Privacy-preserving Group Authentication for RFID Tags Using Bit-Collision Patterns

Anjia Yang, *Member, IEEE,* Dutliff Boshoff, Qiao Hu, Gerhard Hancke*, *Senior Member, IEEE,* Xizhao Luo*, *Member, IEEE,* Jian Weng, *Member, IEEE,* Keith Mayes, and Konstantinos Markantonakis

Abstract—When authenticating a group of RFID tags, a common method is to authenticate each tag with some challenge-response exchanges. However, sequentially authenticating individual tags one by one might not be desirable, especially when considering that a reader often has to deal with multiple tags within a limited period, since it will incur long scanning time and heavy communication costs. To address these problems, we put forward a novel efficient group authentication protocol where a group of tags can be authenticated simultaneously with only one challenge and one response. The protocol is built on a new designed symmetric key based algorithm and the bit-collision pattern technique, so that authentication responses transmitted by multiple tags in a group at the same time will result in a verifiable bit-collision pattern that represents the authentication response for the entire group. The proposed approach can significantly reduce the authentication time and communication cost in sense that the verifier can authenticate the entire group within a period that is comparable to the time taken to perform a single tag authentication and requires only one challenge. In addition, we extend our protocol to support privacy-preserving property which prevents the tagged items from being tracked by illegitimate parties. A thorough security analysis shows that the proposed protocol can resist common practical attacks and experimental results show that the protocol is very efficient in terms of time and communication costs. We also discuss important practical aspects that should be considered when implementing these protocols.

Index Terms—Group authentication, RFID, bit-collision pattern, privacy-preserving.

1 Introduction

Radio Frequency IDentification (RFID) plays a critical role in the Internet of Things (IoT) such as tracking applications in supply chain systems. Connecting RFID readers and tags to the Internet allows objects attached with tags to be effectively identified and tracked all over the world. With the growing usage of RFID, however, security and privacy issues surrounding this technology have also attracted increasing attention. Even though some RFID devices, such as contactless smart cards, implement a selection of standard cryptographic functions (e.g. DES, AES, RSA), these RFID tags used as electronic 'labels' in item tracking services generally do not. This is mostly due to the fact that item-level labeling requires a large volume of low-cost RFID tags and that a small price increases per tag would cause a large increase in operating expense as a result of implementing extra processing logic or memory required for security mechanisms.

Nevertheless, as the RFID technology progresses, tags could be probably indispensable to run security-related functions for achieving secure tracking systems where attackers should not be able to counterfeit items or remove genuine items without being detected. Methods to implement additional security functionality while keeping tags cost efficient (e.g., implementing 'lightweight', or minimalist, security mechanisms for RFID systems) have been extensively explored. The main idea is to employ authentication protocols [1]–[4] to eliminate these attacks.

In conventional authentication mechanisms, a reader sends a challenge to a tag which will then return a response generated based on the challenge and a secret for authentication purpose. However, it is not always feasible to authenticate each tag individually especially in case that a group of tags are required to be verified within a short period. As an instance, a set of items embedded with RFID tags could be quickly moving on a conveyor belt or placed in a box on a truck that is rapidly passing through the reader with limited scanning time. In this case, all the tags are supposed to be authenticated by the reader as fast as possible. Therefore, the interaction time between the reader and each tag should also be considered upon designing secure protocols for RFID systems. Namely, it is indispensable to construct a highly efficient group authentication protocol which allows the reader to authenticate multiple tags more effectively.

Juels [5] introduced the concept of yoking-proof where the system can verify the existence of two tags at the same time. Later, an extension of yoking-proof is intro-

- *Anjia Yang and Jian Weng are with the College of Cyber security, National Joint Engineering Research Center of Network Security Detection and Protection Technology, and Guangdong Key Laboratory of Data Security and Privacy Preserving, Jinan University, Guangzhou, 510632, China. E-mail: {anjiayang, cryptjweng}@gmail.com*
- *Dutliff Boshoff is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. E-mail: dboshoff2-c@my.cityu.edu.hk*
- *Qiao Hu is with the College of Computer Science and Electronic Engineering , Hunan University, Changsha, 410000, China. Email: qiaohu2-c@my.cityu.edu.hk*
- *Gerhard Hancke is with the Department of Computer Science, City University of Hong Kong, Hong Kong. Email: gp.hancke@cityu.edu.hk*
- *Xizhao Luo is with School of Computer Science and Technology, Soochow University, Suzhou 215006, China. E-mail: xzluo@suda.edu.cn*
- *Keith Mayes and Konstantinos Markantonakis are with the Information Security Group at Royal Holloway, University of London, UK. Email: {Keith.Mayes, K.Markantonakis}@rhul.ac.uk*
- ** Corresponding authors (Gerhard Hancke, Xizhao Luo)*

duced: grouping proof, in which a group of tags (could be more than two) can prove their existence to the system simultaneously. There have been a number of grouping proof protocols in the literature [6]–[16]. Nevertheless, the majority of existing grouping proof protocols essentially do not authenticate the group of tags simultaneously, but actually authenticate them sequentially within a specified time. In particular, the reader sends a request for a grouping proof, and the first tag receiving the query will generate a response that will be passed to the next tag which takes the response as an input to calculate its own response. The same process is repeated by the other tags until the last one which will send the final response to the reader. These responses constitute the so-called grouping proof. It is not hard to figure out that this kind of grouping proof protocols have marginal difference in terms of running time from protocols that authenticate tags one by one individually, since the reader cannot verify the grouping proof until having received all the tags' separate responses. On the other hand, a few protocols require the reader to broadcast a request based on which all of the group of tags generate responses (i.e. the grouping proof) at the same time. This method could be scalable, but it is critical to deal with the collision issues if all the tags send their responses simultaneously through the same communication channel. Thus designing an efficient anti-collision mechanism becomes the key factor. Due to the collision avoidance, the group of tags cannot send all their responses at the same time.

