RFID distance bounding protocol with mixed challenges to prevent relay attacks

Chong Hee Kim and Gildas Avoine

Université Catholique de Louvain
Louvain-la-Neuve, B-1348, Belgium

Abstract. RFID systems suffer from different location-based attacks such as distance fraud, mafia fraud and terrorist fraud attacks. Among them mafia fraud attack is the most serious since this attack can be mounted without the notice of both the reader and the tag. An adversary performs a kind of man-in-the-middle attack between the reader and the tag. It is very difficult to prevent this attack since the adversary does not change any data between the reader and the tag. Recently distance bounding protocols measuring the round-trip time between the reader and the tag have been researched to prevent this attack.
All the existing distance bounding protocols based on binary challenges, without final signature, provide an adversary success probability equal to $(3/4)^n$ where n is the number of rounds in the protocol. In this paper, we introduce a new protocol based on binary mixed challenges that converges toward the expected and optimal $(1/2)^n$ bound. We prove its security in case of both noisy and non-noisy channels.

Keywords - RFID, authentication, distance bounding protocol, relay attack.

1 Introduction

RFID (radio frequency identification) tags or contactless smart cards are often used for proximity authentication. For example, Texas Instrument (TI) manufactured an RFID device called a Digital Signature Transponder (DST). The DST serves as a theft-deterrent in millions of automobiles. Present as a tiny, concealed chip in the ignition key of the driver, the DST authenticates the key to a reader near the key slot as a precondition for starting the engine. The DST is also present in SpeedPass^{TM} wireless payment devices, used by millions of customers primarily at ExxonMobil petrol stations in North America.

RFID tags and contactless smart cards are normally passive; they operate without any internal battery and receive the power from the reader. This offers long lifetime but results in short read ranges and limited processing power. They are also vulnerable to different attacks related to the location: distance fraud and relay attacks. Relay attacks occur when a valid reader is tricked by an adversary into believing that it is communication with a valid tag and vice versa. That is, the adversary performs a kind of man-in-the-middle attack between the reader and the tag. It is difficult to prevent these attacks since the adversary does not change any data between the reader and the tag. Therefore relay attacks cannot be prevented by cryptographic protocols that operates at the application layer.

Although one could verify location through the use of GPS coordinates, RFID tags do not lend themselves to such applications. *Distance bounding protocols* is a good solution to prevent such distance fraud and relay attacks. These protocols measure the signal strength or the round-trip time between the reader and the tag. However the proof based on measuring signal strength is not secure as an adversary can easily amplify signal strength as desired or use stronger signals to read from afar.

1.1 Distance fraud and relay attacks

There are three types of attacks related with distance between the reader and the tag. The dishonest tag may claim to be closer than he really is. This attack is called **distance fraud attack**. There are two types of relay attacks: mafia fraud and terrorist fraud attacks.

Mafia fraud attack was first described by Desmedt [2]. In this attack scenario, both the reader (R) and the tag (T) are honest, but a malicious adversary is performing a man-in-the-middle attack between the reader and the tag by putting fraudulent tag ($\overline{T}$) and receiver ($\overline{R}$). The fraudulent tag $\overline{T}$ interacts with the honest reader R and the fraudulent reader $\overline{R}$ interacts with the honest tag T. $\overline{T}$ and $\overline{R}$ cooperate together. It enables $\overline{T}$ to convince R of a statement related to the secret information of an honest tag T, without actually needing to know anything about the secret information.

Terrorist fraud attack is an extension of the mafia fraud attack. The tag T is not honest and collaborate with fraudulent tag $\overline{T}$ in this attack. The dishonest tag T uses $\overline{T}$ to convince the reader that he is close, while in fact he is not. $\overline{T}$ does not know the long-term private or secret key of T.

Among these attacks, mafia fraud attack is the most serious since this attack can be mounted without the notice of neither the reader nor the tag. Many works are devoted to prevent this attack [1, 3–7].

1.2 Distance bounding protocols

In 1993, Brands and Chaum presented their distance bounding protocol [1]. It consists of a *fast bit exchanges* 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 uses the round trip time to extract the propagation time. After series of n rounds (n is a security parameter), the reader decides whether the tag is within the limitation of the distance. In order to extract the propagation time, the processing time of the tag must be as short and invariant as possible. The communication method used for these exchanges is different from the used one for the ordinary communication. An ultra wide band (UWB) channel is used to achieve a resolution of 10 cm. It does not contain any error detection or correction mechanism in order not to make additional variable cycles of processing.

Hancke and Kuhn proposed a distance bounding protocol (HKP) [3] that has been chosen as a reference-point because it is the most popular distance bounding protocol in the RFID framework. As depicted in Fig. 1, the protocol is carried out as follows. After exchanges 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 (typically a MAC algorithm, a hash function, etc.). Then the reader sends a random bit for n times. Upon receiving a bit, the tag sends back a bit R_i from v^0 if the received bit C_i equals 0. If C_i equals 1, then it sends back a bit from v^1. After n iterations, the reader checks the correctness of R_i's and the propagation time.

