4-2020

Energy-efficient distance-bounding with residual charge computation

Yunhui ZHUANG

Anjia YANG

Gerhard HANCKE

Duncan S. WONG

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

IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 1 June 2017; revised 31 August 2017; accepted 23 September 2017.
Date of publication 13 October 2017; date of current version 9 June 2020.

Digital Object Identifier 10.1109/TETC.2017.2761702

Energy-Efficient Distance-Bounding with Residual Charge Computation

YUNHUI ZHUANG, ANJIA YANG, GERHARD P. HANCKE, (Senior Member, IEEE), DUNCAN S. WONG, (Senior Member, IEEE), AND GUOMIN YANG, (Member, IEEE)

Y. Zhuang is with the Department of Information Systems, College of Business, City University of Hong Kong, Kowloon Tong, Hong Kong
A. Yang is with the College of Cybersecurity/College of Information Science and Technology, Jinan University, Guangzhou 510632, China
G.P. Hancke is with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong
D.S. Wong is with the CryptoBLK Limited, Hong Kong
G. Yang is with the University of Wollongong, NSW 2522, Australia
CORRESPONDING AUTHOR: A. YANG (anjiayang@gmail.com)

ABSTRACT Real-time location systems are often required in industrial applications. In addition to securely determining an item's location, these systems also need to accommodate energy-limited tracking tokens. Distance-bounding protocols enable a Verifier to cryptographically determine an upper-bound on the physical distance to a Prover by measuring the round-trip time of specially designed challenge-response messages. This type of protocols serve as countermeasure to three common attacks on location-based systems and have been extensively studied with the goal of achieving optimal security bounds for the respective attacks. In this paper, we propose a new energy-efficient distance-bounding protocol that protects against all three common attacks in a distance-bounding scenario with improved security bounds. We provide a new approach to combining the response registers and Prover's key to determine responses. Furthermore, the protocol design allows offline pre-computation of the function f used to determine the Prover's response registers. This results in faster protocol execution, the reader does not wait for the tag to compute any cryptographic function during the protocol execution, and also allows passive tokens to effectively use residual energy after the preceding transaction to compute response registers for the next protocol run.

INDEX TERMS Distance bounding, embedded hardware, RFID security

I. INTRODUCTION

Reliably tracking the location of items is an important service in some industrial settings. Radio Frequency Identification (RFID) is a prominent technology often used for identification and real-time localisation of items. RFID systems use low-cost, low-power and multi-functional embedded tags, which are activated and powered by the RFID reader's communication. RFID technology is used in numerous consumer and industrial applications. It is used extensively in supply chain systems and for item tracking. For example, in the medical industry, local hospitals use RFID systems to enable the tracking of in-patients and medical equipments, and the routine delivery of drugs and specimens simultaneously, and in the aviation industry, some major airports, such as Hong Kong, Dubai, Las Vegas, have adopted innovative RFID-enabled baggage handling systems to improve sorting and tracking efficiency.

Some of these RFID systems are operating in potentially security sensitive applications. Apart from the importance of tracking high value items, other security-sensitive system include e-passports and both card and mobile-based contactless payments. The provision of basic security services, such as data confidentiality, data integrity, and authentication, is therefore an important requirement in such systems. In addition to basic security, secure RFID and ad-hoc wireless communications in general also need to provide assurance as to the physical proximity of the communicating devices [1], [2]. However, any security mechanisms implementing these services must be designed keeping in mind the constrained nature of the embedded tags. RFID systems, particularly supply chain and real-time location systems, rely heavily on the notion of device proximity [3], [4]. The operational range of typical RFID tags used in such applications is known to be between 10 cm (high-frequency (HF) tags) to 10 m (ultrahigh-frequency (UHF) devices). Intuitively, if a reader is able to communicate with a tag, then the tag must be in close physical proximity to the location of the reader. However, if a

malicious tag replaced it with an radio transceiver that simply relays the messages from an RFID reader to the real tag and then forwards the real tag's response back to the reader, the reader will still consider the tag to be in close proximity, as there is still an entity that appears exactly the same as the real tag from a communication perspective. Therefore, the secure verification of an RFID tag's physical location relative to the reader is crucial to the secure and reliable operation of RFID-enabled real-time location applications. The distance-bounding protocol, discussed in Section II, is a prominent approach to verifying tag proximity in RFID systems, and enables an RFID reader to securely compute an upper bound on the physical distance between it and a tag based on the round-trip time of multiple cryptographic challenge-response exchanges.

In this paper, we propose a new distance-bounding protocol with the following properties:

(1) Our protocol allows for offline (when not communicating with the reader) pre-computation of function f used to determine the tag's response registers. This results in faster online protocol execution, conserving reader energy by decreasing the time that the reader transmits an RF carrier, as the reader is not waiting for the tag to perform a significant cryptographic computation during the protocol run. The tag only needs to make a few if-else decisions in online during the exchange stage.
(2) Offline pre-computation also allows for an energy-efficient implementation where the tag effectively uses residual energy accumulated during the protocol run to compute the response registers for the next protocol run. In Section VI we demonstrate that this is practical with a proof-of-concept implementation on a passive UHF computational RFID device. This also allows for faster execution, which means more tags can be authenticated in a shorter time. This is useful in logistics systems as the tags can pass through faster.
(3) Our protocol remains resistant to three common frauds in a distance-bounding scenario with security bounds of $\left(\frac{5}{8}\right)^n$ for distance fraud, $\left(\frac{n}{2}+1\right)\left(\frac{1}{2}\right)^n$ for both Mafia and terrorist frauds. The design of the response function builds on the ideas of [5]'s exclusive or response, [6]'s response register, and [7]'s responses with memory. This combination produces a new approach to response calculation that allows offline computation of the response register and key-dependent responses. Our protocol is described and analysed in Sections III, IV, and V.

II. DISTANCE-BOUNDING PROTOCOLS

Desmedt [8] in 1987, introduced the idea of 'Mafia fraud' that could defeat any authentication protocol. An adversary using this approach in an RFID context can defeat any protocol by simply relaying messages between the legitimate reader and a remote legitimate tag. Distance-bounding protocols are a family of challenge-response authentication protocols, first introduced by Brands and Chaum [5] in 1993, which mitigate this attack. They essentially allow an RFID reader to both authenticate a tag and verify that latter's physical proximity. These protocols are of particular interest in contactless card systems, e.g., electronic payment or access control systems. The protocol allows the reader to establish an upper bound on the physical distance to a tag based on the round-trip time of n cryptographic challenge-response exchanges. In 2005, Hancke and Kuhn (HK) [6] designed a distance-bounding protocol, targeted at resource-constrained RFID technology, using a simple pseudo-random function and not requiring a final cryptographic verification of all the challenge-response exchanges. These two protocols have since served as key-references in numerous RFID distance-bounding proposals, e.g., [9]–[17].

A. WHY WE NEED DISTANCE BOUNDING IN INDUSTRY APPLICATIONS

As stated in [18], the verification of a tag's physical proximity to a reader is crucial to the secure and reliable operation of industrial RFID and real-time location system (RTLS) applications. The security of RTLS and location-based services in general has been subject to increased scrutiny. For instance, an attacker can relay HF RFID tag communication using off-the-shelf NFC-enabled mobile phones, by configuring one as a reader and the other as a card [19]–[22], to make the tag appear in a different location. It is also possible for an attacker to spoof her location in other location systems, such as wireless local area networks (WLAN). If an RTLS is used to continually track items of high value within an industrial setting, such as expensive parts in a supply chain management or logistics of valuable shipments, the required level of security could justify the use of distance bounding. Although this paper focuses on distance bounding in the context of RFID, these protocols could be applied to any ad-hoc wireless communication for verifying next-hop neighbours or ranging measurements in alternative RTLS implementations using, for example, wireless sensor networks or Ultra-Wideband (UWB). These systems can also accommodate the design for energy-saving [23], [24].

B. SECURITY MODEL DEFINITIONS

Distance-bounding protocols aim to detect three main attack scenarios, namely Distance fraud [5], Mafia fraud [8], and Terrorist fraud [8], and distance-bounding protocols are usually evaluated in terms of the attacker's success probability in executing each of these attacks. Formal models and analysis for distance-bounding is still a developing discipline, but to be consistent with related work we have adopted the distance-bounding analysis framework by Avoine *et al.* [25]. This framework presents a complete model of all three main attack scenarios in both a "black-box" and "white-box" environment, i.e., whether the Prover has no control, or full control, over the protocol execution. We formalize our security model, i.e., the attack scenarios our protocol is trying to prevent, primarily using the definitions in [25].