To further reduce the time cost in terms of authenticating a group of tags, we proposed a new group authentication protocol [17] which enables the reader to authenticate all the tags in a group simultaneously. The time of authenticating multiple tags is comparable to that of authenticating a single tag. As an extension of this work, in this article we improve the protocol with more comprehensive elaboration of the design and thorough security analysis, and extend the basic protocol to a privacy-preserving version. Moreover, we also make performance comparison and evaluation of the proposed protocols with both theoretical analysis and real experiments. The main contributions are summarized as follows:

- We propose a new lightweight group authentication protocol that can verify a group of tags simultaneously with only one challenge and one response, that is, the entire group of tags can be authenticated within a period comparable to the time occurred to perform a single tag authentication with minimum communication cost. In particular, the protocol is built on a new designed symmetric key based algorithm and the bit-collision pattern technique, so that authentication responses transmitted by multiple tags in a group at the same time will result in a verifiable bit-collision pattern that represents the authentication response for the entire group.
- We extend the basic protocol to support privacy-preserving property which prevents the tagged items from being tracked by illegitimate parties based on a pseudonym technique. We further provide a thorough security analysis of the proposed protocol, demonstrating how the five prominent attacks can be prevented. The analysis results show that a low attack probability can be ensured as long as we select proper parameters. In addition, both protocols employ only a keyed pseudo-random function, permutation operations and simple bit rotations, which makes our protocols lightweight. To verify this, we compare the proposed protocol with existing group authentication protocols in terms of time cost, communication cost, and security. The results show that our protocol is much more efficient without sacrificing security.
- We discuss important practical aspects that should be considered when implementing the proposed protocol. We have conducted experiments with groups of different sizes consisting of multiple tags transmitting the same sequence simultaneously. The reader we used allowed for a maximum group size of 11, after which the collision was no longer reliably detected. A more powerful reader with a different receiver architecture might allow for a larger group. Therefore, our protocol is specially suitable for practical applications that require to authenticate a large number of groups containing a limited number of tags.

The remainder of this article is organized as follows. The background and related work on group authentication for RFID tags are introduced in Section 2, where we classify group authentication into three different modes in terms of the interaction time. The research problems are formulated in Section 3. In Section 4, the basic group authentication protocol using bit-collision patterns is presented in Section 4. Then Section 4.2 elaborates how to extend the basic protocol to a privacy-preserving one. The comparison of some selected existing group authentication protocols with our proposed protocol is shown in Section 5. Finally the practical considerations when implementing the proposed protocol are discussed in Section 6 and this paper is concluded in Section 7.

2 Background and Related Work

The concept of cryptographically authenticating a group of items has been introduced for more than a decade. In 2004, Juels [5] took the first step with coining a similar definition (i.e., the yoking-proof) where a reader can verify the existence of two tags at almost the same time. Their protocol is shown in Figure 1 and briefly described as the following:

1) Two tags t_A and t_B share secret keys k_A and k_B with the verifier V, which in this case is not the reader but the backend system.
2) The reader R *Query* t_A which randomly selects a nonce r_A and sends r_A, A to R, where A is t_A's identity.
3) R forwards r_A and queries tag t_B, which calculates $m_B = \mathrm{MAC}_{k_B}[r_A]$ using its secret key k_B and tag t_A's random nonce. Similarly, t_B also selects a random nonce r_B and sends r_B, m_B as well as its identity B to R.

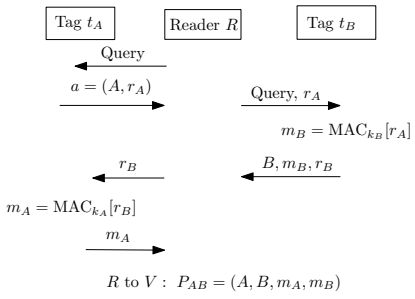


Fig. 1. Yoking Proof for RFID Tags

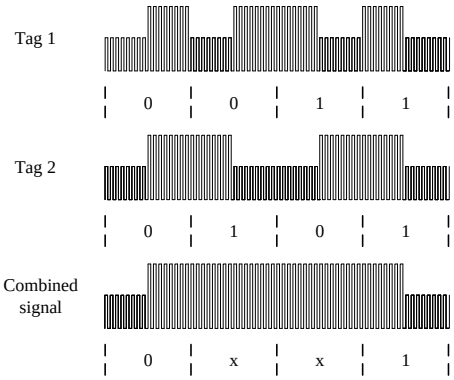


Fig. 3. Constructing bit-collision patterns with Manchester code

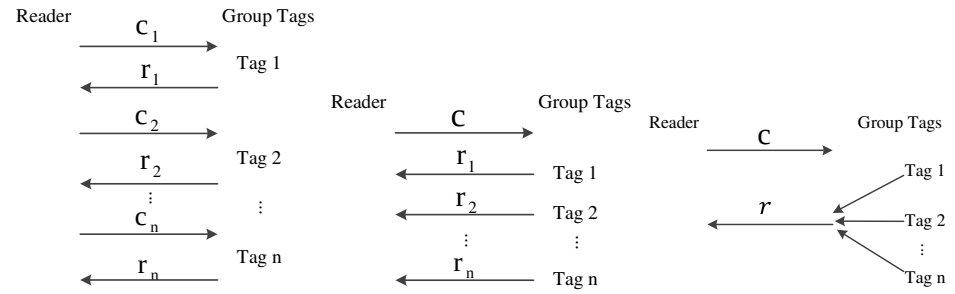
4) R forwards r_B to t_A, which calculates $m_A = \mathrm{MAC}_{k_A}[r_B]$ using its secret key k_A and the r_B value it just received. m_A is then sent to R.
5) R sends the proof, $P_{AB} = (A, B, m_A, m_B)$ and r_A, r_B to the backend system V.

