9-2013

A highly efficient RFID distance bounding protocol without real-time PRF evaluation

Yunhui ZHUANG

Anjia YANG

Duncan S. WONG

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Qi XIE

A Highly Efficient RFID Distance Bounding Protocol without Real-Time PRF Evaluation

Yunhui Zhuang[1], Anjia Yang[1], Duncan S. Wong[1], Guomin Yang[2], and Qi Xie[3]

[1] City University of Hong Kong, Hong Kong
{yhzhuang2-c,ayang3-c}@my.cityu.edu.hk,
duncan@cityu.edu.hk
[2] University of Wollongong, Australia
gyang@uow.edu.au
[3] Hangzhou Normal University, China

Abstract. There is a common situation among current distance bounding protocols in the literature: they set the fast bit exchange phase after a slow phase in which the nonces for both the reader and a tag are exchanged. The output computed in the slow phase is acting as the responses in the subsequent fast phase. Due to the calculation constrained RFID environment of being lightweight and efficient, it is the important objective of building the protocol which can have fewer number of message flows and less number of cryptographic operations in real time performed by the tag. In this paper, we propose a new highly efficient mutually-authenticated RFID distance bounding protocol that enables pre-computation which is carried out off-line by the tag. There is no evaluation on any PRF during the real time protocol running which makes the tag significantly more efficient at a low-cost. The protocol requires only O(1) complexity for achieving tag privacy. In addition, we give a detailed security analysis to prove that our protocol is secure against all common attacks in distance bounding.

Keywords: RFID, Distance Bounding, Privacy, Mutual Authentication.

1 Introduction

Radio Frequency IDentification (RFID) technology mainly consists of tags and readers that can be used to identify and encode a variety of information. It has been widely applied in many applications in the modern world. For example, the building access control, library book borrowing services, and E-channel for immigration, etc. In general, there are two types of RFID tags, namely active and passive tags. Active tags contain an internal power source while the low-cost passive tags don't. Nowadays, many RFID-enabled authentication protocols are based on symmetric-key encryption system in order to keep them low-cost.

In 1987, Desmedt et al. [4] introduced the Mafia fraud that could defeat any authentication protocol. An adversary can successfully pass the protocol by

J. Lopez, X. Huang, and R. Sandhu (Eds.): NSS 2013, LNCS 7873, pp. 451–464, 2013.

relaying the messages between the legitimate reader and a remote legitimate tag. One way to prevent such attack is using distance-bounding protocol. It was first designed by Brands and Chaum in 1993 [1]. The concept of distance bounding is based on the combination of distance checking and authentication, under the measurement of the *Round Trip Time* (RTT) of messages exchanged by the reader and a tag. Based on RTT, the reader can evaluate the distance between itself and a tag in order to compare the value with an upper bound which can be estimated according to the assumption that nothing propagates faster than light. Brands and Chaum's protocol is too expensive in practice because there is a signature at the end in order to realize mutual authentication. In 2005, Hancke and Kuhn [2] designed another protocol without the final signature that contains only one slow phase and one fast bit exchange phase. Their protocol has been treated as a key-reference in the state-of-art publications regarding to RFID distance-bounding.

Since then quite a few distance bounding protocols have been published [3,7,8,9,10,11,14,15,16,17]. There are five common attacks in RFID distance bounding scenario: Impersonation fraud [1], Distance fraud [1], Mafia fraud [4], Terrorist fraud [4], and Distance hijacking attack [6]. In this paper, we only consider distance hijacking attack in the single-protocol environment defined in [6]. In 2011, Avoine et al. [5] used secret-sharing scheme to defeat terrorist frauds. They made the conclusion that at least a (3, 3) threshold secret-sharing scheme should be applied to resist terrorist fraud, while most existing works only used (2, 2) schemes that is susceptible to the terrorist fraud attack.

We introduce in this paper a prominent feature called "Pre-Computation" in RFID distance bounding protocols. This idea let us break away from traditional approach that the slow phase should always be ahead of the fast phase. Actually, the computation in the very beginning can be carried out off-line. In fact, the pre-computation in RFID is not new [13], but it has never been deployed in distance bounding. The existing protocols proposed recently require the tag to perform one/more time-consuming PRFs or signatures in real time. It is susceptible to the high power and high cost. The pre-computation is done by an ultra low power micro-controller which is powered by a large capacitor. It has been implemented and proved in [12]. And most important of all, the cost for planting a large capacitor in an RFID tag is negligible.

We find most distance bounding protocols use the idea of the fast bit exchange by transmitting only one-bit challenges for each round. In fact, the communication channels used in nowadays have a much bigger bandwidth to transmit more than one bit. As pointed out in [16], the two-bit challenges sent in the fast phase can be encapsulated to a much bigger packet over the communication channels. In addition, [18] pointed out a practical terrorist attack to the protocol proposed by Yang et al.[15] due to only one-bit challenges sent in each round. Having these observations in mind, our proposed protocol is designed by adopting two-bit challenges in the fast phase in order to prevent such attacks and make better use of the communication channels.

Our Contributions
In this paper, we combine all the features described above in an RFID authentication system and propose a highly efficient RFID distance bounding protocol with tag privacy.