Definition 1 (Distance-Bounding Protocol): A distance-bounding protocol is a combination process of both authentication and distance checking. The former ensures the soundness and correctness of a given protocol, and the latter verifies an upper bound on the distance between two parties.

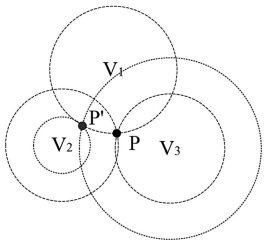


FIGURE 1. An example of distance fraud in RTLS [18].

Definition 2 (Impersonation Fraud): An impersonation fraud is an attack where a malicious Prover pretends to be another.

Definition 3 (Distance Fraud): A distance fraud is an attack where a dishonest and lonely Prover convinces the Verifier that she is closer than is really the case.

A practical example of how this attack could affect an RTLS is shown in Figure 1. In this example, three verifiers V_1, V_2, and V_3 need to determine the location of a tag P based on the received signal strength of their respective communication with P. A malicious tag P could easily pretend to be at position P' by attenuating its signal when communicating with V_2 and increasing the signal when communicating with V_3. Moreover, encrypting data or authenticating P do not prevent this fraud, as P is seen as a "legitimate tag" in the network and in possession of the secret information to both correctly encrypt/decrypt data and thus it can authenticate itself to others. However, a distance-bounding protocol detects whether a malicious tag is pretending to be closer to the reader than it really is. If the localization system used distance bounding, P would not be able to effectively spoof a chosen location as it cannot pretend to be closer to V_2.

Definition 4 (Mafia Fraud): A mafia fraud is an attack where an adversary defeats a distance-bounding protocol using a man-in-the-middle (MITM) between the Verifier and an honest Prover located outside the acceptable distance.

Practical Mafia fraud has been successfully demonstrated in real-world contactless/RFID applications with good conventional security mechanisms, such as keyless entry and payment systems. This has security implications for industrial supply chain and RTLS based RFID technology, as illustrated in Figure 2. An attacker can convince a reader that a tagged item is closer by using two relaying proxies, even though this item has been removed and is actually far away.

Definition 5 (Terrorist Fraud): A terrorist fraud is a variant of distance fraud, in which the Prover colludes with an adversary who uses man-in-the-middle to deceive the Verifier on her distance. A terrorist fraud is an attack where an adversary defeats a distance bounding protocol using a man-in-the-middle between the Verifier and a dishonest Prover located outside the acceptable distance. The Prover helps the adversary to maximize her attack success probability in a single protocol execution, without giving her any further advantage for future attacks.

In practice, this involves the Prover sharing protocol information, other than key material, with a third-party in such a way that it allows this third-party to convince the reader that it is the legitimate tag without having to relay all the reader's messages. The attack fails if the Prover has to reveal any portion of its secret key and increase the effort of the attacker to impersonate the tag in current and subsequent protocol executions. This attack is more theoretically relevant, and arguably has limited practical usage in the context of industrial RFID and RTLS applications as an inanimate object is not likely to share selected information with an adversary.

We then define a realistic and fair model for adversary's capabilities. We consider in our protocol a Dolev-Yao adversary [26] and define her generic capabilities and possible actions.

Definition 6 (Dolev-Yao Adversary): A Dolev-Yao adversary is an "active" eavesdropper who can eavesdrop, intercept, and synthesis messages transmitted over the communication channel. But she cannot perform unbounded computations and cannot obtain the secret keys from any honest parties.

It is also defined that when an adversary executes the protocol several times, it does not increase her success probability in future executions. Thus, security analysis can consider only one protocol execution.

Definition 7 (Definition of Function f): The function f has the following properties:

(1) Given any output value y, it is computationally infeasible to find any input message x such that $y = f(x)$.

(2) Given an input message x, it is computationally infeasible to find another message x' such that $f(x) = f(x')$.

There are several standard cryptographic primitives that meet these requirements for function f. For example, one-way collision-resistant hash function or block cipher. In Section VI, we will use AES with a block size of 128 bits and a key size of 128 bits as a building block in the proof-of-concept experiment for realising f.

III. OUR PROTOCOL

In this section, we briefly define the RFID system, and then present our protocol design in Section III-A. Section III-B discusses a working example and some concerns to our proposed protocol.

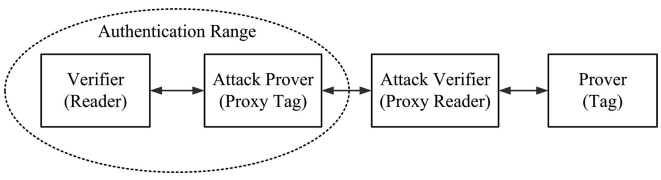


FIGURE 2. An example of mafia/terrorist fraud in RTLS.

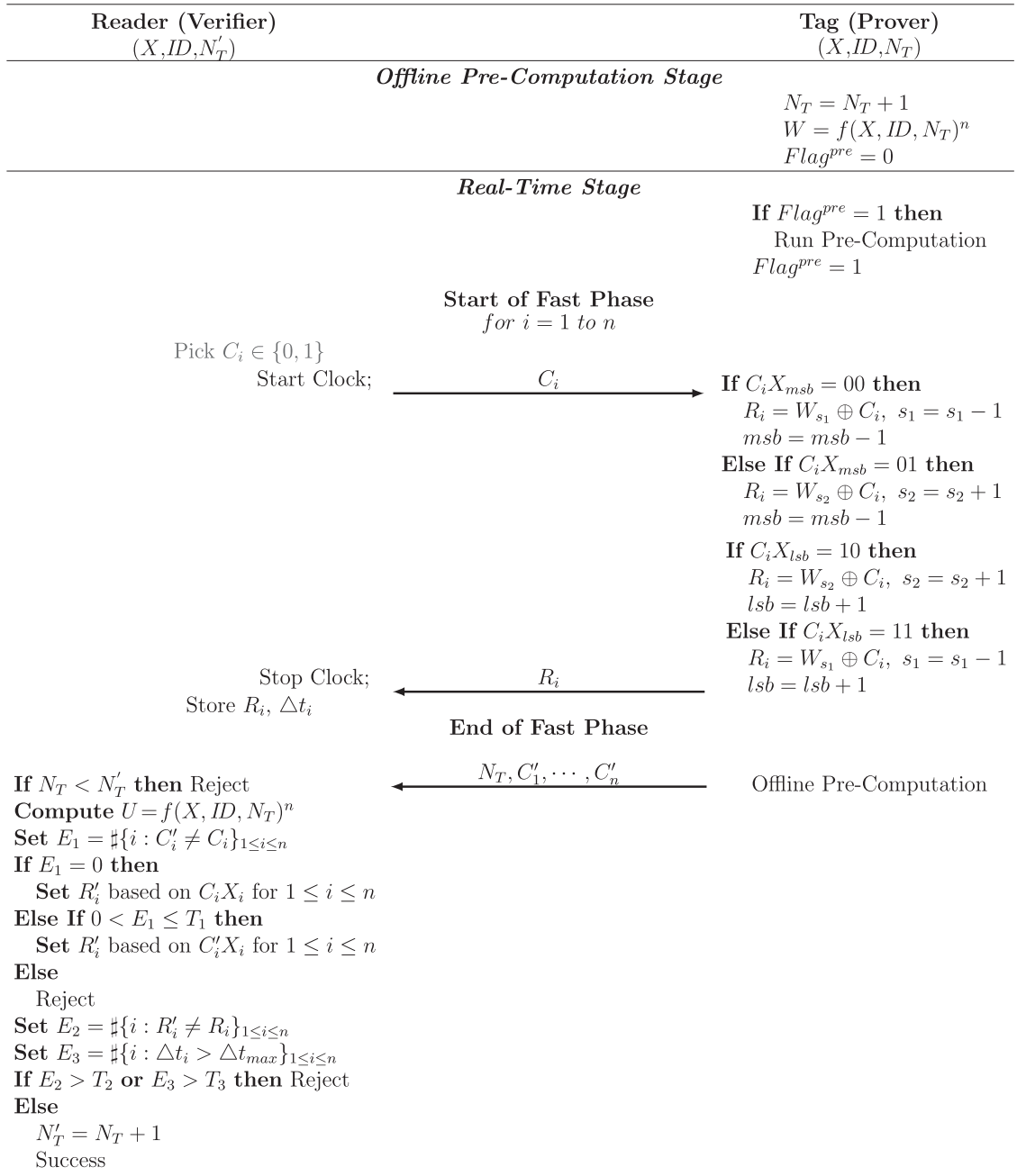


FIGURE 3. Our protocol proposal.