The verifier V can generate P'_{AB} using its version of the tag keys and the nonces contained in the proof received from the reader. It can then check that $P'_{AB} = P_{AB}$ to verify that t_A and t_B were presented simultaneously. Despite the verifier never communicating with the tags directly it has also verified that the two tags the reader communicated with are the genuine t_A and t_B, as both used the correct secret shared keys k_A and k_B. The proof therefore also serves as a basic method for group authentication. As an extension, the notion of grouping-proof which allows authentication of a group of tags (could be more than two) is introduced and explored by [6]. Liu *et al.* [11] proposed a grouping-proof based authentication protocol for distributed RFID systems in 2013. After that until recently, a lot of researchers have proposed new grouping proof protocols with efforts on providing security or privacy for RFID systems [7]–[10], [12]–[16], [18]–[22]. For instance, Shen *et al.* [20] proposed an efficient grouping-proof protocol that requires less message interaction and computation overhead on the tags. However, it is analyzed by Dhailah *et al.* [22] that their protocol was vulnerable to a full-disclosure attack.

Although existing grouping proofs support the verification of completeness and soundness of a group, the factor of transaction time has been seldom considered. Some protocols claim that they can authenticate a group of tags "simultaneously". However, essentially, the reader authenticates the group of tags sequentially within a specified time bound in their protocols. To make it more clear, we draw a figure to show different types of protocols that aim to provide group authentication in Figure 2. Existing grouping proof protocols belong to a variety of the first type, where each tag sends an individual response sequentially. In a grouping proof protocol, the verifier sends the first challenge to a tag which outputs its response. Next tag's challenge is the last tag's response, just like a chain. In total, with n tags, there are n challenges and n responses transmitted at different time in the grouping proof protocol. Recently, Chen *et al.* [12] presented a provable secure batch authentication scheme for EPCGen2 RFID tags. The reader sends only one challenge and obtains multiple responses from different tags and then makes aggregative computation to generate an authenticated message which is used for batch authentication. However, they do not mention how the group tags' responses are collected simultaneously without collisions.

To allow for any potential timing restrictions it would be ideal if a group of tags are able to truly be authenticated simultaneously utilizing only a single challenge-response sequence, namely, type 3 in Figure 2(c). This would require that all tags send their corresponding responses simultaneously. Fortunately, some RFID technologies already allow for this scenario. For instance, in systems adhering to the ISO 14443 [23] and ISO 15693 [24] (ISO 18000-3 [24]) standards multiple tags simultaneously transmit information to the reader during the anti-collision process. The tokens' responses are Manchester coded and they are synchronized to start transmitting in the same bit period, which as a consequence can create clearly defined bit collisions during certain bit periods, as shown in Figure 3. This figure shows the Manchester-encoded data, which is first modulated onto a sub-carrier, load modulated onto the HF carrier with 8-12 % modulation depth. The reader therefore observes a bit pattern containing both collisions and non-collision values. Although not all RFID tags use Manchester encoding, there could be some way to build bit-collision patterns for tags using other encoding methods. Taking EPC Class-1 Generation-2 (ISO 18000-6C) [25] tags as an example, the data are encoded as the following: a signal that keeps high or low for the entire bit duration represents '1' while a signal changed from high to low or from low to high at the half of the bit period represents '0'. The two '0' symbols are therefore identical to the symbols used in Manchester coding and could thus be utilized to build a bit-collision pattern similarly.

Bit-collisions have been applied to privacy-preserving security mechanisms by intentionally blocking tag responses to unauthorized readers [26] and to key exchange protocols [27], [28], but have not been employed to design authentication protocols. The intuitive idea of exploiting bit-collisions is based on the assumption that no attacker can distinguish the individual bits sent from multiple tags if a collision happens. Hancke [29] showed that in the case of two tags sending messages at the same time and resulting in a bit-collision pattern, an attacker has some chance to infer the individual responses due to variations in the communication channel of passive tags. However, the



(a) Multiple Challenges Multiple Responses. (b) One Challenge Multiple Responses. (c) One Challenge One Response.

Fig. 2. Three Types of Protocols that Aim to Provide Group Authentication.

attacker's success probability will decrease if the number of tags increases.

One similar research topic is to achieve fast identification of large-scale RFID tags based on anti-collision mechanisms including *Aloha-based* [30] and *Tree-based* [31] protocols. The combination of *Aloha-based* and *Tree-based* protocol is also considered [32]. However, these designs are essentially the same as per-tag identification (type 1), i.e., identifying tags sequentially, with the advantage of making full use of the communication channel. Yang *et al.* [33] proposed a probabilistic batch authentication protocol which means that a group of tags are authentic with probability greater than $1-\delta$ if the number of fake tags is less than $n*\epsilon$, where δ and ϵ are two security parameters. Based on this work, they proposed an extended batch authentication scheme that further addressed the scalability issue recently [34]. In contrast, our protocol is deterministic which guarantees a batch is authentic with probability of 1.

With regards to the privacy issues, there have been a lot of related works in the literature [18], [35]–[46] that can be adapted to RFID systems. Generally, pseudonym-based approach like [43] provides untraceability of the prover and employs computationally efficient algorithms such as symmetric encryption algorithms and hash functions. On contrast, public-key cryptography based approach can provide more desirable functions such as key exchange [18] except for mere untraceability, but these schemes require much heavier computation on the prover side and thus might not be enjoyable for RFID systems.

3 System Model, Threat Model, and Design Goals

We formulate the problem by describing the system and threat models and identifying the design goals of group authentication for RFID systems.

3.1 System Model

As shown in Fig. 4, the system contains a group of tags, a reader and a verifier. Each of the entities is defined as follows:

Group Tags. A number of tags embedded in related objects (one tag per object) are divided into a group. Note that the group of tags are physically packed together and thus they will be interrogated together by the same reader.

Fig. 4. The system model

Each tag stores some secret information shared with the verifier that will be used for the authentication.