The protocol features pre-computation on the tag, mutual authentication, resistance to all common attacks, and significantly more efficient at a low cost. It eliminates all online PRF evaluation and leaves only two if-else decisions to make in runtime for the tag. One more advantage of this elimination can minimize the processing time for response and make the propagation time of the bits dominate the round trip time (RTT), and at the same time, make the response processing time as invariant as possible. Consequently, we can get a more accurate estimation on the distance between the reader and a tag. To the best of our knowledge, our protocol is the first distance bounding protocol that realizes the tag online PRF-free by introducing the concept of pre-computation.

We also provide privacy-preserving in our protocol by an anonymous way, which requires only O(1) complexity for achieving privacy. We show our protocol is much more efficient in terms of tag's cost when compared with existing ones.

To show our contributions more precisely, we make a detailed comparison between our proposed protocol and others in Table 2, Section 4.

Paper Organization
The rest of the paper is organized as follows. In Section 2, we show our protocol with detailed description. In section 3, we give the security analysis with respect to the five attacks and how reader authentication is realized. In section 4, there will be a comparison between our protocol and previous proposed protocols. In the last section, we conclude the paper.

2 Our Proposed Protocol

In this section, we first give some preliminaries including the system description, the adversary model, and the definition of Pseudo-Random Function that we used as the underlying cryptographic primitive. Then we describe our proposed protocol in detail. At last, we have a discussion of several important issues in our protocol.

2.1 Preliminaries

System Description. The RFID system consists of multiple tags T_1,T_2,$\cdots$,T_n and a reader R, associated with a database. Each tag T_i stores a secret key x_i which is shared with the reader R, its identity ID, pseudonym ID', as well as the counter N_T which is initialized to zero. The reader maintains tag's identity ID, counter N'_T as well. In addition, the reader also maintains TID and TID' for achieving tag privacy. The reader and a tag communicate via the wireless channel. The upper bound for the transmission speed cannot exceed the speed of light.

Adversary's Capabilities. It is important to define a generic model for adversary's capabilities in a realistic and fair condition. In our model, an adversary $\mathcal{A}$ can be "active" which means she can eavesdrop, intercept, modify messages. $\mathcal{A}$ can control the transmission time between the reader and a tag. But $\mathcal{A}$ cannot perform unlimited computations. In addition, we assume that an honest tag will not give its security parameters to any third party.

Pseudo-random Function. Our protocol uses an *Pseudo-Random Function* (PRF) as the underlying cryptographic primitive. A family of efficiently computable functions $\boldsymbol{f} = \{F_K : \mathcal{D} \to \mathcal{R} | K \in \mathcal{K}\}$ is called a pseudo-random function family, if for any polynomial time algorithm $\mathcal{C}$,

$$\mathbf{Adv}^{prf}_{\boldsymbol{f},\mathcal{C}}(k) = \mathbf{Pr}[\mathcal{C}^{F_K(\cdot)}(1^k) = 1] - \mathbf{Pr}[\mathcal{C}^{\mathsf{RF}(\cdot)}(1^k) = 1].$$

is a negligible function of the security parameter k, where K is randomly selected from the key space $\mathcal{K}$, F_K is an instance of function family $\boldsymbol{f}$, and $\mathsf{RF} : \mathcal{D} \to \mathcal{R}$ is a truly random function.

2.2 Protocol Description

Our protocol has two stages, namely the pre-computation stage and real-time stage as shown in Figure 1.

Pre-computation Stage. We introduce two flag bits to facilitate the steps during the pre-computation stage: $Flag^{pre}$ and $Flag^{sync}$, where the former indicates whether the pre-computation has been done successfully; while the latter is to determine whether the reader authentication was successful in last execution. The $Flag^{pre}$ should be set to 1 before performing the pre-computation by the tag. In the meantime, $Flag^{sync}$ has to be 1 before updating pseudonym ID'.

We use a counter N_T (initialized to zero) as one of the three inputs to compute the Pseudo-Random Function (PRF) f and N_T should be updated each time at the very beginning whenever the tag is powered up. It is also worth mentioning that N_T cannot be a random number. Otherwise, it may suffer from the replay attack due to the absence of reader's nonce.

Because the contents of the input for computing of f and ID' are independent from the reader. Therefore, they can be computed before the protocol starts. This stage can cope with the limited resources of RFID tags, who will compute $v = f(x, ID, N_T)^{3n}$. Then split v into three shares: v^1, v^2, and v^3, respectively. Each of them carries n bits. After that the tag needs to check the value of $Flag^{sync}$ such that it will update ID' in pre-computation only when $Flag^{sync}=1$, which means the reader authentication was successful in the last execution. The update is computed as $ID' = f(x, ID')$. In the meanwhile, the tag is going to flip $Flag^{pre}$ to 0 indicating that the pre-computation has been finished. The tag now has the updated value of v^1, v^2, v^3, and k for running the real-time stage.