A. SYSTEM AND PROTOCOL DESCRIPTION

The RFID system consists of a reader associated with a back-end server, and a set of tags. Each tag stores a secret key X_i, which is shared with the reader. The reader maintains the tag's identity ID and a counter N_T' as well. The communication after the end of the fast phase is assumed to be over an error-corrected channel. The speed of propagation is assumed to be the speed of light. The protocol consists of one offline pre-computation stage and one online stage as shown in Figure 3. The length for challenge C, register W, and secret key X are all n bits.

Offline Pre-Computation Stage. Our protocol applies a counter N_T (initialized to 0 and incremented by one when the tag is powered up) as one of the three inputs to compute f. This prevents the possibility of response replay attacks as the resulting W is different for each protocol run. This design choice means the input for computing f is independent from the reader. The tag can therefore compute f at any stage before the protocol actually starts. The tag computes $W = f(X, ID, N_T)^n$ and stores W into one register that carries n bits. As shown in Table 1, the tag sets *msb*/*lsb* to the corresponding bit of the secret key X and s_1/s_2 to the corresponding bit of the register W.

There is one flag bit to facilitate the steps during the offline pre-computation stage: $Flag^{pre}$, which ensures the pre-computation is done successfully by updating the counter N_T and whole share of register W.

TABLE 1. Notations.

X / ID	:	tag's secret key shared with the reader / tag's real identity
n	:	number of rounds for challenge-response exchanges in the Fast Phase of *Real-Time Stage*, or number of bits.
N_T / N'_T	:	the counter initialized to zero
$Flag^{pre}$	:	flag bit indicates whether the pre-computation has been done successfully
f	:	the underlying cryptographic function f, see Definition 7
W / U	:	the register calculated by both parties through f: $W/U = f(X, ID, N_T)^n$
C_i / R_i	:	reader's challenge bit / tag's response bit
msb / lsb	:	cursor that points to the most / least significant bit of secret key X, initialized as $msb \leftarrow n$, $lsb \leftarrow 1$
s_1 / s_2	:	cursor that points to the most / least significant bit of register W, initialized as $s_1 \leftarrow n$, $s_2 \leftarrow 1$
$\triangle t_i$	:	time difference between the challenge sent and response received by the reader
$\triangle t_{max}$	:	maximum time difference allowed in one round
E_1, E_2, E_3	:	number of errors in challenge C_i, response R_i, and time difference, respectively
T_1, T_2, T_3	:	threshold value

Real-Time Stage. The online stage has one *fast bit exchange phase*, which has a total of n rounds, and one *slow phase* during which additional data is exchanged to facilitate fault tolerance during the fast bit exchange. The stage requires no cryptographic calculation to be made by the tag and the tag only needs to make some if-else decisions during the fast phase. Before executing the fast phase, the tag needs to check the status of the flag bit again to prevent replay attack (details in Section III-B). Therefore, the tag needs to perform the pre-computation in real time once if $Flag^{pre} = 1$.

The protocol now moves to the fast bit exchange phase:

(1) The reader randomly picks a challenge bit C_i, starts the clock and sends it to the tag.
(2) The tag sends back corresponding response bit R_i based on the received C_i.
(3) Upon receiving R_i, the reader immediately stops the clock, records the round-trip time (RTT) $\triangle t_i$, and R_i.
(4) Above three steps are repeated for n rounds.

During each round, the response bit R_i is dependent on the challenge bit C_i received and corresponding bit of the secret key X. The selection process is as follows:

(1) If $C_i = 0$, msb is activated:
  (1.1) If $X_{msb} = 0$, then the cursor s_1 is activated and W_i is XORed with the received challenge C_i. The result is sent back and s_1 is decremented by one.
  (1.2) Otherwise, the cursor s_2 is activated and W_i is XORed with the received challenge C_i. The result is sent back and s_2 is incremented by one.
  (1.3) msb is decremented by one.
(2) If $C_i = 1$, lsb is activated:
  (2.1) If $X_{lsb} = 0$, then the cursor s_2 is activated and W_i is XORed with the received challenge C_i. The result is sent back and s_2 is incremented by one.
  (2.2) Otherwise, the cursor s_1 is activated and W_i is XORed with the received challenge C_i. Then the result is sent back and s_1 is decremented by one.
  (2.3) lsb is incremented by one.

No round-trip time is measured in the final verification phase:

(1) The tag sends the counter N_T, along with a set of received challenges $C'_1, \ldots, C'_n$, to the reader. Note that N_T is always updated during the pre-computation no matter what decision the tag made. The tag then performs the pre-computation using its residual stored energy to update W for next round protocol execution.
(2) There are two parts during the checking mechanism by means of several if-else decision makings:

Counter Checking. The reader checks if $N_T < N'_T$ to prevent replay attack, and rejects the tag if it holds. Otherwise, the reader's counter N'_T will be updated as $N'_T = N_T + 1$ after it successfully verifies the tag.

Fault Tolerance. The reader computes its version of register, $U = f(X, ID, N_T)^n$, and set corresponding R' based on the challenges it picked during the fast phase to facilitate the fault tolerance by checking on the validity of E_1, E_2, and E_3, as shown in Table 2. We also denote T_1, T_2, and T_3 as the fault tolerance thresholds.

B. FURTHER DISCUSSION

A Working Example. To explain how the response function work we provide a simple working example. Let register W and secret key X both be 6-bit as shown in Table 3.

Figure 4 gives a detailed example of how the response function is executed, where the first four challenges are $C_1 = 0, C_2 = 1, C_3 = 1, C_4 = 0$. In particular, the response function has 'memory' in the sense that the values of respective cursors are depended on the previous challenge.

The Counter. Intuitively, the counter N'_T stored in the reader should be synchronized with the counter N_T stored in the tag. However, an adversary could try to execute a de-synchronization attack to deliberately increase the reader's N'_T.

TABLE 2. Errors and thresholds.

E_1 : counts the number of errors of positions for the challenges $C'_i \neq C_i$.
E_2 : counts the number of errors of positions for the responses $R'_i \neq R_i$.
E_3 : counts the number of errors of the transmission delay $\triangle t_i > \triangle t_{max}$.
If $E_1 = 0$: the reader sets the corresponding R_i according to both $C_i X_i$.
If $E_1 \leq T_1$: the reader sets R'_i based on received C'_i and X_i.
If $E_1 > T_1$ or $E_2 > T_2$ or $E_3 > T_3$: the reader rejects the tag.

TABLE 3. A Working example (6-Bit).

(a) Register W and Its Cursors						
	s_1 $\rightarrow$					s_2 $\leftarrow$
W	1	0	1	0	1	1
(b) Secret Key X and Its Cursors						
	msb $\rightarrow$					*lsb* $\leftarrow$
X	0	1	0	1	0	0

If the adversary succeeds, N'_T will be larger than the tag's counter N_T and any legitimate responses from the tag will therefore be dismissed as replay attempts. Our protocol ensures that the reader updates its counter N'_T only if the protocol has successfully bounded the tag involved, so an adversary would not be able to increase N'_T unless she has full knowledge of the tag's secret key X.

The Flag Bit. An RFID tag can lose power at any time, i.e., move away from the reader, thus it might be possible to force a tag to re-use pre-computed values more than once. To mitigate this we use one flag bit $Flag^{pre}$ to ensure the integrity of the RFID environment and to make sure the pre-computation has been successfully done before going into online stage. If $Flag^{pre} = 1$ before starting the fast phase, the tag is aware that either insufficient residual energy was stored so that the tag could not perform the pre-computation, or some sort of attacks have been launched. To recover from either problem the tag only needs to perform the pre-computation during the online stage once if $Flag^{pre} = 1$.

IV. SECURITY ANALYSIS

The analysis provided here follows the framework introduced in [25], as per the protocol and main attack definitions given in Section II-B. For each of the attacks we consider a "white box" environment, which is the more powerful attack model where the adversary has control over the execution of the protocol. We perform the analysis in the noise-free case ($T_1 = T_2 = 0$) and then briefly comment on the noisy case.

A. ANALYSIS IN THE NOISE-FREE CASE

Impersonation Fraud Resistance. The adversary tries to impersonate another tag to deceive the reader. To succeed she must correctly respond the challenges during the fast phase.