Reader. An RFID reader is a device that gathers information from RFID tags. It starts a protocol session with a querying message and relays the information transmitted between the verifier and the tags, which helps the verifier authenticate the tags. The reader itself does not need to share secret keys with the tags or the verifier.

Verifier. The verifier could be a PC or a server connected with a database and stores the necessary information used to authenticate the group tags. The verifier does not have to be online all the time, but be notified to start a protocol by a reader as long as the reader queries a group of tags. Whenever the tags are transported to a checking place, a reader will connect with the verifier and start the protocol.

A group authentication protocol is comprised of three phases: initialization phase, challenge-response phase and verification phase. The tags and the verifier prepare to start a protocol session after receiving a start command from the reader in the initialization phase. The verifier sends out challenges to the tags and receives corresponding responses back in the challenge-response phase. Finally, the verifier verifies the responses to make a decision in the verification phase. Unlike a common authentication protocol, in the group authentication protocol, the verifier authenticates a group of tags in batch, rather than authenticating each tag of the group one by one in sequence. The group authentication protocol is said to be *complete* if valid group tags are always accepted by the verifier, and to be *sound* if only valid group tags are accepted, that is, no attacker can impersonate a tag without being detected.

3.2 Threat Model

We consider a probabilistic polynomial time adversary which can eavesdrop, intercept and modify messages trans-

mitted in the protocol run. The adversary could also observe the result of a session, that is, whether the verifier and the group tags accept each other.

We also consider a more powerful attacker which has two more capabilities. First, the strong attacker can observe the individual output of each tag. This is truly a powerful capability for the attacker since it is unlikely during normal system operations. In practice, the adversary can only eavesdrop the composite outputs from the group tags, rather than individual tag outputs. Nevertheless, we'd like to explore whether our protocol is still secure against so powerful attacks. The second capability is that when transmitting the authentication proof/state, the strong attacker can determine the bit value and prepare its response right at the start of the bit period, even though this might be unrealistic in practice as RF receivers usually integrate or sample over the entire bit period. In particular, the authentication state consists of multiple bit pairs. For each bit pair, the attacker could observe the first composite bit from the other tags and then decide what to do for the second bit. The benefits of this for the attacker will be discussed later in the security analysis in Section 4.3.

3.3 Design Goals

Based on the system model and security threats above, we define the design goals for a group authentication protocol in terms of three aspects: security, efficiency and privacy, elaborated as follows.

- *Security*: The group authentication protocol should allow the verifier to authenticate a group of tags simultaneously and detect whether any tag in the group has been replaced or simply lost with an overwhelming probability. More precisely, the protocol should be able to prevent the five prominent attacks that will be discussed in Section 4.3.
- *Efficiency*: The design should be computationally efficient especially on the tag side.
- *Privacy*: A privacy-preserving protocol should ensure that no attacker can trace the group of tags according to their transaction histories.

4 Group Authentication based on Bit-Collision Patterns

A usual construction of authentication protocols is built on a challenge-response interaction between the prover and the verifier. The proposed protocol works on the principle that a group of tags transmitting their individual authentication responses simultaneously will result in a verifiable bit collision pattern that represents the authentication response for the entire group. Before presenting the protocol design, we first introduce the bit collision operation with more details. Let $\beta, \beta' \in \{0, 1\}$ be two bit symbols. The collision operation between β and β' is denoted as $\beta \between \beta'$ and x is denoted as a bit collision. The values of $\beta \between \beta'$ with different inputs are as the following:

β	β'	$\beta \between \beta'$
0	0	0
0	1	x
1	0	x
1	1	1

In particular, as long as there are two tags sending different bit values in the same bit slot, the composite value will be a bit collision x regardless of the values sent by other tags in the group, i.e. $1 \between x = x$ and $0 \between x = x$.

In the presented protocol, each tag t_i ($1 \leq i \leq n$) owns an authentication state s_i, where n is the number of tags in the group. If these tags send their individual authentication states at the same time, the reader will receive a composite group authentication state S, i.e. $S = s_1 \between s_2 \between \ldots \between s_n$, which can be utilized for authentication of the group tags. If each tag contributes equal number of bit collisions (at least one), the completeness of the group can be verified easily by checking the total number of bit collisions. If the number of bit collisions is less than expected values, the verifier knows that at least one tag is missing or some fake tags with incorrect authentication states have been placed in the group. In the case of a complete group but some collisions are in wrong bit positions, the verifier knows that the group is not sound. More precisely, the verifier can precompute and know which tag should contribute bit collisions to which positions. If some bit collision occurs in an unexpected position, the verifier can conclude that this is a fake tag. For instance, suppose a group containing four tags and each tag contributes one collision to an 8-bits group authentication state, then the group authentication procedure runs as the following:

tags	correct	missing s_4	fake s_4
s_1	01000011	01000011	01000011
s_2	01001001	01001001	01001001
s_3	01100001	01100001	01100001
s_4	11000001	missing	01010001
S	$x1x0x0x1$	$01x0x0x1$	$01xxx0x1$

In the case that the group is sound and complete, namely, all the tags in the group transmit correct authentication states as depicted in the second column, the verifier will observe four bit collisions in the positions of $(1, 3, 5, 7)$. Otherwise, in the case that the group is incomplete, i.e., there is some tag missing such as s_4, the verifier will only observe three bit collisions and thus can detect a missing tag event, as depicted in the third column. In the case that the group is not sound, i.e., there is some tag (e.g., s_4) replaced by a counterfeit one, the verifier will observe wrong bit-collision patterns rather than expected, as depicted in the fourth column.

Based on the bit-collision patterns, we propose two group authentication protocols, where the basic one allows a verifier to check the the completeness and soundness of a group of tags, hence protecting the tags in the group from being stolen or replaced by attackers during shipment, and the extended one also considers the privacy of the group/tags. To keep the privacy, we mean that the group/tags cannot be traced. The basic protocol is much more efficient but violating the group privacy, while the extended protocol requires more computation but preserving the privacy. In practice, it depends on the purpose or the usability to decide which protocol to be deployed.