The pre-computation is carried out off-line as follows: by using an ultra low power micro-controller and a large capacitor, each time during the protocol running, the tag will receive enough RF (Radio Frequency) energy and rectifies it

Reader (x,ID,TID,TID',N_T')		**Tag** (x,ID,ID',N_T)
Pre-Computation Stage		
		$Flag^{pre} = 0$ $N_T = N_T + 1$ $v^1 \| v^2 \| v^3 = f(x, ID, N_T)^{3n}$ $k = v^1 \oplus v^2 \oplus x$ **If** $Flag^{sync} = 1$ **then** $ID' = f(x, ID')$ $Flag^{sync} = 0$
Real-Time Stage		
		If $Flag^{pre} = 1$ **then** Run Pre-Computation $Flag^{pre} = 1$
	Start of Fast Phase *for* $i = 1$ *to* n	
Pick $C_i, D_i \in_R \{0,1\}$ Start Clock;	Send $[C_i D_i]$ →	$R_i = \begin{cases} v_i^1, & C_i' D_i' = 00 \\ v_i^2, & C_i' D_i' = 11 \\ k_i, & C_i' D_i' = 01 \text{ or } 10 \end{cases}$
Stop Clock; Store R_i, $\triangle t_i$	← Send R_i	
	End of Fast Phase	
	← Send N_T, ID'	
If $N_T < N_T'$ **then** Reject $N_T' = N_T + 1$		
If $ID' \neq TID$ **And** $ID' \neq TID'$ **then** Reject **Else If** $ID' = TID'$ **then** $TID = TID', TID' = f(x, TID)$		
Compute $u^1 \| u^2 \| u^3 = f(x, ID, N_T)^{3n}$ **For** $i = 1$ **to** n **set** R_i' based on $C_i D_i$		
Check $e_1 = \sharp\{i : R_i' \neq R_i\}$ $e_2 = \sharp\{i : \triangle t_i > \triangle t_{max}\}$ **If** $e_1 + e_2 > T$ **then** Reject **Else** Success		
	Send u^3 →	**If** $u^3 = v^3$ **then** $Flag^{sync} = 1$ Success **Else** Reject

Fig. 1. The Protocol without Real-Time PRF Evaluation

into DC (Direct Current) voltage stored in the large capacitor. After protocol finished, the tag will use this stored DC voltage to power the system in order to compute two PRFs and then stored in the non-volatile memory for next round protocol execution.

Real-Time Stage. The real-time stage consists of one fast bit exchange phase (a.k.a. fast phase), which has total n rounds, and one slow phase in which mutual authentication is provided. The communication channel used during the fast phase may suffer from noises. Hence the reader should setup a checking mechanism by a given error threshold (Fault Tolerance). The reader must abort the protocol if the threshold has been exceeded. This stage requires no PRF evaluation but only if-else decisions to make for the tag.

Before starting the fast phase, the tag needs to check the status of the flag bit once more. If $Flag^{pre} = 1$, the tag is aware that the pre-computation is not completed due to several reasons (details in Section 2.3). Under whatever circumstances, the tag needs to ensure the pre-computation has been completely done before running the fast phase. Therefore, the tag should perform the pre-computation in real time for once if $Flag^{pre} = 1$. In contrast, if the tag identifies that $Flag^{pre} = 0$, which means the pre-computation stage has been successfully finished, then it flips $Flag^{pre}$ to 1. The protocol now moves to the fast phase:

(1) The reader randomly picks two-bit challenge C_iD_i, starts the clock and sends C_iD_i to the tag.
(2) The tag sends corresponding R_i according to both $C'_iD'_i$.
(3) Upon receiving R_i, the reader immediately stops the clock, stores the time delay $\triangle t_i$, and R_i. There will be no checking at this time.
(4) Above three steps are repeated for n rounds.

When proceed to the last slow phase, no time delays are measured:

(1) The tag sends the counter N_T together with its pseudonym ID' to the reader.
(2) The reader then produces the checking procedures by means of several if-else decision makings. Note that the reader's database also maintains N'_T, TID and TID', where N'_T is the tag's counter which maintained on the reader side; TID and TID' are used as the index to quickly search the tag's ID, they are initialized as $TID = f(x, ID)$ and $TID' = f(x, f(x, ID))$, respectively. There are three parts during the checking mechanism.

(2.1) **Counter Checking.** When received all the information from the tag, the reader will first check whether the received counter N_T is equal to or greater than its stored value N'_T. If N_T is small than N'_T, the reader is going to reject the tag and abort the protocol in the sense that a replay attack has been launched because an honest tag will never use an old counter value when initiating a new protocol execution. If it is satisfied, then the reader's counter N'_T will be updated as $N'_T = N_T + 1$. Otherwise the reader will reject the tag and leave the counter unchanged.

(2.2) **Index Searching on Tag's ID.** After checking the counter, the reader moves to the 2nd part by comparing ID' with either TID or TID'. If none of them is equal to ID', there may be an attack launched by the adversary (i.e. de-synchronization attack). Therefore, the reader will reject the tag and abort the protocol immediately. In contrast, if ID' is indeed equal to TID', the reader is going to update its local stored TID and TID' to synchronize with the tag in the sense that there may be one step ahead by the tag. Similarly, if ID' is equal to TID, which means the reader has already catch up with the tag and no update is needed.