Claim 1: An impersonation fraud adversary has a success probability of $\left(\frac{1}{2}\right)^n$ to defeat the protocol.

Discussion 1: The adversary does not have X and she needs to guess the response and get them all correct for the attack to succeed. Let P^i_{rand} be the probability that the adversary replies with the correct response in round i, for $i \in \{1, \ldots, n\}$. The success probability of the impersonation attack is therefore

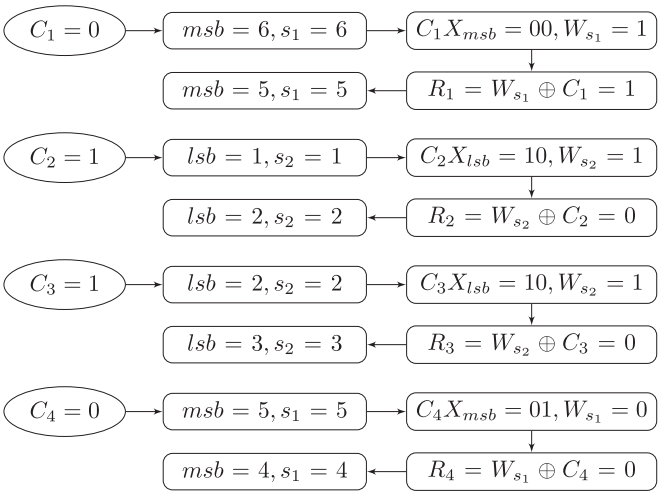


FIGURE 4. Simulation of first four rounds.

$$P_{Imp} = \prod_{i=1}^{n} P^i_{rand} = \left(\frac{1}{2}\right)^n.$$

Distance Fraud Resistance. The adversary is the tag herself and she needs to carry out an early-reply strategy to ensure the RTT is within the threshold $\triangle t_{max}$, which means during the fast phase, she has to send a response R_i before receiving the challenge C_i in order to accomplish the attack. It is important to note that the attack on Hamming distance [25] does not apply to our scheme in the sense that the choice of response bit in our scheme is cursor dependent, not sequential.

Claim 2: Given three strategies in a distance fraud, the adversary's success probability is upper-bounded by $\left(\frac{5}{8}\right)^n$.

Discussion 2: There are three strategies to accommodate a distance fraud attack:

Strategy 1: Reply Randomly. Regardless of the reader's challenges, the adversary replies with random bit in each round. Recall that P^i_{rand} is the probability that the adversary replies with the correct response in each round. Then the success probability under this strategy is calculated as

$$P_{dist1} = \prod_{i=1}^{n} P^i_{rand} = \left(\frac{1}{2}\right)^n.$$

Strategy 2: Bit Dependent. The adversary observes the possible response bits in each round based on next bit positions in the registers and tries to increase her probability of success. For example, it has been shown that for [6], where the response is chosen from two registers based on the challenge, that the adversary will know the correct response regardless of the challenge if the value of the potential response bit in both registers are equal. In the proposed protocol if $W^i_{s_1} = W^i_{s_2}$ in one round then the adversary does not gain any advantage since R_i also depends on C_i, which is unknown to her, so the probability of guessing the response correctly is $\frac{1}{2}$. Otherwise, if $W^i_{s_1} \neq W^i_{s_2}$ then the adversary also has to guess the correct response with probability $\frac{1}{2}$. Let P^i_R be the probability that $W_{s_1} = W_{s_2}$ in round i, for $i \in \{1, \ldots, n\}$, we have

$$P_R^i = \frac{1}{2},$$

and let $P^i_{correct}$ be the probability that the adversary replies with the good answer in round i, for $i \in \{1, \ldots, n\}$. Hence the success probability under this strategy is given by

$$P_{dist2} = \prod_{i=1}^{n} P^i_{correct} = \prod_{i=1}^{n} \left(\frac{1}{2} \times P_R^i + \frac{1}{2} \times \left(1 - P_R^i\right) \right)$$
$$= \left(\frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} \right)^n = \left(\frac{1}{2} \right)^n.$$

Strategy 3: Cursor Dependent. The tag takes advantage of arriving challenge bit C_i to correctly update cursors and msb/lsb pointers afterwards. With this strategy, the adversary knows the exact response if $X_{lsb} = X_{msb}$ and $W_{s1} \neq W_{s2}$ in each round. For example, suppose in the first round $X_{lsb} = X_{msb} = 0$ and $W_{s1} = 0$ and $W_{s2} = 1$. If $C_1 = 1$, then the response bit $R_1 = 0$, while if $C_1 = 0$, then the response bit $R_1 = 0$ as well. On the other hand, if $W_{s1} = 1$ and $W_{s2} = 0$, and if $C_1 = 1$, then the response bit $R_1 = 1$, if $C_1 = 0$, then the response bit is still $R_1 = 1$. Let P_x^i be the probability that $X_{lsb} = X_{msb}$ and P_w^i be the probability that $W_{s1} \neq W_{s2}$ in a given round i, for $i \in \{1, \ldots, n\}$, such that

$$P_x^i = P_w^i = \frac{1}{2}.$$

Then the success probability for this strategy is computed as

$$P_{dist3} = \prod_{i=1}^{n} \left(P_x \times P_w \times 1 + (1 - P_x \times P_w) \times \frac{1}{2} \right)$$
$$= \left(\frac{1}{2} \times \frac{1}{2} + \left(1 - \frac{1}{2} \times \frac{1}{2}\right) \times \frac{1}{2} \right)^n$$
$$= \left(\frac{5}{8} \right)^n.$$

Therefore, the adversary would choose *Strategy 3* as it provides the highest success probability with $\left(\frac{5}{8}\right)^n$.

Mafia Fraud Resistance. The adversary does not know any secret information and she needs to "authenticate" herself to the reader on both identity and distance. The honest tag does not collude with the adversary. There are two strategies to accommodate a mafia fraud attack.

Claim 3: Given two strategies in a mafia fraud, the optimal success probability for the adversary is $\left(\frac{n}{2} + 1\right)\left(\frac{1}{2}\right)^n$.

Discussion 3: The adversary launches her attack with one of the following two strategies.

(1) *Post-ask strategy.* The adversary first executes the fast phase with the reader, trying to guess the responses and learning the challenges C. After obtaining all challenges from the reader, the adversary executes the fast phase with the tag by acting as a "fake" reader so that she can obtain further information for the final slow phase, e.g, the counter N_T. Let event D_1^i be when the adversary correctly replies with an arbitrary answer in each round i, where $i \in \{1, \ldots, n\}$. Then her success probability is

$$P_{post\text{-}ask} = \prod_{i=1}^{n} \mathbf{Pr}[D_1^i] = \left(\frac{1}{2} \right)^n.$$

(2) *Pre-ask strategy.* During the execution of fast phase, the adversary cannot relay challenges and responses between the reader and a tag without being detected. With this strategy, the adversary first executes the fast phase with the tag by sending a sequence of challenge bits $\tilde{C}_i$. Next the adversary executes another fast phase with the reader to obtain the genuine challenge bits C_i, and then relays the second slow phase. The analysis for this strategy follows a similar approach as of [7], in which the choice of response bit in each round is depended on the value of previous challenge bit. The adversary's reply at time t must be independent of the reader's challenge at time t (for any $t \in \{1, 2, \ldots, n\}$). Without loss of generality we suppose that the adversary queries the tag with an all 0 sequence, that is, $\tilde{C}_1, \ldots, \tilde{C}_n = 0$. Let $\tilde{R}_t$ denote as the adversary's reply at time t. Thus, the adversary is successful at time t if and only if $\tilde{R}_t = R_t$. Now let k be the first time when $C_t = 1$ ($t \geq 1$), thus the probability of having $\tilde{R}_t = R_t$ is 1 for every $t \in \{1, 2, \ldots, k-1\}$, whilst with probability $\frac{1}{2}$ of having $\tilde{R}_t = R_t$ for every $t \in \{k, k+1, \ldots, n\}$, since the adversary could still randomly guess the response when $C_t = 1$ occurs. Thus, the probability of pre-ask strategy is given by

$$P_{pre\text{-}ask} = \prod_{t=1}^{n} \mathbf{Pr}(\tilde{R}_t = R_t | C_t = 0) \mathbf{Pr}(C_t = 0)$$
$$+ \sum_{t=1}^{n} \mathbf{Pr}(\tilde{R}_n = R_n | k = t) \mathbf{Pr}(k = t)$$
$$= 1^n \cdot \left(\frac{1}{2} \right)^n + \sum_{t=1}^{n} \left[\left(\frac{1}{2} \right)^{(n-(t-1))} \cdot \left(\frac{1}{2} \right)^t \right]$$
$$= \left(\frac{n}{2} + 1 \right) \left(\frac{1}{2} \right)^n.$$

Therefore, the adversary would choose *pre-ask strategy* to maximize her success probability to $\left(\frac{n}{2} + 1\right)\left(\frac{1}{2}\right)^n$.