In each round, the probability that the adversary sends a correct response is *a priori* $\frac{1}{2}$. However the adversary can query the tag in advance with some arbitrary C_i's, between the nonces are sent and the rapid bit exchange starts. Doing so, the adversary obtains n bits of the registers. For example, if the adversary queries the with some zeroes only, he will entirely get v^0. In half of all cases, the adversary will have the correct guesses, that is $C_i' = C_i$, and therefore will have obtained in advance the correct value R_i that is needed to satisfy the reader. In the other half of all cases, the adversary can reply with a guessed

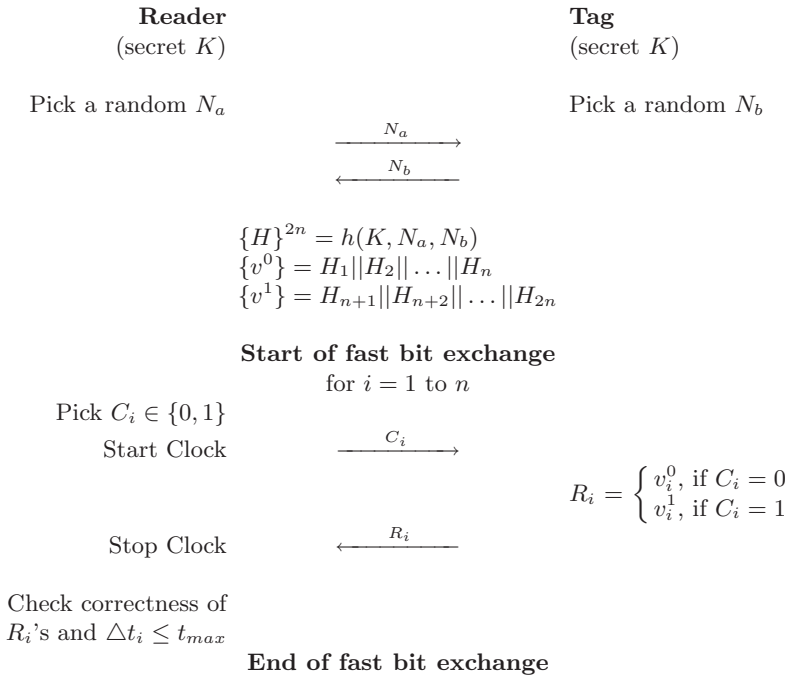

Fig. 1. Hancke and Kuhn's protocol

bit, which will be correct in half of all cases. Therefore, the adversary has $\frac{3}{4}$ probability of replying correctly.

One of the solutions to reduce the probability less than $(\frac{3}{4})^n$ is to include signed messages [1, 6, 7]. However signed messages could not be sent with UWB as it is very sensitive to the background noise. It should be sent by normal communication method with error detection or correction technique. Therefore this approach would put an overload on computation of a tag as well as communication, which causes the protocol slower.

Munilla, Ortiz, and Peinado [4, 5] modified the 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 challenge* and *void challenge*. After exchanges of random nonces (N_a and N_b), the reader and the tag compute $3n$-bit sequence, $P||v^0||v^1$, using a pseudorandom function. The string P indicates the void-challenges; that is, if $P_i = 1$ reader sends a random challenge and if $P_i = 0$ it does not. These void-challenges allow the tag to detect if an adversary is trying to get the responses in advance. When the tag detects an adversary it stops sending responses. The protocol ends with a message to verify that no adversary has been detected.

The adversary can choose between two main attack strategies: asking in advance to the tag, taking the risk that the tag uncovers him, and without taking in advance and trying to guess the responses to the challenges when they occur. The adversary's success probability depends on p_f, the probability of the occurrence of full challenge, and can be calculated:

$$p_{MP} = \begin{cases} (1 - \frac{p_f}{2})^n, \text{if } \ p_f \leq \frac{4}{5} \text{ (without asking in advance)}, \\ (p_f \cdot \frac{3}{4})^n, \text{if } \ p_f > \frac{4}{5} \text{ (asking in advance)}. \end{cases} \quad (1)$$