(2.3) **Fault Tolerance.** The reader is going to compute $u = f(x, ID, N_T)^{3n}$ and split it into three shares, u^1, u^2, and u^3, respectively. Each of them carries n bits. Based on challenge $C_i D_i$ picked in the fast phase, the reader should set R'_i in order to facilitate the fault tolerance. Now the reader will perform two concurrent checking on the validity of two different values:

– it counts the number of errors e_1 of positions for the responses $R'_i \neq R_i$;

– it counts the number of errors e_2 of the transmission delay $\triangle t_i > \triangle t_{max}$;

If $e_1 + e_2 > T$, where T is the fault tolerance threshold, the reader will reject the tag and abort the protocol. Otherwise, the reader can accept the tag.

(3) After above three checking parts, the reader is able to tell whether the protocol succeeds or not and sends u^3 to the tag for mutual authentication.

(4) Finally, the tag is going to check the validity of the u^3 computed by the reader and flip the flag bit $Flag^{sync}$ to 1 if reader authentication is successful. But on the tag side, the counter N_T is always updated at the very beginning no matter what decision the tag made.

2.3 Discussions

The Counter. Intuitively, the counter N'_T stored on the reader side should be synchronized with the counter N_T that stored on the tag. However, if some attacks are launched (i.e. de-synchronization attack), the tag's counter N_T is always greater than reader's N'_T. But it has no effect on the protocol execution in the sense that the checking mechanism only ensures N_T should be equal to or greater than N'_T to prevent replay attack.

The Flag Bit. The RFID chip can loose power at any time. If that happens, it might be possible to force a tag to reuse pre-computed values more than once. If there is no flag bit presented in the tag, the adversary is able to extract the secret key. Nevertheless, we use in our protocol two flag bits $Flag^{pre}$ and $Flag^{sync}$ to ensure the integrity of the RFID environment and the pre-computation has been completely done before the real-time protocol execution. The purpose for $Flag^{pre}$ is to guarantee the counter N_T is updated each time before computing f. If the $Flag^{pre} = 1$ before starting the fast phase, the tag is aware that either insufficient power stored in the large capacitor so that the tag cannot perform the pre-computation or some sort of attacks have been launched, such as the reset attack. Even this kind of attack has been identified, the tag only needs to perform

the pre-computation in real time once. This makes our protocol much more robust. On the other hand, $Flag^{sync}$ can prevent the de-synchronization attack in the sense that the tag should only update ID' after reader authentication is successful. Otherwise, if the adversary modifies/blocks u^3 twice, then the tag can no longer be identified by the reader anymore.

3 Security Analysis

We will make a detailed security analysis against all common attacks in the distance bounding protocols.

Impersonation Fraud Resistance. In the impersonation attack, the adversary $\mathcal{A}$ does not know the tag's secret key x and must correctly answer the challenge C_iD_i during the fast phase. Thus, the success probability of the impersonation attack for one round is given by:

$$P_{imp} = \mathbf{Pr}[\mathcal{A}\ guesses\ R_i\ correctly] = \frac{1}{2}$$

The overall success probability is $\left(\frac{1}{2}\right)^n$ since there are n rounds in the fast phase.

Distance Fraud Resistance. The adversary $\mathcal{A}$ is the tag itself in a distance fraud. There are three choices for $\mathcal{A}$ to launch the distance fraud attack. In addition, $\mathcal{A}$ has to carry out the early-reply strategy (to send each reply before receiving the challenges) for all choices during the fast phase in order to make the RTT within the threshold $\triangle t_{max}$.

(1) **Randomly Reply.** $\mathcal{A}$ can choose the most naive way to get a probability of $\frac{1}{2}$ for each round by randomly picking the responses regardless of reader's challenges. Therefore, the success probability for one round is given by:

$$P_{dis-1} = \mathbf{Pr}[\mathcal{A}\ randomly\ replies\ R_i] = \frac{1}{2}$$

Up to n rounds in the fast phase, the overall success probability is $\left(\frac{1}{2}\right)^n$.

(2) **Challenge Guessing.** $\mathcal{A}$ may perform PRF computation during the pre-computation stage to get v^1, v^2, v^3, and k, respectively. With this choice, $\mathcal{A}$ needs to guess the reader's challenges correctly and send the response R_i in advance. Hence the success probability for one round is given by:

$$\begin{aligned}
P_{dis-2} &= \mathbf{Pr}[\mathcal{A}\ guesses\ C_iD_i\ correctly] \\
&= \mathbf{Pr}[C_i'D_i' = 00 \mid C_iD_i = 00] \times \mathbf{Pr}[C_iD_i = 00] \\
&\quad + \mathbf{Pr}[C_i'D_i' = 11 \mid C_iD_i = 11] \times \mathbf{Pr}[C_iD_i = 11] \\
&\quad + \mathbf{Pr}[C_i'D_i' = 01\ or\ 10 \mid C_iD_i = 01] \times \mathbf{Pr}[C_iD_i = 01] \\
&\quad + \mathbf{Pr}[C_i'D_i' = 01\ or\ 10 \mid C_iD_i = 10] \times \mathbf{Pr}[C_iD_i = 10] \\
&= \left(\frac{1}{4}\cdot\frac{1}{4} + \frac{1}{4}\cdot\frac{1}{4} + \frac{1}{2}\cdot\frac{1}{4} + \frac{1}{2}\cdot\frac{1}{4}\right) = \frac{3}{8}
\end{aligned}$$

Up to n rounds in the fast phase, the overall success probability is $\left(\frac{3}{8}\right)^n$.