4.1 The Basic Protocol

The basic proposal is depicted in Figure 5. We consider a group with n tags. Each tag shares a keyed pseudo-random

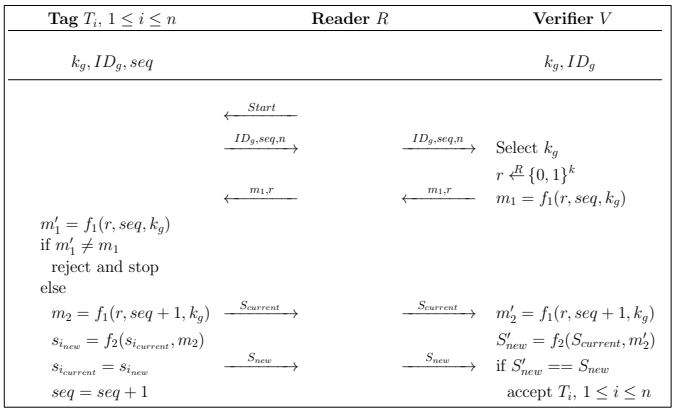


Fig. 5. The proposed basic group-authentication protocol

function f_1, a state permutation function f_2, the group identifier ID_g and the group key k_g with the verifier. In addition, each tag t_i ($1 \le i \le n$) is initialized with a unique authentication state $s_{i_{\text{current}}}$ and a common counter seq.

1) The reader R broadcasts a query information such as a *Start* command to start the protocol.
2) Upon receiving the query, all the tags return ID_g, seq and n simultaneously. Due to the fact that these tags are sending the same information at the same time, there should be no bit collisions at the reader side. R relays the received composite information to the verifier V. In case of the existence of bit collisions, V can decide what kind of errors is happening. In particular, if the collisions occur only in seq, V judges that some tags must have been desynchronized. If the collisions occur in seq and ID_g, V judges that that illegitimate tags have responded.
3) V selects a random string r and computes $m_1 = f_1(r, seq, k_g)$. It then transmits the bit string r and m_1 to R which relays the information to the tags.
4) All the tags in group ID_g compute $m_1' = f_1(r, seq, k_g)$ and check whether m_1' is equivalent to m_1, which achieves the authentication of V. If V is authentic, all the tags send out their individual current authentication state $s_{i_{\text{current}}}$, which allows V to learn the current composite group authentication state S_{current}.
5) Each tag t_i updates its current authentication state with f_2 by calculating $s_{i_{\text{new}}} = f_2(s_{i_{\text{current}}}, m_2)$, where $m_2 = f_1(r, seq+1, k_g)$. Subsequently, all the tags send out their new states $s_{i_{\text{new}}} (1 \le i \le n)$ which will form into a new composite group state S_{new}. Then, R relays S_{new} to V.
6) V computes $S'_{\text{new}} = f_2(S_{\text{current}}, m_2')$, where $m_2' = f_1(r, seq+1, k_g)$. Then it can verify the group by checking $S_{\text{new}} \stackrel{?}{=} S'_{\text{new}}$.
7) All the tags update the value of seq and $s_{i_{\text{current}}}$.

If $S_{\text{new}} \neq S'_{\text{new}}$, the verifier detects an exception which helps people to make further investigation on the group of objects. With the advantage of the proposed protocol that any subset of the group can be simultaneously authenticated by the verifier with running the same protocol, the tags causing the exception can be identified very efficiently. For instance, the verifier can authenticate a single tag t_i by only running the protocol with t_i and comparing whether $s_{i_{\text{new}}} = f_2(s_{i_{\text{current}}})$. In this special case, both $s_{i_{\text{new}}}$ and $f_2(s_{i_{\text{current}}})$ are traditional bit strings without collisions. Similarly, the verifier can simultaneously verify a subgroup of ℓ tags each of which contributes c collisions. In this case, the verifier utilizes $f_2(S^{\ell}_{\text{current}})$ to check whether S^{ℓ}_{new} includes ℓc collisions in the correct bit positions along with an additional $(2n-\ell)c$ correct non-collision values. Therefore, the verifier can execute the protocol with individual tags or split the group into smaller subgroups, in order to identify missing or counterfeit tags.

4.1.1 Construction of S and f_2

Since the group initialization and the authentication state update are critical to the proposed protocol, we now elaborate how to initialize and update the authentication state (i.e., how to construct the state permutation function f_2).

Initialization of authentication state: Before the items (embedded with tags) are shipped, the sender needs to initialize the authentication state for each individual tag in a group so that they can be authenticated by the verifier during shipment. In order to verify the completeness and soundness of a group, each of tags in the group should contribute at least one bit collision to the composite group authentication state. Otherwise, if there are some tags which do not cause any bit collision, the attacker can just simply remove them from the group without affecting the authentication results. Furthermore, the number of bit collisions contributed by each of the tags should be identical, which would enable the verifier to calculate the number of missing tags in case that the group is not complete. For example, considering a group that contains two tags, A and B each of which contributes one collision, and another two tags, C and D each of which contributes two collisions. In case of lacking two collisions in the group authentication state, the verifier cannot determine whether two tags ($A+B$) or one tag (either C or D) are missing. Finally, in our design, each bit collision is paired with a non-collision bit such that the bit swap operation in f_2 (described in the following section) will always occur a change on the position of a bit collision. As a consequence, suppose each tag contributes c bit collisions, each of which is associated with a non-collision bit, then the bit length of the group authentication state S and the individual tags' states s_i should be $2cn$, where n is the number of tags in the group. In the following, we illustrate how the authentication state is initialized (by the sender).