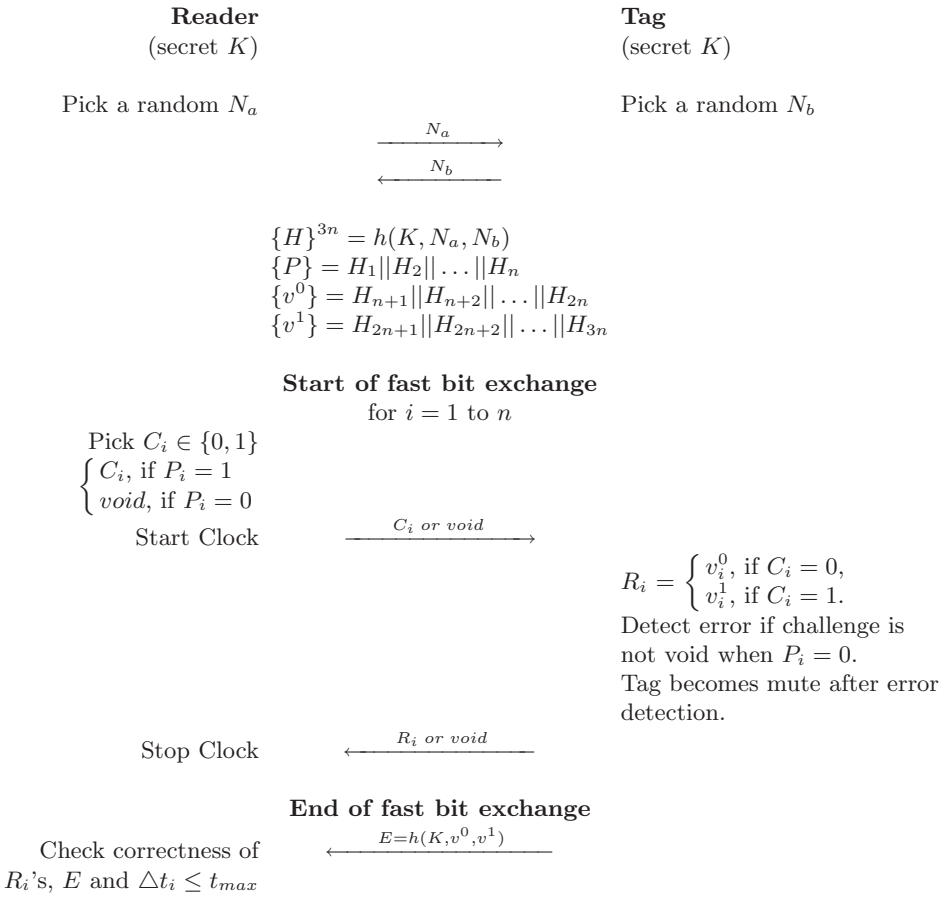

Fig. 2. Munilla et al.'s protocol

The adversary's success probability is the lowest when $p_f = 4/5$, but it is not easy to generate a bit string P with such value of p_f. However, the value $p_f = 3/4$ is close to $4/5$ and it 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 total $3n + 1$ bits ($2n$ bits for P, $n + 1$ bits for responses) are required to store. The success probability of the adversary is $(\frac{5}{8})^n$ if the string P is random [5], which is less than $(\frac{3}{4})^n$.

Note that the final confirmation message $h(K, v^0, v^1)$ does not take any challenges C_i as an input. So it can be pre-computed before the start of the fast bit exchange. On the other side, the disadvantage of their solution is that it requires three (physical) states: 0, 1, and *void*, which may be difficult to implement. Furthermore the success probability of the adversary is higher than $(\frac{1}{2})^n$.

2 Distance bounding protocol using mixed challenges

2.1 Description

To overcome the disadvantage of Munilla et al.'s protocol, we present a modification using *mixed challenges*: the challenges from the reader to the tag in the fast bit exchanges are divided into two categories, *random challenges* and *predefined challenges*. The earlier are random bits from the reader and the latter are predefined bits known to both the reader and the tag in advance.