(3) **"Majority Vote" Attack.** Although the presence of PRF guarantees the pure random output each time, it could happen that v_i^1, v_i^2 and k_i will have the same value. By Dirichlet's Box principle[1], at least two of them are the same.

Table 1. "Majority Vote" Attack

v_i^1	0	0	0	0	1	1	1	1
v_i^2	0	0	1	1	0	0	1	1
k_i	0	1	0	1	0	1	0	1
Success Probability	1	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{2}$	1

Table 1 shows all possible success probabilities for three registers with respect to the "Majority Vote" attack. The 1st column provides three registers v_i^1, v_i^2, and k_i, together with the success probability for each i. There are eight different combinations for three registers in the table (From 2nd to 9th columns). With this attack, the adversary $\mathcal{A}$ can simply select the value which has a majority, that is, two or three equal registers (Majority wins) and reply this particular value to the reader. Since k_i is determined by either 01 or 10 for C_iD_i, it has the higher probability if the majority wins. Thus, the probability that $\mathcal{A}$ can succeed in this case is given by:

$$\begin{aligned} P_{dis-3} &= \mathbf{Pr}[Majority\ Wins] \\ &= \left(\frac{1}{8} \cdot \left(1 + \frac{1}{2} + \frac{3}{4} + \frac{3}{4} + \frac{3}{4} + \frac{3}{4} + \frac{1}{2} + 1\right)\right) = \frac{3}{4} \end{aligned}$$

Up to n rounds in fast phase, the overall success probability is $\left(\frac{3}{4}\right)^n$.

Remark 1. $\mathcal{A}$ may choose the "Majority Vote" attack since it provides the highest success probability among three different choices in distance fraud attack.

Mafia Fraud Resistance. The tag does not collude with the adversary $\mathcal{A}$ in the Mafia fraud. $\mathcal{A}$ may launch the attack by using one of the following strategies.

(1) **Post-ask strategy.** By acting as a malicious tag, $\mathcal{A}$ first executes the fast phase with the reader in order to learn the correct challenges C_iD_i. After knowing all challenges $\mathcal{A}$ pretends to be a fake reader and runs the fast phase with the legitimate tag so that $\mathcal{A}$ can obtain valid response R_i. At last, $\mathcal{A}$ relays the final slow phase. With this strategy, $\mathcal{A}$ has to answer to the reader with arbitrary answers. This strategy has the same probability as in the impersonation fraud. Thus, the success probability in the post-ask strategy is given by:

$$P_{maf-1} = \mathbf{Pr}[\mathcal{A}\ guesses\ R_i\ correctly] = \frac{1}{2}$$

For n rounds in fast phase, the overall success probability is $\left(\frac{1}{2}\right)^n$.

[1] Dirichlet's Box principle: Given n boxes and m ($m > n$) objects, if m objects are placed into n boxes, at least one box must contain more than one (m/n) object.

(2) **Pre-ask strategy.** $\mathcal{A}$ needs to pretend to be a fake reader and execute the fast phase with the tag before the reader to do so. Afterwards, $\mathcal{A}$ runs the fast phase by acting as the malicious tag with the reader and relays the final slow phase. With this strategy, $\mathcal{A}$ needs to transmit the anticipated challenge bits $C_i'D_i'$ to the tag before the reader sends out its real challenge C_iD_i. However, there are two special cases with this strategy.

(i) $\mathcal{A}$ chooses C_iD_i solely from [00,11,01,10]. Thus, the success probability in this case is given by:

$$\begin{aligned} P_{maf-2} &= \mathbf{Pr}[\mathcal{A}\ guesses\ C_iD_i\ correctly] \\ &\quad + \mathbf{Pr}[\mathcal{A}\ guesses\ C_iD_i\ incorrectly \wedge \mathcal{A}\ randomly\ replies\ R_i\ correctly] \\ &= \mathbf{Pr}[\mathcal{A}\ guesses\ C_iD_i\ correctly] + \frac{1}{2}[1 - \mathbf{Pr}[\mathcal{A}\ guesses\ C_iD_i\ correctly]] \\ &= \frac{1}{2}[1 + \mathbf{Pr}[\mathcal{A}\ guesses\ C_iD_i\ correctly]] \\ &= \left(\frac{1}{2}\left(1 + \frac{3}{8}\right)\right) = \frac{11}{16} \end{aligned}$$

The overall success probability is $\left(\frac{11}{16}\right)^n$ for n rounds in fast phase. This approach provides a higher success probability when compared with the post-ask strategy. But $\mathcal{A}$ may choose another special case.

