n$.

We depict the mafia fraud success probabilities in Fig. 6 and 7, where $p_f = 0.75$ for MP as recommended by Munilla and Peinado. In Fig. 6, we depict the probabilities when $p_d = 0.5$. AT demonstrates the best security level and HK the worst. KA2, KA1, and Poulidor are the second ones. We point out that AT3 is not as good as expected, staying between MP and HK.

In Fig. 7, we compare the mafia fraud success probabilities by varying p_d. The probabilities of HK, AT3, Poulidor, and AT do not change. Those of KA2, KA1, and MP depend on the value of p_d (p_f in MP). The probabilities of KA2 and KA1 are bounded by HK and AT. When $p_d = 0$, they are the same than HK. When $p_d = 1$, they are the same than AT. We emphasize that KA2 shows a better security than KA1 in any case.

One of the advantages of our protocols is that we can easily change the mafia fraud success probability by varying p_d. As we raised in the security analysis, the distance fraud success probability also depends on p_d. Therefore according to the applications, one can choose an appropriate p_d.

The storage requirement is listed in Table I. All protocols except AT need a linear memory. In order to supply a comparison, we chose AT with a linear memory, i.e., AT3, which needs $5n$ bits of memory. KA2 and HK need the smallest memory, namely $2n$ bits. KA1 and Poulidor need $4n$ bits.

Therefore we can conclude that KA2 shows the best security with the smallest memory. Furthermore it has a flexibility of changing the mafia fraud and the distance fraud success probabilities easily, an interesting feature that most protocols do not have.

VI. CONCLUSION

In this article, we provided new distance bounding protocols with mixed challenges. KA1 uses predefined and random challenges in an arbitrary way. In KA2, all the predefined challenges are sent before the random challenges, decreasing so the success probability of the adversary. KA2 needs half of the memory required by KA1 and decreases the success

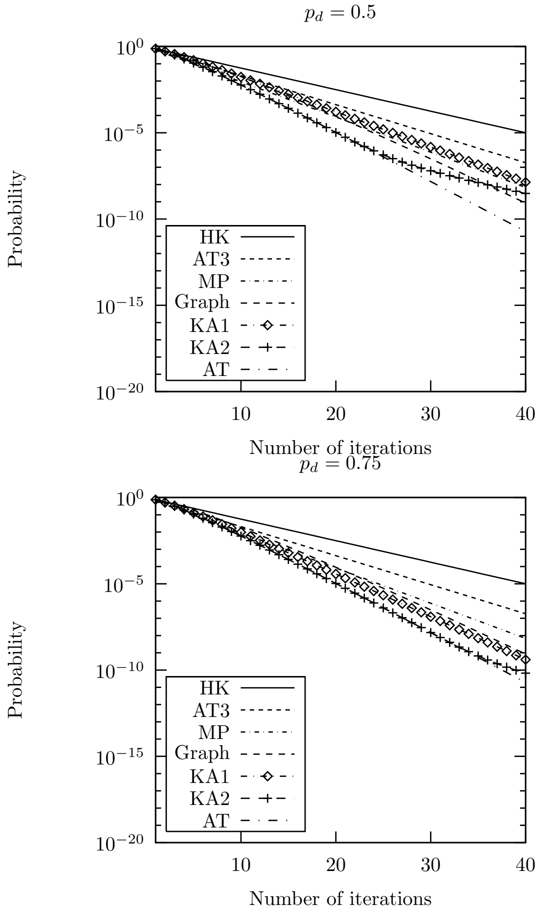


Fig. 6. Mafia fraud success probabilities of distance bounding protocols. In MP protocol, $p_f = 0.75$.

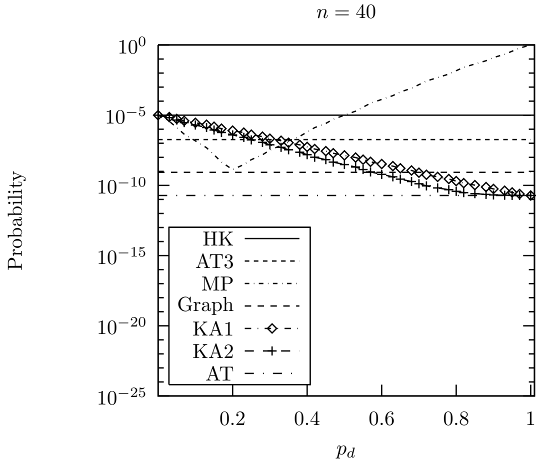


Fig. 7. Mafia fraud success probabilities of distance bounding protocols. The x-axis shows $1 - p_f$ in MP protocol.

[1] Most protocols do not consider the distance fraud and do not provide the distance fraud success probabilities. Therefore we only compare the mafia fraud success probabilities. For the same reason, we only consider a noise-free case.

probability of the mafia and distance frauds. KA2 is thus the distance bounding protocol that provides the current best trade-off between mafia fraud resistance and memory.

Although KA2 brushes the theoretical limits, future works are still needed in the domain of distance bounding. First of all, our security analysis mostly focuses on mafia fraud and keeps aside the distance fraud. Although the success probability of the distance fraud can be computed with KA2, this value is not known for AT and Poulidor, which makes the comparison difficult with respect to this fraud.