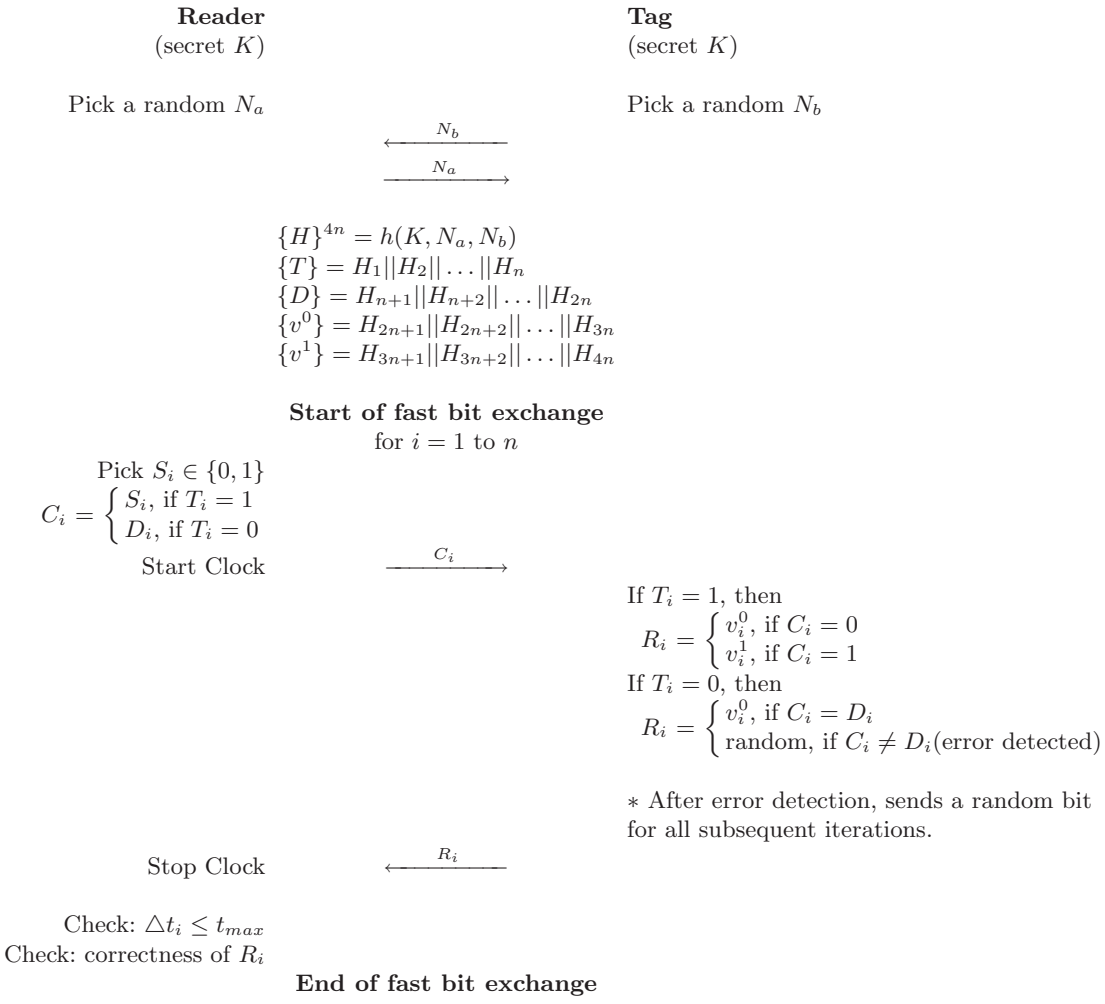

Fig. 3. Distance bounding protocol using mixed challenges

As shown in Fig. 3, the reader and the tag compute $4n$-bit sequence for $T||D||v^0||v^1$, after exchange of random nonces (N_a and N_b). The string T indicates random-challenges: if $T_i = 1$ the reader sends a random bit $S_i \in \{0, 1\}$ and if $T_i = 0$ it sends a predefined bit D_i to the tag. From the point of the adversary's view, all C_i's from the reader look like random.

Therefore he cannot distinguish random challenges from predefined challenges. However, thanks to these predefined challenges, the tag is able to detect an adversary sending random challenges in order to get responses in advance. Upon reception of a challenge C_i from the reader, either $T_i = 1$ (random challenge), in this case the tag sends out the bit $v_i^{C_i}$; or $T_i = 0$ (predefined challenge), in this case the tag sends out the bit v_i^0 if $C_i = D_i$ and a random bit if $C_i \neq D_i$ (it detects an error).

From the moment the tag detects an error, it replies a random value to all the subsequent challenges. By doing this, both the reader and the tag fight the adversary.

We note that we do not use any confirmation message after the end of fast bit exchanges, which improves efficiency in terms of computation and communication compared to Munilla et al. [5].

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

In our protocol, the tag always replies with a random bit after 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 will simply conclude 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 consists for the tag in sending the complementary bits of the right answers once a received predefined challenge is wrong. In this way, the tag helps the reader to detect early that an attack occurs. Indeed, if the strategy of the adversary is to exactly send to the reader what he previously received from the tag, then his probability of success is 0 once he 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 he 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 the right response when $T_i = 0$ but sends the complementary one when $T_i = 1$. As the adversary cannot distinguish between $T_i = 0$ and $T_i = 1$, he cannot decide when he 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 as in the basic protocol, the tag may reply with the remaining D_is after an attack is detected. Indeed, after a wrong challenge is received, the D_is 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 detects that the answers match the D_is). Here, because the D_is are still used for the reader's challenges when $T_i = 0$, this variant gives to the adversary the ability to observe that he has been detected by the tag. He 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; however, 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 some random bits after an error is detected.

3 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 $p_d + p_r = 1$. We start the analysis from the noise-free case then analyze in the noisy case.

3.1 Analysis in the noise-free case

The adversary can choose between two main attack strategies. First he can guess the responses to the challenges without asking in advance to tag. Secondly he can ask in advance to the tag, taking the risk that the tag uncovers him. Let us denote the adversary's probability of success without asking as P_{no-ask} and that with asking as P_{ask}. The adversary's probability of success of Munilla et al.'s protocol depends on the probability of the occurrence of full or void challenges as shown in Eq. 1. Therefore P_{no-ask} and P_{ask} are different according to the probability of the occurrence of full challenges. With the recommended $p_f = 3/4$, the adversary's probability of the success without asking, P_{no-ask}, is $(5/8)^n$, which is higher than P_{ask}. Therefore it is better for the adversary to choose the first attack approach in Munilla et al.'s protocol.

In our proposed protocol the adversary's probability of the success without asking in advance, P_{no-ask}, is always $(1/2)^n$. Therefore we compute P_{ask} and compare it with P_{no-ask} in the next section.