(ii) $\mathcal{A}$ only sends $C_i'D_i'$=01 (or 10) to the tag in the first fast phase execution in order to obtain the whole share of k. Then $\mathcal{A}$ runs the second fast phase with the reader. Therefore, $\mathcal{A}$ can succeed in this case will be:

$$\begin{aligned} P_{maf-3} &= \mathbf{Pr}[\mathcal{A}\ replies\ R_i\ correctly \mid C_iD_i = 01\ or\ 10] \times \mathbf{Pr}[C_iD_i = 01\ or\ 10] \\ &\quad + \mathbf{Pr}[\mathcal{A}\ guesses\ R_i\ correctly \mid C_iD_i = 00\ or\ 11] \times \mathbf{Pr}[C_iD_i = 00\ or\ 11] \\ &= \left(1 \cdot \frac{2}{4} + \frac{1}{2} \cdot \frac{2}{4}\right) = \frac{3}{4} \end{aligned}$$

For n rounds in fast phase, the overall success probability is $\left(\frac{3}{4}\right)^n$.

Remark 2. For any strategy, the success probability is upper bounded by $\left(\frac{3}{4}\right)^n$. It is obvious that the pre-ask strategy has the higher success probability.

Terrorist Fraud Resistance. In a terrorist fraud attack, the malicious tag colludes with the adversary who will run the fast phase and relay the last slow phase on behalf of the malicious tag. The tag could give some sensitive information to the adversary so that she could defeat the protocol for one session. To be more specific, the malicious tag cannot give all registers v^1, v^2, and k to $\mathcal{A}$, since $\mathcal{A}$ will be able to recover the secret key x by $x = v^1 \oplus v^2 \oplus k$. But N_T and ID' can be passed to $\mathcal{A}$ directly. Hence there are three scenarios to be considered.

(1) $\mathcal{A}$ has k and v^1 (same probability for k and v^2) at hand. When receiving the challenge C_iD_i, $\mathcal{A}$ knows the exact response from v_i^1 or k_i. But $\mathcal{A}$ needs to guess the value of v_i^2 when $C_iD_i = 11$. Thus, for $\mathcal{A}$ has (k, v^1) , the success probability in this situation is given by:

$$\begin{aligned} P_{terr-1} &= \mathbf{Pr}[\mathcal{A}\ \textit{guesses}\ R_i\ \textit{correctly} \mid C_iD_i = 11] \times \mathbf{Pr}[C_iD_i = 11] \\ &\quad + \mathbf{Pr}[\mathcal{A}\ \textit{replies}\ R_i\ \textit{correctly} \mid C_iD_i \neq 11] \times \mathbf{Pr}[C_iD_i \neq 11] \\ &= \left(\frac{1}{2} \cdot \frac{1}{4} + 1 \cdot \frac{3}{4}\right) = \frac{7}{8} \end{aligned}$$

For n rounds in fast phase, the overall success probability is $\left(\frac{7}{8}\right)^n$

(2) $\mathcal{A}$ has both v^1 and v^2. $\mathcal{A}$ can reply with good answer when $C_i = D_i$. But $\mathcal{A}$ needs to randomly guess when $C_i \neq D_i$ since she has no knowledge of k. Then the success probability in this situation is given by:

$$\begin{aligned} P_{terr-2} &= \mathbf{Pr}[\mathcal{A}\ \textit{guesses}\ R_i\ \textit{correctly} \mid C_iD_i = 01\ \textit{or}\ 10] \times \mathbf{Pr}[C_iD_i = 01\ \textit{or}\ 10] \\ &\quad + \mathbf{Pr}[\mathcal{A}\ \textit{replies}\ R_i\ \textit{correctly} \mid C_iD_i = 00\ \textit{or}\ 11] \times \mathbf{Pr}[C_iD_i = 00\ \textit{or}\ 11] \\ &= \left(\frac{1}{2} \cdot \frac{2}{4} + 1 \cdot \frac{2}{4}\right) = \frac{3}{4} \end{aligned}$$

For n rounds in fast phase, the overall success probability is $\left(\frac{3}{4}\right)^n$

(3) It's the opposite of case (2) when $\mathcal{A}$ only obtains k. Therefore, the success probability in this situation is given by:

$$\begin{aligned} P_{terr-3} &= \mathbf{Pr}[\mathcal{A}\ \textit{replies}\ R_i\ \textit{correctly} \mid C_iD_i = 01\ \textit{or}\ 10] \times \mathbf{Pr}[C_iD_i = 01\ \textit{or}\ 10] \\ &\quad + \mathbf{Pr}[\mathcal{A}\ \textit{guesses}\ R_i\ \textit{correctly} \mid C_iD_i = 00\ \textit{or}\ 11] \times \mathbf{Pr}[C_iD_i = 00\ \textit{or}\ 11] \\ &= \left(1 \cdot \frac{2}{4} + \frac{1}{2} \cdot \frac{2}{4}\right) = \frac{3}{4} \end{aligned}$$

For n rounds in fast phase, the overall success probability is $\left(\frac{3}{4}\right)^n$