Terrorist Fraud Resistance. The malicious tag $\mathcal{T}$ colludes with the adversary $\mathcal{A}$ without disclosing any portion of its secret key as defined in our model. $\mathcal{T}$ cannot guide $\mathcal{A}$ on which response to pick from register W for each challenge C since $\mathcal{A}$ would be able to deduce the corresponding bit of secret key X. Thus, $\mathcal{A}$ executes the fast phase and relays the slow phase. $\mathcal{T}$ has to share some secret information, e.g., the entire register W, with $\mathcal{A}$ in advance to help her to defeat the protocol without getting any advantage to further impersonate $\mathcal{T}$.

Claim 4: Given the entire register W, the success probability for a terrorist fraud adversary bounds of $\left(\frac{n}{2} + 1\right)\left(\frac{1}{2}\right)^n$.

Discussion 4: The adversary has entire register W shared by the malicious tag. It is similar to the strategy 2 of the distance fraud in which the adversary would have an advantage to win the first round if $W^1_{s_1} = W^1_{s_2}$, regardless of the challenge bit C_1. Otherwise, she will have to guess the correct R_1 with probability $\frac{1}{2}$. Recall that $P_{correct}$ is the probability that

the adversary replies with the correct response and P_w is the probability that $W_{s_1} \neq W_{s_2}$ in every round i, for $i \in \{1, \ldots, n\}$. The first round success probability is

$$P^1_{correct} = \left(1 \times \left(1 - P^1_w\right) + \frac{1}{2} \times P^1_w\right) = \frac{3}{4}.$$

Since the adversary cannot determine which cursor(s) will be activated from the second round onwards as she does not have X, she can only randomly guess the response even if she has the register W. Recall that event D^i_1 is when the adversary replies correctly with an arbitrary answer in each round i, where $i \in \{1, \ldots, n\}$. Hence the overall success probability of the terrorist fraud is given by

$$P_{Terrorist} = P^1_{correct} \cdot \prod_{i=2}^{n} \mathbf{Pr}[D^i_1] = \frac{3}{4}\left(\frac{1}{2}\right)^{n-1}.$$

However, we notice that the adversary does not gain any advantage in mounting a terrorist fraud above as $P_{Terrorist} < P_{pre\text{-}ask}$ due to restrictions on adversary's capabilities (See Definition 5) as such in the mafia fraud (certainly affects the success probability over mafia fraud). The adversary may stick to the mafia fraud as a Man-in-the-middle with pre-ask strategy to "increase" her success probability to $\left(\frac{n}{2} + 1\right)\left(\frac{1}{2}\right)^n$.

B. ADDITIONAL REMARKS

Alternative View on Terrorist Fraud. Some prior work propose a different definition of the model for terrorist fraud [29], in which partial secret key or related information can be shared with the adversary as long as the entire key is not revealed to the attacker. Considering this case, if a malicious tag in our protocol is willing to share up to T_2 continuous bits of secret key X with the adversary then the overall success probability in this case is increased to $\left(\frac{n-T_2}{2} + 1\right)\left(\frac{1}{2}\right)^{n-T_2}$.

Non-Narrow MiTM Attacks via Return Channel. As pointed out by Bay *et al.* [30], the return channel strongly affects the security of many DB protocols. The adversary could use the fact that the protocol succeeded or not as a potential side channel in key recovery. For certain response functions, the adversary can deduce a bit of the secret key from each protocol run by flipping the challenge sent from the reader during a chosen challenge-response round and observing the outcome of the protocol execution. The type of protocols shown to be vulnerable generally use two response registers where knowledge of both possible responses would lead to knowledge of the corresponding key bit. For example, $R_0 = a$ and $R_1 = a \oplus X$. If an attacker flips the challenge during the ith round and the protocol succeeds she knows that $R^i_0 = R^i_1$, and if it fails that $R^i_0 \neq R^i_1$. This allows him to calculate X^i.

Although we cannot prevent the adversary learning the outcome via side channel, the function f used in our protocol can prevent the adversary learning any bit of the secret key. It does not use a double register design like the effected protocols in [30] and no information on X is disclosed by the responses R_i. The two cursors that point to the msb/lsb of secret key X cannot be restored when the protocol starts.

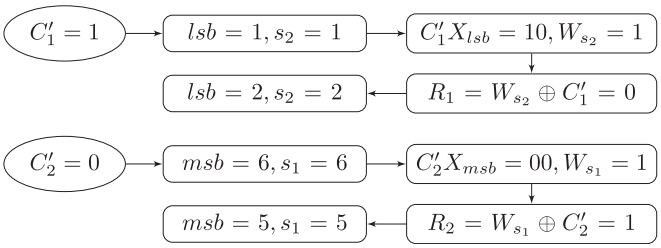


FIGURE 5. Example of first two rounds (challenges flipped).

Either the msb or lsb will stop increasing or decreasing somewhere in the middle. The extreme case is $msb = 1$ (decrease from n to 1) or $lsb = n$ (increase from 1 to n), which means the reader's challenge bits are always 0 or 1. Note that s_1 and s_2 cannot be restored either. We hereby show another working example with the same register and secret key as in Section III-B, except we flip the first two challenges to be $C'_1 = 1$ $(C_1 = 0)$ and $C'_2 = 0$ $(C_2 = 1)$. Figure 5 highlights the difference occurred when flipping first two challenges.

It is clear that a MiTM adversary could learn nothing via return channel because msb/lsb and s_1/s_2 are switched in two rounds. In addition, without the knowledge of register W, the adversary cannot tell the exact value at which the cursors are pointing. Thus, our protocol is secure against such attack.

PRF Assumption. Boureanu *et al.* [31] states that simply specifying f to be a pseudo-random function (PRF) possibly results in insecure protocols. They state that a PRF construction $\mathcal{F}$ where the adversary could program (influence) the result $\mathcal{W}$ is possible. As such, malicious entities can generate an output exhibiting properties of their choice be providing chosen input to the PRF. In such a case, $\mathcal{F}$ meets the requirements set of f but the protocol is not secure.

If an adversary could choose $W = \mathcal{W}$ it could lead to several attacks. However, our specification of f as set out in Definition 7 does not allow for such an adversary. No polynomial time adversary $\mathcal{A}$ is able to choose $\mathcal{W}$ and influence f to provide this value because it is not possible to find input x so that $f(x) = \mathcal{W}$. This property of f is not restrictive on practical implementation as it is provided by standard cryptographic primitives, e.g., hash or encryption. If our protocol is implemented using secure primitives, using algorithms found in security standards, like SHA-3 or AES, our scheme is not effected by the distance fraud and man-in-the-middle attacks in the context of PRF-based attack scenarios presented in [31].

C. VERIFICATION OF OUR THEORETICAL ANALYSIS

To verify our theoretical security analysis, we simulated the distance fraud, mafia fraud and terrorist fraud in Matlab. Both the theoretical and experimental values are depicted in Figure 6. The success probability is shown on the y-axis on a logarithmic scale for the sake of easy reading. The simulation results correspond well with those of the theoretical analysis.