Adversary's probability of success of not being detected by reader. To compute P_{ask}, we assume that an adversary asks in advance to tag, taking the risk that the tag uncovers him. If the challenge, C_i^*, that the adversary asks to the tag in advance is the same than the challenge, C_i, that the reader sends to the tag, he sends the response received from the tag to the reader. If $C_i^* \neq C_i$, then he sends a random response to the reader.

Although the adversary's attack is detected by the tag there is still a chance of not being detected by the reader. To analyze this probability, we define some events as follows:

- A_i: the event that the adversary's attack is detected by the reader in the i^{th} round,
- $\overline{A}_i$: the event that the adversary's attack is *not* detected by the reader in the i^{th} round,
- B_i: the event that the adversary's attack is detected by the tag in the i^{th} round,
- $\overline{B}_i$: the event that the adversary's attack is *not* detected by the tag in the i^{th} round.

The event of not being detected by the reader in the i^{th} round, $\overline{A}_i$, depends on the event of being detected by the tag in the $(i-1)^{th}$ round. Because tag gives random values once it detects an error and the adversary does not know that it is an original or a random value.

The probability of not being detected by the reader in the i^{th} round provided that it is not detected by the tag in the $(i-1)^{th}$ round, $P(\overline{A}_i|\overline{B}_{i-1})$, is

$$\begin{aligned}
P(\overline{A}_i|\overline{B}_{i-1}) &= P_{\text{random ch. \& not detected}} + P_{\text{predefined ch. \& not detected}} \\
&= p_r \cdot (P_{C_i^*=C_i\text{\& not det. by Reader}} + P_{C_i^*\neq C_i\text{\& not det. by Reader}}) \\
&\quad + p_d \cdot (P_{C_i^*=C_i\text{\& not det. by Reader}} + P_{C_i^*\neq C_i\text{\& not det. by Reader}}) \\
&= p_r(\frac{1}{2}\cdot 1 + \frac{1}{2}\cdot\frac{1}{2}) + p_d(\frac{1}{2}\cdot 1 + \frac{1}{2}\cdot\frac{1}{2}) \\
&= \frac{3}{4}p_r + \frac{3}{4}p_d \\
&= \frac{3}{4}. \qquad (2)
\end{aligned}$$

If the challenge is random and $C_i^* = C_i$, then the adversary can correctly answer the response. If the challenge is random and $C_i^* \neq C_i$, then the adversary have a chance of $\frac{1}{2}$ of giving a correct response. If the challenge is predefined and $C_i^* = C_i$, he can correctly answer the response. If the challenge is predefined and $C_i^* \neq C_i$, he has a chance of $\frac{1}{2}$ of giving a correct response to the reader although he is always detected by the tag.

The probability of not being detected by the reader in the i^{th} round provided that it is detected by the tag in the $(i-1)^{th}$ round, $P(\overline{A}_i|B_{i-1})$, is

$$\begin{aligned}
P(\overline{A}_i|B_{i-1}) &= P_{\text{random ch. \& not detected}} + P_{\text{predefined ch. \& not detected}} \\
&= p_r \cdot (P_{C_i^*=C_i\text{\& not det. by Reader}} + P_{C_i^*\neq C_i\text{\& not det. by Reader}}) \\
&\quad + p_d \cdot (P_{C_i^*=C_i\text{\& not det. by Reader}} + P_{C_i^*\neq C_i\text{\& not det. by Reader}}) \\
&= p_r(\frac{1}{2}\cdot\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2}) + p_d(\frac{1}{2}\cdot\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2}) \\
&= \frac{1}{2}p_r + \frac{1}{2}p_d \\
&= \frac{1}{2}. \qquad (3)
\end{aligned}$$

If the challenge is random and $C_i^* = C_i$, then the adversary sends the same response from the tag (not knowing that he was detected in the previous round). This response from the tag is a random value as the tag detected an error in the previous round. So he has a chance of $\frac{1}{2}$ of not detected by the reader. If the challenge is random and $C_i^* \neq C_i$, then the adversary have a chance of $\frac{1}{2}$ of giving correct response. Because he chooses a random response by himself. If the challenge is predefined and $C_i^* = C_i$, he again sends the same response from the tag. Therefore he has a chance of $\frac{1}{2}$ of not being detected by the reader. If the challenge is predefined and $C_i^* \neq C_i$, he has a chance of $\frac{1}{2}$ of giving a correct response to the reader as he chooses a random response by himself.