Distance Hijacking Attack Resistance. We only consider the distance hijacking attack in the single-protocol environment. Under this situation, the adversary $\mathcal{A}$ outside the legal authentication region exploits an inside legitimate tag to execute the fast phase so that $\mathcal{A}$ can cheat on its real distance to the reader. To launch such attack, $\mathcal{A}$ first does nothing during the fast phase as she is far away from the reader. When the fast phase ends, $\mathcal{A}$ will impersonate a fake reader to communicate with the exploited tag in order to get the counter N_T, and tag's pseudonym ID'. Upon receiving these information, $\mathcal{A}$ is going to act as a fraudulent tag to send N_T (untouched) and her own pseudonym $ID'_{\mathcal{A}}$ to the legitimate reader. Finally, the reader will make decision on acceptance of the fraudulent tag. It is obvious that $\mathcal{A}$ cannot win because she does not have the secret key x of the exploited tag. Besides, the $ID'_{\mathcal{A}}$ is different so that the output of the PRF is absolutely different. Therefore, the success probability of $\mathcal{A}$ is $(\frac{1}{2})^n$.

Reader Authentication. Up to now, many distance bounding protocols do not feature reader authentication. They focus on unilateral authentication where the tag tries to convince the reader of a statement related identity and the physical distance between them. They make the assumption that the reader should be honest, but we would like to argue that this may not be the case when considering

the Mafia fraud attack, the adversary launches the attack by exchanging the roles of the reader and a tag. Therefore, it is crucial to support mutual authentication in distance bounding protocol as well. In fact, our protocol is the one providing reader authentication by introducing v^3. The presence of v^3 as one of the three registers let the tag be able to make a decision on reader's authenticity.

4 Comparison

In Table 2, we make a comparison between our proposed protocol and others with respect to several properties: the success probabilities of the Mafia fraud and terrorist fraud; mutual authentication (MA); tag privacy; number of message flows in slow phase; real-time tag computation, as well as the pre-computation.

Table 2. Comparison of distance bounding protocols

	Mafia	Terrorist	MA	Privacy	# of Msg Flows	Real-time Tag Comp	Pre-Comp
BC [1]	$(\frac{1}{2})^n$	No	No	No	2	1 commit, 1 signature	No
SP [14]	$(\frac{1}{2})^n$	No	No	No	2	1 commit, 1 MAC, ECC	No
HK [2]	$(\frac{3}{4})^n$	No	No	No	2	1 PRF	No
MP [7]	$(\frac{1}{2})^n$	No	Partial	No	3	2 Hash	No
KA [11]	$(\frac{1}{2})^n$	No	Partial	No	2	1 PRF	No
Swiss-Knife [3]	$(\frac{1}{2})^n$	$(\frac{3}{4})^n$	Yes	Yes	4	3 PRF	Partial
YZW [15]	$(\frac{3}{4})^n$	No	Yes	Yes	2	2 PRF	No
Our Protocol	$(\frac{3}{4})^n$	$(\frac{3}{4})^n/(\frac{7}{8})^{n\dagger}$	Yes	Yes	2	0 PRF	Yes

† The success probability for the terrorist fraud depends on how many registers the adversary obtains. There are three situations discussed in Section 3.

As we can see from the table, most protocols achieve the success probability of $\left(\frac{1}{2}\right)^n$ for the Mafia fraud resistance except HK's [2] and YZW's [15] since the absence of the signature in the last slow phase. It might be high risk but more efficient. MP [7] and KA [11] used mixed challenges in the fast phase that could converge toward the expected probability of $\left(\frac{1}{2}\right)^n$. Our proposed protocol, which does not have a signature, has two strategies that yield three different success probabilities by using two-bit challenges in the fast phase.

Speaking of the terrorist fraud attack, only the Swiss-Knife [3] and ours which are secure against it. However, [15] is not secure against terrorist fraud when considering the attack in [18]. Regarding to the distance hijacking attack [6], it seems that most protocols are secure in the single protocol environment with an ideal probability except Brands and Chaum's [1].

Next we consider mutual authentication (MA) of the distance bounding protocols. Most protocols assume that the reader should be honest, but this may

not be the case when considering the Mafia fraud attack. Among all previous protocols in Table 2, only the Swiss-Knife [3] and YZW [15] are mutually authenticated that is achieved in our proposed protocol as well. For MP [7] and KA [11], their protocols are based on (binary) mixed challenges that enable partially mutual authentication during the fast phase.

When referring to the privacy of the tag, the Swiss-Knife [3], YZW [15] and ours support tag privacy protection. Our protocol realizes the privacy by using an index on tag's ID in an anonymous way. Note that our protocol requires only O(1) complexity for achieving privacy in the sense that the reader's cost is O(1) PRF, rather than O(n) PRF (For example, in Swiss-Knife [3]) in order to protect tag's privacy. Our protocol prevents the de-synchronization attack with the presence of both TID and TID'. When launching a de-synchronization attack, the adversary either prevents the tag updating ID' or prevents the reader updating TID and TID'. It is obvious that no matter what the adversary does, the value of ID' sent by the tag will always be the same as either TID or TID' so that the tag is synchronized with the reader, and vice versa.