D. COMMENT ON NOISE RESISTANCE

In order to facilitate state of the art powerful devices which are less error-prone, our protocol supports fault tolerance

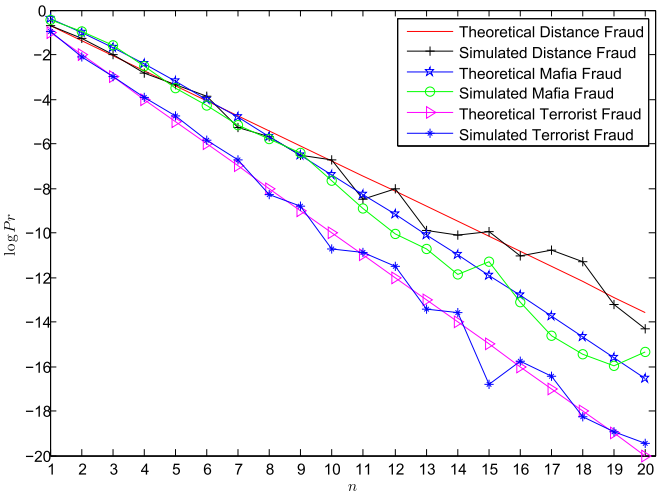


FIGURE 6. Verification of our theoretical analysis.

occurring during the fast phase. Since the communication channel can suffer from unexpected delay, or if some of the challenge-response bits get corrupted, authentication can succeed anyway, provided that the number of errors fall within a reasonable threshold τ (τ can be either T_1, T_2, or T_3, see Table 2). So the success probability in a noise environment follows a Binomial distribution, and the reader only allows at most $n - \tau$ errors (incorrect or delayed responses). Let p be the probability of a correct response in round i, we have the probability p of at least τ responses are correct as

$$Pr(n, \tau, p) = \sum_{i=\tau}^{n} \binom{n}{i} p^i (1-p)^{n-i},$$

where n is the number of rounds in the fast bit exchange phase. Thus, the success probabilities against three frauds with noise are subject to the fault tolerance threshold τ. The probability p can be replaced by either one of previous analyzed frauds.

V. COMPARISON

It is difficult to compare distance-bounding protocols like-for-like, given that different proposals might take into account different operational and threat scenarios. For example, one proposal's goal might be to minimise communicated data, while another wishes to achieve an optimal security bound for only two of the three attack scenarios. We can therefore not claim that our proposal is best for all applications, but in this section we highlight some of the positive features, and attempt to place it in context to related work. Table 4 compares protocols performance with reference to the three common frauds, needing a final signature, complexity of computation during the online stage, the option of offline computation, memory requirements and data transmitted. Furthermore, we assumed channels that are noise-free in this table. These parameters give a good indication of distance-bounding requirements, including security bounds, resource cost (memory) and execution time (computation and transmission). We follow the approach in [18] to provide values for bits stored, in brackets, and transmitted using typical values for essential elements as shown in the Table footnotes. This approach involves some measure of normalisation across different protocols for purpose of comparison, such as assuming the f is AES 128 as in Section VI and taking into account that n depends on how F is divided to create response registers, which differs between protocols (possibilities are $n = 128 = \mathrm{F}$, $n = 64 = \mathrm{F}/2$, $n = 42 \approx \mathrm{F}/3$ or $n = 32 = F/4$), so values given are indicative and not absolute.

Our proposal performs well in terms of security bounds, offering the lowest bound for terrorist fraud overall and the lowest bound for distance fraud of the protocols that consider terrorist fraud. Protocols 1 and 9 has lower mafia fraud bounds, but 1 does not consider terrorist fraud, and 9 has higher bounds for both distance and terrorist fraud while also requiring 2 PRF operations. Protocol 11 has three further variants $\mathrm{SKI}_{\mathrm{shamir,4,lite}}$ with distance fraud bounds of $(\frac{5}{8}; \frac{3}{4}; \frac{3}{4})$ and mafia fraud bounds of $(\frac{1}{2}; \frac{5}{8}; \frac{3}{4})$ but we list $\mathrm{SKI}_{\mathrm{PRO}}$ as it is the only variant explicitly stated to be terrorist resistant. It should also be noted that the terrorist fraud bounds for protocols 10 and 11 are not proportional to n but to m, an integer value that is smaller than n.

Our proposal requires no final verification signature, just a plaintext transmission after the exchange phase, equivalent to other proposals' nonce exchanges prior to the exchange phase. Our proposal also does not perform any online hash or AES computation. This is the same as Protocol 6, but our proposal has much lower security bounds for all three attack scenarios. This protocol also takes the unconventional approach to transmit two challenge bits in each round, while our protocol uses conventional single-bit exchanges as specified in the general properties of secure distance bounding. Our protocol is towards the lower end in terms of memory stored and the middle range for data transmission.

VI. PRACTICAL BENEFITS AND PROOF-OF-CONCEPT

One of the key contributions of the proposed protocol in terms of practical performance is the reduction of execution time from the reader perspective. The reader is only required during the fast stage and the final verification stage, and never has to wait for the tag to complete a significant cryptographic computation. The obvious benefit is the increase of the reader's read rate, which means it can run the protocol with more tags in the same amount of time or it can interact with tags moving quickly in and out of its read range. For example, this is potentially beneficial in a online location system tracking multiple containers in a passing vehicle. Another benefit is that it allows for the reader to behave in a more energy efficient manner, i.e., the reader can conserve power by reducing the time it transmits the RF carrier to power the tag, which is potentially a useful property for mobile RFID readers[32]. In Figure 7(a), the reader needs to transfer power to the tag for time t_R, which is the time taken for the computation of function f and the challenge-response exchange. In our protocol as shown in Figure 7(b), the reader only needs to supply power for t'_R during the exchange, with the function f being

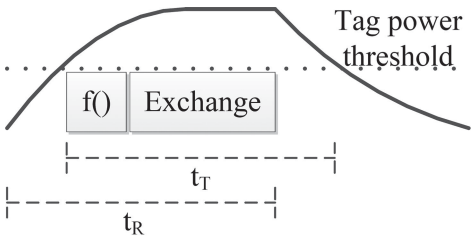
TABLE 4. Comparison to selected existing protocols.

Protocol	Distance	Mafia	Terrorist	Final Signature	Online Complexity	Offline Pre-Comp	Memory Requirement (Total Bits)	Data Transmission (Total Bits)
1. BC [5]	$(\frac{1}{2})^n$	$(\frac{1}{2})^n$	×	√	$2f$	×	sk+3n+2F (768)	$5n$+2F (896)
2. HK [6]	$(\frac{3}{4})^n$	$(\frac{3}{4})^n$	×	×	$0f$	√	sk+2N_{rand}+F (320)	2N_{rand}+2n (192)
3. MP [10]	$(\frac{3}{4})^n$	$(\frac{3}{5})^n$	×	×	$2f$	×	sk+2N_{rand}+2F (448)	2N_{rand}+2n+F (276)
4. KA [12]	$(\frac{7}{8})^n$	$(\frac{1}{2})^n$	×	×	$1f$	×	sk+2N_{rand}+F (320)	2N_{rand}+2n (128)
5. Yang *et al.* [14]	$(\frac{3}{4})^n$	$(\frac{3}{4})^n$	×	×	$2f$	×	sk+ID+ID'+2N_{rand} +n+F (554)	2N_{rand}+ID' +3n (288)
6. Zhuang *et al.* [15]	$(\frac{3}{4})^n$	$(\frac{3}{4})^n$	$(\frac{7}{8})^n$	×	$0f$	√	sk+N_{ctr}+ID+ID' +2Flag+n+F (524)	4n+ID'+N_{ctr} (298)
7. Reid *et al.* [13]	$(\frac{3}{4})^n$	1	$(\frac{3}{4})^n$	×	$1f$	×	sk+2ID+2N_{rand}+F +n (640)	2ID+2N_{rand}+2n (512)
8. TP [11]	$(\frac{3}{4})^n$	1	$(\frac{3}{4})^n$	√	$2f$	×	sk+2N_{rand}+n+2F (576)	2N_{rand}+2n+F (448)
9. Swiss-Knife [9]	$(\frac{3}{4})^n$	$(\frac{1}{2})^n$	$(\frac{3}{4})^n$	√	$2f$	√	sk+ID+3N_{rand}+2n +3F (832)	2N_{rand}+3n+2F (704)
10. TDB [27]	$(\frac{3}{4})^n$	$(\frac{2}{3})^n$	$(\frac{5}{6})^m$	×	$1f$	×	sk+2N_{rand} + n^{sh} (192 + 128$^{\text{sh}}$)	2N_{rand}+2n (320)
11. SKI$_{\text{PRO}}$ [28]	$(\frac{3}{4})^n$	$(\frac{2}{3})^n$	$(\frac{5}{6})^m$	×	$2f$	×	sk+2N_{rand}+2F+2n (704)	2N_{rand}+4n (544)
12. AT[7]	$\frac{1}{2}[1+(\frac{1}{2})^n]$	$(\frac{n}{2}+1)(\frac{1}{2})^n$	×	×	$1f$	×	sk+2N_{rand}+F+n See table note†	2N_{rand}+2n+F$\|_1^m$ (384)
Our Protocol	$(\frac{5}{8})^n$	$(\frac{n}{2}+1)(\frac{1}{2})^n$	$(\frac{n}{2}+1)(\frac{n}{9})^n$	×	$0f$	√	sk+ID+N_{ctr}+Flag +4Cur+n+F (541)	3n+N_{ctr} (416)