The probability of not being detected by the reader in the i^{th} round is computed by

$$\begin{aligned}
P(\overline{A}_i) &= P(\overline{A}_i|\overline{B}_{i-1})P(\overline{B}_{i-1}) + P(\overline{A}_i|B_{i-1})P(B_{i-1}) \\
&= \frac{3}{4}P(\overline{B}_{i-1}) + \frac{1}{2}P(B_{i-1}), \qquad (4)
\end{aligned}$$

where, $P(\overline{A}_1) = \frac{3}{4}$ and $i = 2, 3, ...$

The probability of being detected by the tag in the i^{th} round depends on the probability of being detected by the tag in the $(i-1)^{th}$ round. Therefore we have

$$P(B_i) = P(B_i|B_{i-1})P(B_{i-1}) + P(B_i|\overline{B}_{i-1})P(\overline{B}_{i-1}). \tag{5}$$

The $P(B_i|B_{i-1})$ is always 1 since the tag already detected an error in the $(i-1)^{th}$ round. The $P(B_i|\overline{B}_{i-1})$ is $\frac{1}{2}p_d$ since the tag can detect an error when the challenge is a predefined one and $C_i^* \neq C_i$. Therefore we can rewrite Eq 5 as follows:

$$P(B_i) = P(B_{i-1}) + \frac{1}{2}p_d P(\overline{B}_{i-1}) \tag{6}$$

$$= P(B_{i-1}) + \frac{1}{2}(1-p_r)P(\overline{B}_{i-1}), \tag{7}$$

where, $P(B_0) = 0$. From Eq. 4 and Eq. 7, we compute:

$$P(\overline{A}_i) = \frac{1}{2} + \frac{1}{4}\left(\frac{1}{2} + \frac{1}{2}p_r\right)^{i-1}.$$

When $p_r = \frac{1}{2}$, we obtain:

$$P(\overline{A}_i) = \frac{1}{2} + \frac{1}{4}\left(\frac{3}{4}\right)^{i-1}.$$

For example, when $p_r = \frac{1}{2}$, $P(\overline{A}_1) = \frac{3}{4}$ $P(\overline{A}_2) = \frac{11}{16}$, $P(\overline{A}_3) = \frac{41}{64}$, $P(\overline{A}_4) = \frac{155}{256}$, etc.

We depict the probabilities of not being detected by the reader by varying p_r and n in Fig. 4. For the comparison we also show the success probabilities of the adversary of HKP and MP. The adversary's probability of success of our protocol is smaller than those of HKP and MP. And the adversary's probability of success with asking, P_{ask}, is higher than that with no asking, $P_{no-ask} = (\frac{1}{2})^n$. So it is better for the adversary to choose the strategy of asking advance. However as the number of iterations increases, the success probability of the adversary approaches $(\frac{1}{2})^n$.

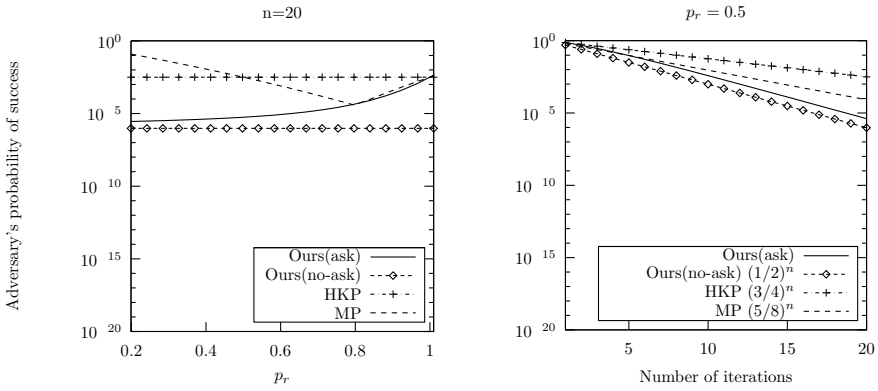

Fig. 4. Probability of not being detected by the reader

Distance fraud attack. Until now, we suppose that the tag is honest and the adversary tries to perform a mafia fraud attack. In this section, we consider the case of a dishonest tag. That is, we analyze the distance fraud attack on our protocol.

The tag knows the predefined challenges before the start of the fast bit exchanges as he knows T. Therefore he may try to deceive the reader with distance fraud attack. The probability of the success of the distance fraud attack by the dishonest tag for a round is

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

If the challenge is a random challenge ($T_i = 1$) and i^{th} bits of v^0 and v^1 are equal, then the tag can send its response early. If i^{th} bits of v^0 and v^1 are not equal when $T_i = 1$, then the tag chooses the response randomly. Finally if the challenge is a predefined challenge ($T_i = 0$), then he can send its response early.

We depict the success probabilities according to the variation of p_r and n in Fig. 5. If $p_r = \frac{1}{2}$, which is the average case when we use a pseudorandom function to generate a bit string, then we have $(\frac{7}{8})^n$.