1) Determine the value of c, i.e., the number of bit collisions which each tag shall contribute. If both c and n are even numbers then $2cn$ will always being a multiple of eight, which is a useful property if the communication channel is byte oriented. If this is a desired property and a shipment consists of an uneven number of items the sender could add a single 'padding' tag to make n even.
2) Generate a state matrix M with n rows and $2cn$ columns and initialize M with all zeros. Each row

represents a tag's authentication state comprised of cn bit pairs.

3) Choose random c bit pairs, $b_1, \ldots, b_c$, from the set of cn pairs (i.e., $b_i \in [1, cn]$). In the first row set the first bit value of each chosen pair equal to 1, i.e. $M_{1,2b_i-1} = 1$ for $i = 1, \ldots, c$. Remove the previously chosen $b_1, \ldots, b_c$ from the set, select another c pairs from the remaining $c(n-1)$ pairs and set the corresponding bit values in row 2 to 1 in a similar way as before. Repeat until all n rows contain c collisions. For instance, in the case of $n = 4$ and $c = 2$, the authentication states are initialized as the following:

	Tag states	Choosing bit pairs
s_1	1000000000001000	1, 7 of $(1, 2, 3, 4, 5, 6, 7, 8)$
s_2	0010100000000000	2, 3 of $(2, 3, 4, 5, 6, 8)$
s_3	0000001000100000	4, 6 of $(4, 5, 6, 8)$
s_4	0000000010000010	5, 8 of $(5, 8)$
S	$x0x0x0x0x0x0x0x0$	

4) Load each tag with the group ID ID_g, group key k_g and set the sequence counter seq.
5) Finally, the sender executes the group authentication protocol with the group of tags to randomize the values of the non-collision bits and the positions of the bit collisions before shipment.

State permutation function f_2: f_2 should satisfy the property: $f_2(s_1) \between f_2(s_2) \between \ldots \between f_2(s_n) = f_2(S)$. We create a permutation function with bit operations such as XOR, shift and bit swap. A shift operation rotates the entire authentication state clockwise in unit of a pair (i.e., bits wrap around). A bit swap exchanges the two bit values in a pair. Note that neither the shift nor the bit swap operations will occur a change on the number of bit collisions or shuffle the relationship between tags and their contribution bits (i.e., which tag contributes to which specific bit collision). However, they will effect the positions of these bit collisions. An XOR operation XORs the authentication state with a pseudorandom string and thus will not change the number or positions of the bit collisions, but it may effect the bit values that contribute to the collision, i.e. if a, b, c are binary bits, then if $a \neq b$, $(a \oplus c) \between (b \oplus c) = x$. This property always holds because if $a \neq b$ then $(a \oplus c) \neq (b \oplus c)$, which results in $(a \oplus c) \between (b \oplus c) = x$.

In the proposed group authentication protocol, one of the inputs for f_2 is the output of a keyed pseudo-random function f_1. Considering the case of a group with n tags each of which contributes c collisions (resulting in cn bit pairs), the authentication state is able to be shifted by x places where $x \in [1, cn-1]$. To define f_2, the input of f_2 can thus be parsed into three bit strings, i.e., XOR string ($2cn$ bits), bit swap string (cn bits) and shift string ($\lceil \log_2(cn) \rceil$ bits). As a consequence, the input of f_2 (i.e., the output of f_1) should be a bit string with the length of $3cn + \lceil \log_2(cn) \rceil$. If a single output of f_1 is too short, we can recursively generate a longer pseudo-random string by concatenating the previous output with a new one created with the same function taking the previous output as the input.

We elaborate the work principle of f_2 by giving the following example, where there are four tags in the group and each tag contributes to one bit collision. The initial authentication states are as follows:

s_1	01000011
s_2	01001001
s_3	01100001
s_4	11000001
S_{old}	$x1x0x0x1$

The pseudo-random function yields a bit string:

$$\underbrace{0101}_{\text{Swap}}\ \underbrace{01}_{\text{Rotate}}\ \underbrace{10100101}_{\text{XOR}}$$

Thus, we need to swap bit pairs 2 and 4, rotate right by one bit pair and XOR the result with 10100101.

	$Swap$	$Shift$	XOR
s_1	01000011	11010000	01110101
s_2	01001010	10010010	00110111
s_3	01010010	10010100	00110001
s_4	11000010	10110000	00010101
S_{old}	$x10xx01x$	$1xx10xx0$	$0xx10xx1$

Obviously, we can judge that $S_{\text{new}} = f_2(S_{\text{old}}) = f_2(s_1) \between f_2(s_2) \between f_2(s_3) \between f_2(s_4)$=$0xx10xx1$.

4.2 The Extended Protocol

The basic protocol guarantees the security but not privacy of the group/tags. Indeed, an attacker can record the group identity ID_g and thus trace the group easily. We now extend the basic protocol to provide the group privacy. We essentially employ a pseudo anonymous index of a group and update it once after a successful session.

The extended protocol is shown in Figure 6. To keep the privacy of the group, each tag stores an index I of the group identity ID_g. The index I is initialized to be the hash output of ID_g and k_g, i.e. $h(ID_g, k_g)$. The verifier stores two indices for each group, i.e., TID and TID', which are initialized to be $h(ID_g, k_g)$ as well. The purpose of using two indices in the verifier side is to prevent the desynchronization attack. TID stores the old value of the index while TID' stores the current value.

1) Like the basic protocol, the extended protocol starts by the reader transmitting a *Start* command.
2) All the tags simultaneously respond with I, seq and n. The reader relays the received composite information to the verifier.
3) The verifier searches for the corresponding group from the database according to the index I. Note that since the verifier stores indices in the database, it is very easy to match the group with $O(1)$ computation. If I cannot be found neither in the TID column nor in the TID' column, the verifier rejects and stops the protocol. Otherwise, if $I == TID'$, then the verifier updates TID and TID' to synchronize with the group tags. Meanwhile, the verifier creates a random string r and computes $m_1 = f_1(r, seq, k_g)$. It then transmits the bit string r and m_1 to the reader which relays the information to the tags.