Finally, Hancke and Kuhn's protocol (HK) has been the key-reference in the domain of distance bounding and is always considered in the comparison of protocols. This is due to its simple design, but also because it is the protocol that requires the smallest memory among the existing distance bounding protocols of its category. KA2 is actually the only protocol that achieves some better security than HK, while keeping the same memory and the same number of rounds than HK.

ACKNOWLEDGMENT

The authors would like to thank the reviewers of the IEEE Transactions on Wireless Communications for their comments and suggestions. We would also like to thank Xavier Carpent for helpful comments. This work has been partially funded by the Walloon Region Marshall plan through the SPW DG06 Project TRASILUX.

REFERENCES

[1] G. Avoine, M. Bingol, S. Kardas, C. Lauradoux, and B. Martin. Framework for analyzing RFID distance bounding protocols. *Journal of Computer Security - Special Issue on RFID System Security*, 2010.

[2] G. Avoine and A. Tchamkerten. An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement. In *12th Information Security Conference, ISC 2009*, volume 5735 of *Lecture Notes in Computer Science*, pages 250–261. Springer, 2009.

[3] S. Bengio, G. Brassard, Y. Desmedt, C. Goutier, and J.-J. Quisquater. Secure implementation ofo identification systems. *Journal of Cryptology*, 4(3):175–183, 1991.

[4] S. Brands and D. Chaum. Distance-Bounding Protocols. In *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359. Springer, 1994.

[5] L. Bussard and W. Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In *IFIP International Federation for Information Processing*, volume 181, pages 223–238. Springer-Verlag, 2005.

[6] L. Bussard and Y. Roudier. Embedding distance-bounding protocols within intuitive interactions. In *Security in Pervasive Computing - SPC*, volume 2802 of *Lecture Notes in Computer Science*, pages 119–142. Springer-Verlag, 2003.

[7] S. Capkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *INFOCOM*, pages 1917–1928. IEEE, 2005.

[8] J. Clulow, G. P. Hancke, M. G. Kuhn, and T. Moore. So near and yet so far: Distance-bounding attacks in wireless networks. In *Security and Privacy in Ad-Hoc and Sensor Networks, Third European Workshop, ESAS 2006*, volume 4357 of *Lecture Notes in Computer Science*, pages 83–97. Springer, 2006.

[9] Y. Desmedt. Major security problems with the “Unforgeable” (Feige)-Fiat-Shamir proofs of identiy and how to overcome them. In *SecuriCom '88*, pages 15–17, 1988.

[10] Y. Desmedt, C. Goutier, and S. Bengio. Special uses and abuses of the Fiat-Shamir passport protocol. In *Advances in Cryptology - CRYPTO '87, 7th Annual International Cryptology Conference*, volume 293 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 1988.

[11] S. Drimer and S. J. Murdoch. Keep your enemies close: distance bounding against smarcard relay attacks. In *16th USENIX Security Sympositum*, pages 1–16. USENIX Association, 2007.

[12] G. Hancke and M. Kuhn. An RFID distance bounding protocol. In *the 1st International Conference on Security and Privacy for Emergin Areas in Communications Networks (SECURECOMM'05)*, pages 67–73. IEEE Computer Society, 2005.

[13] G. Kapoor, W. Zhou, and S. Piramuthu. Distance bounding protocol for multiple RFID tag authentication. In *IEEE/IFIP International conference on Embedded and Ubiquitous Computing - EUC'08*, pages 115–120. IEEE Computer Society, 2008.

[14] C. H. Kim and G. Avoine. RFID distance bounding protocol with mixed challenges to prevent relay attacks. In *The 8th International Conference on Cryptology And Network Security, CANS 2009*, volume 5888 of *Lecture Notes in Computer Science*, pages 119–133. Springer, 2009.

[15] J. Munilla, A. Ortiz, and A. Peinado. Distance bounding protocols with void-challenges for RFID. In *Workshop on RFID Security - RFIDSec '06*, 2006.

[16] J. Munilla and A. Peinado. Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels. *Wireless communications and mobile computing*, 2008. Published online: Jan 17 2008, an extended abstract appears in [15].

[17] D. Singelée and B. 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. Springer, 2007.

[18] R. Trujillo-Rasua, B. Martin, and G. Avoine. The Poulidor distance-bounding protocol. In *Workshop on RFID Security - RFIDSec '10*, 2010.

[19] Y.-J. Tu and S. Piramuthu. RFID distance bounding protocols. In *the 1st International EURASIP Workshop in RFID Technology*, 2007. Vienna, Austria.

Chong Hee Kim received his B.S. from Kyungpook National University, Republic of Korea in 1997, M.S. and Ph.D. from POSTECH (Pohang University of Science and Technology), Republic of Korea in 1999 and 2004, respectively. He worked for Samsung Electronics Co., LTD and NXP Semiconductors N.V. He is currently senior researcher at Université Catholique de Louvain, Belgium. His recent research interests include RFID secure protocols and security of the embedded systems such as side channel analysis and fault attacks.

Gildas Avoine is professor of information security and cryptography at the UCL in Louvain-la-Neuve (Belgium), where he leads the Information Security Group (GSI) in the Department of Computing Science and Engineering. Before joining the UCL, he was researcher at the MIT (USA) hosted by Ron Rivest in the CSAIL, and at the EPFL (Switzerland) in the LASEC headed by Serge Vaudenay, where he obtained a PhD degree in cryptography. Previously, he studied at the University of Caen (France) where he received a Bachelor degree in mathematics and Bachelor and Master degrees in computer science.