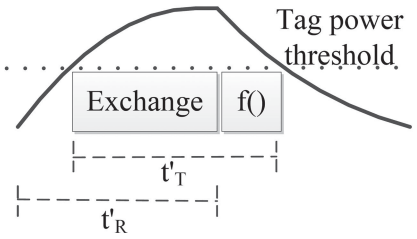
Online Complexity*: Functions computed during online stage.* Offline Pre-Comp: *Offline Pre-Computation.*
f: cryptographic function (it covers all cryptographic computation such as hash/MAC/commitment). sh*: TDB key shares, m/n: Number of iterations ($m < n$).*
Typical/example values for essential elements of last two columns are shown in brackets:
sk: secret key length (128), N_{rand}: random number length (32), N_{ctr}: counter length (32).
ID/ID′: tag's ID or pseudo ID length (96), Cur: cursor length (7), F: cryptographic function result length (128).
n: number of exchanged bits (n depends on F, see discussion below), Flag: flag bit length (1).
†*: Single tree: $(2^{n+1}-2)$ for $n \geq 2$; Multiple tree: $[\frac{2n^2}{\log_2 n}(1-\frac{1}{n})]$ for $n \geq 2$.*

computed using the residual power stored on the tag. The profile of the energy present is simply that of a charging and discharging capacitor, as most RFID tags use a capacitor to store to harvested energy from the reader's carrier, i.e., when charging $V_O = V_{in}(1 - e^{\frac{-t}{RC}})$ and $V_O = V_{max} \cdot e^{\frac{-t}{RC}}$.

For our proposal to bring about these benefits it must be possible for the tag to compute the function f using residual energy, i.e., energy stored in the tag at the point it is no longer powered by the reader. To show that this notion is feasible we implemented a proof-of-concept experiment.



(a) Conventional pre-computation protocol



(b) Proposed post-computation Protocol

FIGURE 7. Timeline of protocols with t_R and t_T showing the period that the reader is supplying power and the time that the tag is activated respectively. The solid line shows the voltage across the power harvesting circuit and the dotted line shows the threshold level for the tag to be activated.

A. EXPERIMENTAL IMPLEMENTATION

Commercial UHF RFID tags do not allow for development of custom on-tag applications. For our RFID tag, we therefore use what is termed a 'computational RFID' tag build using the UMich MOO design [33], which is based on Intel's Wireless Identification and Sensing Platform (WISP) UHF tags [34]. The MOO is a passive UHF device containing a Texas Instruments MSP430 micro-controller unit (MCU), an ultra-low power MCU with a 16-bit instruction set, along with a selection of sensors. The MCU on our device operates from a 2V supply and uses an internal clock of 1.075MHz.

To communicate with our tag we used an Impinj Speedway Revolution UHF RFID reader with an Impinj Brickyard antenna. The tags together with the reader and antenna is shown in Figure 8.

For realising f we use the Advanced Encryption Standard (AES) with a 128-bit key to encrypt one 128-bit data block. The 128-bit data block is sufficient to allow, as an example, for a 32-bit N_T and 96-bit ID. These are realistic values as RFID tags often use a 96-bit Electronic Product Code (EPC) identifiers and a 32-bit counter appears small but is secure in terms of the protocol. For example, if the tag runs continuously and each protocol run is executed in 12.8 ms, theoretical minimal execution time determined by execution time of f given in Section VI-B, then the counter will only wrap around and repeat values after approximately 650 days. Our implementation uses the Texas Instruments AES library for MSP430 devices [35].

B. RESULTS

We measured two tag outputs, as shown in Figure 9, during the experiment: a digital output pin that is set low during the AES encryption, which inherently also goes low as the MCU turns off; and the voltage over the storage capacitor in the

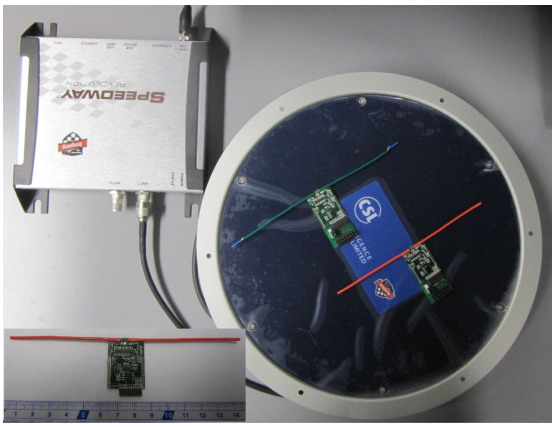


FIGURE 8. Experimental tags together with UHF reader & antenna.

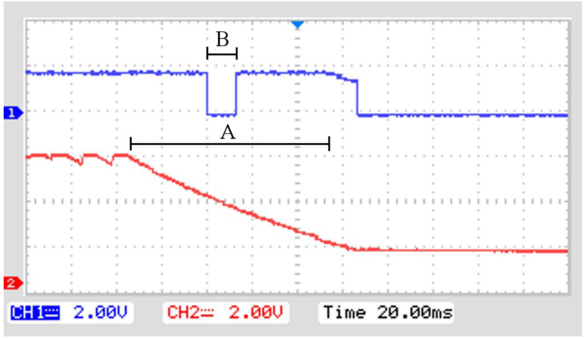


FIGURE 9. Digital output pin of tag MCU [top] and the voltage over the tag's storage capacitor [bottom]. The time taken for the AES computation is indicated by B and the time between the reader's carrier switching off and the MCU starting to run at less than the intended 2V is indicated by A.

tag's power supply/harvester. The time needed to compute f was measured as 12.8 ms. In comparison, the time that the MCU continued to run at the intended supply voltage (at 2V) after the reader's carrier was switched off was measured as 81.2 ms. These results show that the idea of pre-computing f for a protocol run during the preceding run, using residual energy, is feasible.

On the chosen tag, the time the MCU can continue to run is significantly more than is needed and the system design could be modified to allow for encryption of multiple data blocks, which allows for a longer W, N_T or ID. It also means that the size of the capacitor could be reduced, which will decrease the start-up time, i.e., the time needed for the storage capacitor to charge to 2 V. Currently the start-up time of the tag is approximately 13 ms with a storage capacitor of 10 μF. We did an alternative implementation with SHA-256 as f (also using a TI library) and this needed 100 ms to compute, which means that it was not possible for the tag to complete the pre-computation before the residual energy was exhausted.

VII. CONCLUSION

In this paper, we propose an energy and time efficient distance-bounding protocol that provides comparable security bounds to existing protocols. We provide a security analysis of our protocol using an accepted framework for common attack scenarios, compare our protocol's security and data storage/transmission metrics with related work and also demonstrate the feasibility of pre-computing the function f using residual energy in a resource-constrained passive UHF RFID device. The design of the pre-computation input means the calculation of f is independent of reader input, and the tag can compute the response register for the next protocol run at the conclusion of the previous protocol run using residual energy accumulated during interaction with the reader. This also means that there is no significant cryptographic calculation during the reader and tag communication, which shortens the execution time and decreases the period that the reader needs to transmit an RF carrier to power the tag.

ACKNOWLEDGMENTS

The work described in this paper was fully supported by a grant from City University of Hong Kong (Project No. 7004473). And grants from National Science Foundation of China (Grant No. 61702222) and China Postdoctoral Science Foundation (Grant No. 2017M612842).

REFERENCES

[1] P. Papadimitratos, *et al.*, "Secure neighborhood discovery: A fundamental element for mobile ad hoc networking," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 132–139, Feb. 2008.

[2] A. Ranganathan, B. Danev, and S. Capkun, "Proximity verification for contactless access control and authentication systems," in *Proc. 31st Annu. Comput. Secur. Appl. Conf.*, 2015, pp. 271–280.

[3] B. Wang and M. Ma, "A server independent authentication scheme for RFID systems," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 689–696, Aug. 2012.

[4] G. Avoine, *et al.*, "Security of distance-bounding: A survey," *ACM Comput. Surveys*, 2017.

[5] S. Brands and D. Chaum, "Distance-bounding protocols," in *Proc. Workshop Theory Appl. Cryptographic Techn. Advances Cryptology*, 1994, pp. 344–359.

[6] G. P. Hancke and M. Kuhn, "An RFID distance bounding protocol," in *Proc. 1st Int. Conf. Secur. Privacy Emerging Areas Commun. Netw.*, 2005, pp. 67–73.