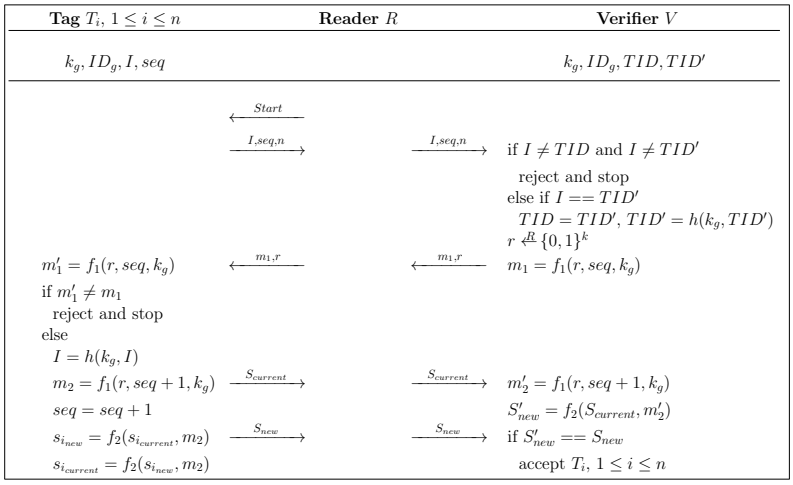


Fig. 6. The proposed extended group-authentication protocol with privacy

4) The same as the basic protocol.
5) All the tags update the index I and the sequence number seq with the same way, and generate their new individual authentication state $s_{i_{\text{new}}}$. Those individual authentication state are transmitted to the reader simultaneously, which will raise a new composite group authentication state S_{new}.
6) The same as the basic protocol.
7) All the tags update the value of $s_{i_{\text{current}}}$ to be $f_2(s_{i_{\text{new}}}, m_2)$.

4.3 Security Analysis

With the proposed protocol, the attacker's behavior of only stealing tags from a group can be easily detected since that will cause the change of the composite group authentication state. Therefore, the attacker need to replace rather than only steal tags in a group. To avoid being detected by the verifier, the attacker has to send a proper response at the same time when other tags transmit theirs, and has to make sure the composite group response is correct. According to the attacker's capabilities, we investigate five attack cases. In each case the success probability p_{a} of the attack will be analyzed in details as follows.

4.3.1 Case 1: Attacker Using a Simple Tag

In this case, the replacement tag set by the attacker performs as a normal tag. Although the attacker can obtain the composite state S_{current}, this does not help it much to compute S_{new} since it still needs to guess the positions and values of the bit collisions that the replacement tag contributes to, as well as the values of the updated non-collision bits. Otherwise, either incorrect contribution to bit collisions or incorrect non-collision values transmitted by the attacker's replacement tag will incur extra collisions. In order to pass the authentication, the attacker has to guess the correct bit values in all positions where the rest of tags do not cause bit collisions. Considering the case of a group with 4 tags and each tag contributes 2 collisions, thus the authentication state is 16-bits and is supposed to contain 8 collision bits. Since the authentication state is indistinguishable from a random string, the probability that the attacker guesses the correct authentication state of the replaced tag is $\frac{1}{2^{16}}$. Nevertheless, for the collisions caused by other tags, the attacker does not need to transmit correct bits since whatever it transmits on those positions has no affects on the result (collision caused by other tags). Therefore, the attacker only needs to guess correctly the 2 bits that actually contribute bit collisions and the 8 bits that are non-collisions need to be correct. For the remaining 6 bits, the attacker would always be successful no mater what it transmits, which means that of the 2^{16} possible bit permutations 2^6 would contribute to the right result. As a consequence, the attacker's success probability of guessing a tag authentication state s_i' which will result in the correct group authentication state S_{new} becomes $\frac{1}{2^{10}}$. In general, if the attack replaces $n_{\text{a}} \le n-2$ tags from a group of n tags, and each tag contributes c collisions, then the attacker's success probability is:

$$p_{\text{a}} = \left(\frac{1}{2}\right)^{c(n+n_{\text{a}})} \tag{1}$$

Specifically, if the attacker replaces $n_{\text{a}} = n - 1$ tags, the attack probability becomes

$$p_{\text{a}} = \left(\frac{1}{2}\right)^{2cn} \tag{2}$$

since the single legitimate tag left cannot contribute any collisions by itself and therefore the attacker would need to guess all the bit values correctly.

4.3.2 Case 2: Attacker Using a Quiet Tag

In this case, the replacement tag set by the attacker does not have to perform adhering to the protocol. Namely, the attacker guesses the bit positions where the replacement tag is supposed to occur bit collisions and then only sends the guessed bit values in those positions. For the other positions, the replacement tag stays quiet. Note that in this scenario, the attacker has no clue of what the remaining tags are transmitting. With this strategy, the attacker does not bother to transmit correct non-collision bit values which helps the attacker avoiding occurring extra bit collisions. In

order to pass the authentication, the attacker should guess the correct bit positions as well as the correct bit values which will cause collisions. Consider the same example as in Case 1 ($n = 4$, $c = 2$), and assume that the group was initialized as described in Section 4.1.1. The attacker first guesses which bit pairs contain a collision caused by the replacement tag, i.e., choose from $\binom{8}{2}$ possible position permutations for the two pairs of interest, where $\binom{8}{2}$ is the binomial function $\frac{8!}{(8-2)!(2)!} = \frac{8 \cdot 7}{2 \cdot 1}$. In addition, the attacker also has to guess the correct position and bit value which occurs a collision within each chosen pair, the probability of which turns out to be $\left(\frac{1}{2}\right)^2$. As a consequence, the attacker's success probability becomes $\left(\frac{2 \cdot 1}{8 \cdot 7}\right) \cdot \left(\frac{1}{2}\right)^4$. In general, if the attack replaces $n_{\mathrm{a}} \leq n - 2$ tags from a group of n tags, and each tag contributes c collisions, then the attacker's success probability is:

$$p_{\mathrm{a}} = \begin{pmatrix} cn \\ cn_{\mathrm{a}} \end{pmatrix}^{-1} \cdot \left(\frac{1}{2}\right)^{2cn_{\mathrm{a}}} \tag{3}$$

If the attacker replaces $n_{\mathrm{a}} = n - 1$ tags, then it has to make one collision in every bit pair because the single remaining legitimate tag is not able to cause any bit collision on its own. Therefore, the success probability of the attacker in this case is identical to that in Equation 2.