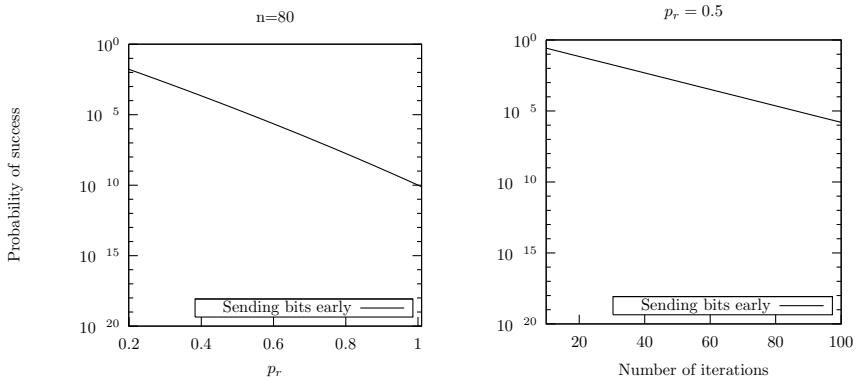

Fig. 5. Probability of distance fraud attack

3.2 Analysis in the noisy case

In a real application, there may exist errors due to the channel noise although the adversary does not attack. Therefore, we need to allow some channel errors for practical reasons. However the adversary may get benefit from this allowance of channels errors. We analyze the success probability of the adversary in the noisy case.

We assume that the maximum $(j-1)$ errors are allowed in the tag. That is, the tag starts to send random values after j^{th} error is detected. We define some events as follows:

- A_i: the event that the error is detected by the reader in the i^{th} round,
- $\overline{A}_i$: the event that the error is *not* detected by the reader in the i^{th} round,
- B_i^j: the event that total j errors are detected by the tag until the i^{th} round,
- $\overline{B}_i^j$: the event that less than j errors are detected by the tag until the i^{th} round.

Then we have

$$\begin{aligned} P({B^j}_i) &= P({B^j}_i|{B^j}_{i-1})P({B^j}_{i-1}) + P({B^j}_i|{\overline{B}^j}_{i-1})P({\overline{B}^j}_{i-1}) \\ &= P({B^j}_i|{B^j}_{i-1})P({B^j}_{i-1}) + P({B^j}_i|{B^{j-1}}_{i-1})P({B^{j-1}}_{i-1})P({\overline{B}^j}_{i-1}). \end{aligned}$$

The probability of $P({B^j}_i|{B^j}_{i-1})$ is 1 as j errors are already detected in $(i-1)^{th}$ round. The probability of $P({B^j}_i|{\overline{B}^j}_{i-1})$ is equal to $P({B^j}_i|{B^{j-1}}_{i-1})\ P({B^{j-1}}_{i-1})$. Because the j^{th} error can be detected in i^{th} round only if $j-1$ errors are detected in the $(i-1)^{th}$ round.

The probability of $P({B^j}_i|{B^{j-1}}_{i-1})$ is $\frac{1}{2}(1-p_r)$. Because the tag can detect an error only when the challenge is predefined and $C_i^* \neq C_i$. The tag starts to send random values after it detects j^{th} error, the probability of not being detected by the reader, $P(\overline{A}_i)$, depends on the $P(B_i^j)$. Once tag detects j^{th} error, probability of not being detected by the reader becomes $\frac{1}{2}$. Otherwise, probability of not being detected by the reader is $\frac{3}{4}$. Therefore we have

$$\begin{aligned} P(\overline{A}_i) &= P(\overline{A}_i|\overline{B}_{i-1}^j)P(\overline{B}_{i-1}^j) + P(\overline{A}_i|B_{i-1}^j)P(B_{i-1}^j) \\ &= \frac{3}{4}P(\overline{B}_{i-1}^j) + \frac{1}{2}P(B_{i-1}^j), \end{aligned} \tag{8}$$

where, $P(\overline{A}_1) = \frac{3}{4}$ and $i = 2, 3, ...$

For example, suppose that $j = 2$ and $p_r = 0.5$. It means that only one error is allowed in the tag and the tag sends random values after it detects two errors. Then, we have

$$\begin{aligned} P({B^2}_i) &= P({B^2}_i|{B^2}_{i-1})P({B^2}_{i-1}) + P({B^2}_i|{B^1}_{i-1})P({B^1}_{i-1})P({\overline{B}^2}_{i-1}), \\ &= P({B^2}_{i-1}) + \frac{1}{4}P({B^1}_{i-1})P({\overline{B}^2}_{i-1}). \end{aligned} \tag{9}$$

To compute $P({B^2}_i)$, we need to compute $P({B^1}_i)$ that is the same with $P(B_i)$ in the previous section. Therefore, we have $P(B_1^1) = \frac{1}{4}$, $P(B_2^1) = \frac{7}{16}$, $P(B_3^1) = \frac{37}{64}$, etc. The $P(B_1^2) = 0$ as two errors can not be detected in the first round. From Eq. 9, we have $P(B_2^2) = \frac{1}{16}$, $P(B_3^2) = \frac{169}{1024}$, etc. Finally we have $P(\overline{A}_1) = \frac{3}{4}$ $P(\overline{A}_2) = \frac{3}{4}$, $P(\overline{A}_3) = \frac{46}{64}$, $P(\overline{A}_4) = \frac{2734}{4096}$, etc. from Eq. 8. The probability of not being detected by reader until i^{th} round, $P_{ask}[i] = \prod_{k=1}^{i} P(\overline{A}_i)$. We depict it in Fig. 6.