The number of message flows in the slow phase is essential to the protocol execution time and power consumption. As for most protocols including ours have only one single slow phase when compared with the Swiss-Knife which needs four message flows in two slow phases. It is susceptible to much power consumption for a low-cost tag.

Finally, we make a special comparison in terms of the real-time tag computation and pre-computation. As all previous proposed protocols do not explicitly have the pre-computation stage, their protocols must have at least one time-consuming PRF, hash or signature evaluation in the real-time stage. But for the Swiss-Knife [3], they state that, one of three PRFs can be pre-computed before starting the protocol in the sense that the contents of the input for this PRF are irrelevant to the reader. It means that they still need two computations of PRFs in real time for achieving mutual authentication. Our proposed protocol, however, let the tag finish two PRF computations in the pre-computation stage by using a large capacitor which makes the real-time stage extremely faster than any of previous protocols.

5 Conclusion

In this paper, we proposed a highly efficient pre-computed RFID distance bounding protocol with tag privacy. It makes use of a large capacitor to store the DC voltage which can power the tag in order to compute the PRF off-line. Our protocol is mutually authenticated and secure against all common attacks in distance bounding. To the best of our knowledge, our proposed protocol is the first one that provides online PRF-free for the tag meaning that there is no evaluation on any PRF during the real-time protocol running which significantly makes the tag more efficient and low-cost. We also take tag's privacy into account through the method of index to search tag's ID which requires only O(1) complexity for achieving privacy. We give the detailed security analysis for our protocol and make a comprehensive comparison against others.

References

1. Brands, S., Chaum, D.: Distance Bounding Protocols. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 344–359. Springer, Heidelberg (1994)
2. Hancke, G., Kuhn, M.: An RFID Distance Bounding Protocol. In: SecureComm 2005, pp. 67–73. IEEE Computer Society (2005)
3. Kim, C.H., Avoine, G., Koeune, F., Standaert, F.-X., Pereira, O.: The Swiss-Knife RFID Distance Bounding Protocol. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 98–115. Springer, Heidelberg (2009)
4. Desmedt, Y.: Major security problems with the 'unforgeable' (Feige)- Fiat- Shamir proofs of identify and how to overcome them. In: SecuriCom 1988, pp. 15–17 (1988)
5. Avoine, G., Lauradoux, C., Marin, B.: How Secret-sharing can Defeat Terrorist Fraud. In: ACM Wisec 2011, pp. 145–156. ACM SIGSAC (2011)
6. Cremers, C., Rasmussen, K.B., Čapkun, S.: Distance Hijacking Attacks on Distance Bounding Protocols. In: IEEE S&P 2012, pp. 113–127 (2012)
7. Munilla, J., Peinado, A.: Distance Bounding Protocols for RFID Enhanced by Using Void-challenges and Analysis in Noisy Channels. Wireless Communications & Mobile Computing 8(9), 1227–1232 (2008)
8. Kardaş, S., Kiraz, M.S., Bingöl, M.A., Demirci, H.: A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 78–93. Springer, Heidelberg (2012)
9. Avoine, G., Tchamkerten, A.: An Efficient Distance Bounding RFID Authentication Protocol: Balancing False-Acceptance Rate and Memory Requirement. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.) ISC 2009. LNCS, vol. 5735, pp. 250–261. Springer, Heidelberg (2009)
10. Tu, Y.J., Piramuthu, S.: RFID Distance Bounding Protocols. In: EURASIP Workshop in RFID Technology, Vienna, Austria (2007)
11. Kim, C.H., Avoine, G.: RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 119–133. Springer, Heidelberg (2009)
12. Chae, H.J., Yeager, D.J., Smith, J.R., Fu, K.: Maximalist Cryptography and Computation on the WISP UHF RFID Tag. In: RFIDSec (2007)
13. Hofferek, G., Wolkerstorfer, J.: Coupon Recalculation for the GPS Authentication Scheme. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 162–175. Springer, Heidelberg (2008)
14. Singelée, D., Preneel, B.: Distance Bounding in Noisy Environments. In: Stajano, F., Meadows, C., Capkun, S., Moore, T. (eds.) ESAS 2007. LNCS, vol. 4572, pp. 101–115. Springer, Heidelberg (2007)
15. Yang, A., Zhuang, Y., Wong, D.S.: An Efficient Single-Slow-Phase Mutually Authenticated RFID Ddistance Bounding Protocol with Tag Privacy. In: Chim, T.W., Yuen, T.H. (eds.) ICICS 2012. LNCS, vol. 7618, pp. 285–292. Springer, Heidelberg (2012)
16. Nikov, V., Vauclair, M.: Yet Another Secure Distacne-Bounding Protocol. IACR ePrint Archive, `http://eprint.iacr.org/2008/319` and SECRYPT 2008
17. Rasmussen, K.B., Čapkun, S.: Location Privacy of Distance Bounding Protocols. In: ACM CCS 2008, pp. 149–160. ACM SIGSAC (2008)
18. Fischlin, M., Onete, C.: Provably Secure Distance-Bounding: an Analysis of Prominent Protocols. In: IACR ePrint Archive (2012), `http://eprint.iacr.org/2012/128`