[7] G. Avoine and A. Tchamkerten, "An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement," in *Proc. Inf. Secur. Conf.*, Sep. 2009, pp. 250–261.

[8] Y. Desmedt, "Major security problems with the 'unforgeable' (Feige)-Fiat- Shamir proofs of identify and how to overcome them," in *Proc. Worldwide Congr. Comput. Commun. Secur. Protection*, 1988, pp. 15–17.

[9] C. H. Kim, G. Avoine, F. Koeune, F.-X. Standaert, and O. Pereira, "The swiss-knife RFID distance bounding protocol," in *Proc. Int. Conf. Inf. Secur. Cryptology*, 2008, pp. 98–115.

[10] J. Munilla and A. Peinado, "Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels," *Wireless Commun. Mobile Comput.*, vol. 8, no. 9, pp. 1227–1232, 2008.

[11] Y. J. Tu and S. Piramuthu, "RFID distance bounding protocols," in *Proc. 1st Int. EURASIP Workshop RFID Technol.*, 2007, pp. 67–68.

[12] C. H. Kim and G. Avoine, "RFID distance bounding protocol with mixed challenges to prevent relay attacks," in *Proc. Int. Conf. Cryptology Netw. Secur.*, Dec. 2009, pp. 119–133.

[13] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji, "Detecting relay attacks with timing-based protocols," in *Proc. 2nd ACM Symp. Inf. Comput. Commun. Secur.*, 2007, pp. 204–213.

[14] A. Yang, Y. Zhuang, and D. Wong, "An efficient single-slow-phase mutually authenticated RFID distance bounding protocol with tag privacy," in *Proc. Int. Conf. Inf. Commun. Secur.*, 2012, pp. 285–292.

[15] Y. Zhuang, A. Yang, D. Wong, G. Yang, and Q. Xie, "A highly efficient RFID distance bounding protocol without real-time PRF evaluation," in *Proc. Int. Conf. Netw. Syst. Secur.*, 2013, pp. 451–464.

[16] N. O. Tippenhauer, H. Luecken, M. Kuhn, and S. Capkun, "UWB rapid-bit-exchange system for distance bounding," in *Proc. 8th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, 2015, pp. 2:1–2:12.
[17] G. P. Hancke, "Distance-bounding for RFID: Effectiveness of 'terrorist fraud' in the presence of bit errors," in *Proc. IEEE Int. Conf. RFID-Technol. Appl.*, Nov. 2012, pp. 91–96.
[18] A. Abu-Mahfouz and G. Hancke, "Distance bounding: A practical security solution for real-time location systems," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 16–27, Feb. 2013.
[19] M. Bolic, M. Rostamian, and P. M. Djuric, "Proximity detection with RFID: A step toward the internet of things," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 70–76, Apr. 2015.
[20] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Potential misuse of NFC enabled mobile phones with embedded security elements as contactless attack platforms," in *Proc. Int. Conf. Internet Technol. Secured Trans.*, Nov. 2009, pp. 1–8.
[21] L. Francis, G. Hancke, K. E. Mayes, and K. Markantonakis, "Practical NFC peer-to-peer relay attack using mobile phones," in *Proc. 6th Int. Conf. Radio Freq. Identification: Secur. Privacy Issues*, 2010, pp. 35–49.
[22] Y. Zhuang, A. C. M. Leung, and J. Hughes, "Matching in proximity authentication and mobile payment ecosystem: What are we missing?" in *Proc. 12th Workshop Radio Freq. Identification IoT Secur.*, 2017, pp. 163–172.
[23] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial internet of things architecture: An energy-efficient perspective," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 48–54, Dec. 2016.
[24] K. Wang, M. Du, Y. Sun, A. Vinel, and Y. Zhang, "Attack detection and distributed forensics in machine-to-machine networks," *IEEE Netw.*, vol. 30, no. 6, pp. 49–55, Nov./Dec. 2016.
[25] G. Avoine, M. A. Bingöl, S. Kardas, C. Lauradoux, and B. Martin, "A framework for analyzing RFID distance bounding protocols," *J. Comput. Secur.*, vol. 19, no. 2, pp. 289–317, 2011.
[26] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.
[27] G. Avoine, C. Lauradoux, and B. Martin, "How secret-sharing can defeat terrorist fraud," in *Proc. ACM Conf. Secur. Privacy Wireless Mobile Netw.*, 2011, pp. 145–156.
[28] I. Boureanu, A. Mitrokotsa, and S. Vaudenay, "Practical & provably secure distance-bounding," in *Proc. Inf. Secur. Conf.*, 2013, pp. 248–258.
[29] M. Fischlin and C. Onete, "Terrorism in distance bounding: Modeling terrorist-fraud resistance," in *Proc. 11th Int. Conf. Appl. Cryptography Netw. Secur.*, 2013, pp. 414–431.
[30] A. Bay, I. C. Boureanu, A. Mitrokotsa, I.-D. Spulber, and S. Vaudenay, "The Bussard-Bagga and other distance-bounding protocols under attacks," in *Proc. 8th China Int. Conf. Inf. Secur. Cryptology*, Nov. 2012, pp. 371–391.
[31] I. Boureanu, A. Mitrokotsa, and S. Vaudenay, "On the pseudorandom function assumption in (secure) distance-bounding protocols," in *Proc. 2nd Int. Conf. Cryptology Inf. Secur. Latin America*, 2012, pp. 100–120.
[32] A. Corrales Paredes, M. Malfaz, and M. Salichs, "Signage system for the navigation of autonomous robots in indoor environments," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 680–688, Feb. 2014.
[33] UMich MOO, "A batteryless programmable RFID-scale sensor device," [Online]. Available: https://spqr.eecs.umich.edu/moo/, Accessed on: Aug. 24, 2014.
[34] A. Sample, D. Yeager, P. Powledge, A. Mamishev, and J. Smith, "Design of an RFID-based battery-free programmable sensing platform," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 11, pp. 2608–2615, Nov. 2008.
[35] U. Kretzschmar, "AES128—A C implementation for encryption and decryption," MSP430 Systems, Texas Instruments, Dallas, TX, USA, Appl. Rep. SLAA397A, 2009.

YUNHUI ZHUANG received the MSc and PhD degrees both in computer science from City University of Hong Kong (CityU), in 2010 and 2016, respectively. He is now a postdoctoral fellow in the Department of Information Systems, College of Business, City University of Hong Kong. His research interests lie at the intersection of economics and information security. In particular, he is interested in applied cryptography, security and privacy of mobile payment, blockchain, business analytics, e-Learning, and applied econometrics. He has served as the co-chair or the Program Committee member for some prestigious international conferences in applied cryptography, information security, and e-Learning.

ANJIA YANG received the BS degree from Jilin University, in 2011 and the PhD degree from City University of Hong Kong, in 2015, respectively. He is currently a postdoctoral researcher in Jinan University, Guangzhou. His research interests include blockchain security, RFID security and privacy, applied cryptography, and cloud computing.

GERHARD P. HANCKE received the BEng and MEng degrees in computer engineering from the University of Pretoria, Pretoria, South Africa, in 2002 and 2003, respectively, and the PhD degree in computer science from the University of Cambridge, United Kingdom, in 2008. He is currently an assistant professor with Department of Computer Science, City University of Hong Kong, Hong Kong. His research interests include system security, embedded platforms, and distributed sensing applications. He is a senior member of the IEEE.

DUNCAN S. WONG received the BEng degree from the University of Hong Kong, in 1994, the MPhil degree from the Chinese University of Hong Kong, in 1998, and the PhD degree from Northeastern University in Boston, in 2002. He was with the City University of Hong Kong for 12 years, and was the vice president of Financial Technologies in ASTRI since 2014. He is the co-founder and CEO of CryptoBLK Limited. He has authored more than 200 research papers in international journals and conferences, and served as a member, including the chair of the Program Committee for more than 90 prestigous international conferences in cryptography and information security. His research interests include applied cryptography, information security, and blockchain. He is a senior member of the IEEE.

GUOMIN YANG received the PhD degree from the Computer Science Department, City University of Hong Kong, in 2009. Formerly, he worked as a research scientist in the Temasek Laboratories, National University of Singapore. He is currently a senior lecturer with the School of Computing and Information Technology, University of Wollongong, Australia. He has been awarded a prestigious Australian Research Council DECRA fellowship award. His research interests include cryptography and network security. He is a member of the IEEE.