Now we assume that the maximum $(j-1)$ errors are allowed in the reader. That is, the reader decides that the attack occurs after j^{th} error is detected. Contrary to the error detection by the tag, the detection of the error by the reader does not change $P(A_i)$ nor $P(B_i)$. Therefore the probability of not being detected by reader until i^{th} round, $P_{ask}[i]$ is

$$P_{ask}[i] = \begin{cases} \sum_{\mathbb{C}} \prod_{k=1}^{i} \{a_k P(A_k) + \overline{a}_k P(\overline{A}_k)\}, & i \geq j, \\ 1, & i < j, \end{cases}$$

Where, $a_k \in \{0, 1\}$ and $\mathbb{C} = \{(a_1, ..., a_i) | \sum a_k < j\}$. That is, $P_{ask}[i]$ is the probability that less than j errors are detected by the reader until i^{th} round. For example, if $j = 2$, then we

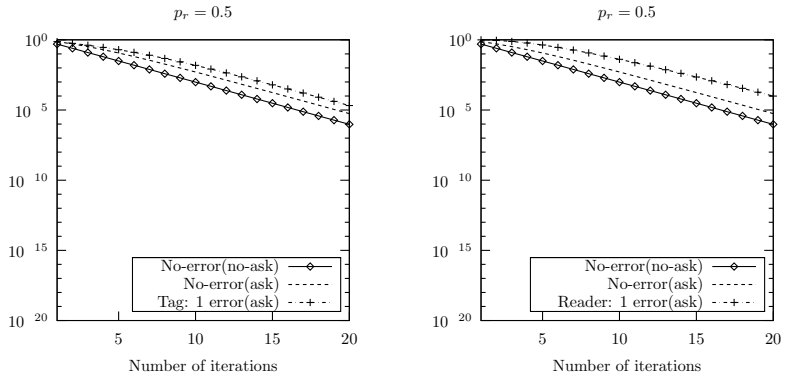

Fig. 6. Probability of not being detected by the reader in noisy channel: left) when tag allows errors, right) when reader allows errors

have

$$
\begin{aligned}
P_{ask}[1] &= 1,\\
P_{ask}[2] &= P(\overline{A}_1)P(\overline{A}_2) + P(A_1)P(\overline{A}_2) + P(\overline{A}_1)P(A_2),\\
P_{ask}[3] &= P(\overline{A}_1)P(\overline{A}_2)P(\overline{A}_3) + P(A_1)P(\overline{A}_2)P(\overline{A}_3) + P(\overline{A}_1)P(A_2)P(\overline{A}_3) +\\
&\quad P(\overline{A}_1)P(\overline{A}_2)P(A_3),\\
P_{ask}[4] &= P(\overline{A}_1)P(\overline{A}_2)P(\overline{A}_3)P(\overline{A}_4) + P(A_1)P(\overline{A}_2)P(\overline{A}_3)P(\overline{A}_4) +\\
&\quad P(\overline{A}_1)P(A_2)P(\overline{A}_3)P(\overline{A}_4) + P(\overline{A}_1)P(\overline{A}_2)P(A_3)P(\overline{A}_4) +\\
&\quad P(\overline{A}_1)P(\overline{A}_2)P(\overline{A}_3)P(A_4),\\
P_{ask}[5] &= ...
\end{aligned}
$$

If $p_r = 0.5$, we have $P_{ask}[i] = \{1, 0.922, 0.776, 0.600, 0.432, ...\}$. We depict the result in Fig. 6.

4 Conclusion

Relay attack is one of the most serious problems in RFID and contactless smartcard applications. Distance bounding protocols prevent relay attacks by computing the distance between the reader and the tag, where they measure the round-trip time between the reader and the tag. The Hancke and Kuhn proposed a distance bounding protocol with the probability of the adversary's success of $(\frac{3}{4})^n$, where n is a security parameter. After that, many tried to decrease the probability of the adversary's success to $(\frac{1}{2})^n$. Almost all of them used signed messages that made the protocol to be slower due to the computation and communication of a signing message. Munilla et al.'s approach does not use signed messages but requires three (physical) states: 0, 1, and *void*, which is difficult to implement.

In this paper, we proposed a new protocol with (binary) *mixed challenges* that provides the best performances among all the existing distance bounding protocols with binary challenges that do no use final signature. Indeed, while all these protocols provide a probability of the adversary success equal to $(3/4)^n$, the probability quickly converges toward $(1/2)^n$ in our case.

References

1. 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.
2. 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.
3. 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.
4. J. Munilla, A. Ortiz, and A. Peinado. Distance bounding protocols with void-challenges for RFID. In *Workshop on RFID Security - RFIDSec '06*, 2006.
5. 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*. Published online: Jan 17 2008, an earlier version appears in [4].
6. 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.
7. Y.-J. Tu and S. Piramuthu. RFID distance bounding protocols. In *the 1st International EURASIP Workshop in RFID Technology*. Vienna, Austria.