4.3.3 Case 3: Attacker Using a Smart Tag

In this case, the replacement tag set by the attacker does not have to perform adhering to the protocol, either. Moreover, the attacker can see what bit values are transmitted by the other tags. The attacker is also assumed to have the ability to determine the bit value and prepare its response right at the start of the bit period, even though it may not be practical since typical RF receivers integrate or sample over the entire bit period. With this capability, the attacker can first observe what bit values transmitted by the other tags and then just transmits the alternative value in order to make a bit collision in that position. In addition, if the attacker observes a bit collision, then it does not need to send anything out, which obviously help it guess the bit positions where it should occur collisions. Similar with Case 2, the attacker does not need to guess the non-collision bit values since it could either keep quiet or simply observe correct values from the other tags.

Consider the same example as in Case 1 ($n = 4, c = 2$), and assume that the group was initialized as described in Section 4.1.1. The attacker observes the first position of each bit pairs. As long as the first bit has been made a collision by the other tags, it can keep quiet or just send the same bit value as what the other tags transmit without occurring extra collisions. If f_1 is a secure pseudorandom function, then half of the 6 collisions contributed by the other tags would be in the first bit position of the pair on average. Thus, the attacker only needs to select c (2) bit pairs to which it will contribute collisions from 8-3=5 bit pairs instead of 8, namely, it chooses from $\binom{5}{2} = \frac{5 \cdot 4}{2 \cdot 1}$ possible position permutations rather than $\binom{8}{2}$. Nevertheless, it still has to determine whether to pose a bit collision in the first or second position within the rest of pairs, with a probability of $\frac{1}{2}$ to guess the correct value and position that occurs a collision in each chosen pair. The attacker's success probability thus becomes $\left(\frac{2 \cdot 1}{5 \cdot 4}\right) \cdot \left(\frac{1}{2}\right)^2$. In general, if the attack replaces $n_{\mathrm{a}} \leq n - 2$ tags from a group of n tags, and each tag contributes c collisions, then the attacker's success probability is:

$$p_{\mathrm{a}} = \begin{pmatrix} (cn + cn_{\mathrm{a}})/2 \\ cn_{\mathrm{a}} \end{pmatrix}^{-1} \cdot \left(\frac{1}{2}\right)^{cn_{\mathrm{a}}} \tag{4}$$

In case that $cn + cn_{\mathrm{a}}$ is an odd number, we can round down it to give the attacker more chances to succeed. If the attacker replaces $n_{\mathrm{a}} = n - 1$ tags, then it has to make one collision in every bit pair, but it can still see the response of the remaining legitimate tag which can help it choose correct bit values in collision positions (i.e., simply transmitting the alternative bit). Thus, the attacker's success probability becomes

$$p_{\mathrm{a}} = \left(\frac{1}{2}\right)^{cn} \tag{5}$$

4.3.4 Case 4: Attacker Knowing Tag States

In this case we allow the attacker to observe individual authentication state transmitted by the tags. Namely, it can observe the current authentication state s_i of each tag upon transmitted, which may help it determine the collision positions that shall be caused by the tag it replaces through comparing its state with the other tags' states. If the other tags transmit the same bit value while the replaced tag transmits a distinct value, the attacker can determine that this collision position is contributed by the replaced tag. Actually, it is the worst case since this is unlikely to happen during normal operation of the system. In practice, the attacker can only observe the composite authentication state S which as a whole does not leak much information about individual tag authentication states.

With this capability, in the scenario of Case 1, i.e., using a simple tag adhering to the protocol, the attacker earns no benefits from the knowledge of the current individual authenticate state, since it still has no clue of how to generate the new individual authentication state that it has to transmit in order to pass the authentication. Therefore, its success probability is the same as Equation 1. However,in case 2 and 3, the attacker's success probability increases. The attacker only needs to guess how to rotate and swap the authentication state, and the positions where the collisions should be caused in the new group authentication state. Consider that the attacker replaces $n_{\mathrm{a}} \leq n - 2$ tags from a group of n tags, and each tag contributes c collisions. As to an attacker using a smart tag case, the success probability is

$$p_{\mathrm{a}} = \frac{1}{cn} \cdot \left(\frac{1}{2}\right)^{cn_{\mathrm{a}}} \tag{6}$$

As to an attacker using a quiet tag, it also needs to guess the bit values that will cause collisions, and thus success probability becomes:

$$p_{\mathrm{a}} = \frac{1}{cn} \cdot \left(\frac{1}{2}\right)^{2cn_{\mathrm{a}}} \tag{7}$$

In case that the attacker removes $n_{\mathrm{a}} = n - 1$ tags, the success probability is the same as previously described for this scenario in Case 2 and Case 3.

4.3.5 Case 5: Attacker Creating a New Group

In this case, the attacker tries to forge a whole group, i.e., replacing all the tags in the group. Intuitively, this is the most difficult attack since the attacker must guess all the correct collision positions as well as the values of non-collision positions. As a result, considering a group of n tags, and each tag contributes c collisions, the attacker's success probability is:

$$p_{\mathrm{a}} = \left(\frac{1}{2}\right)^{2cn} \tag{8}$$

Note that the above five attack cases have covered the most effective attack strategies that aim to replace or impersonate tags. We have considered adversaries with increasing capabilities, for example, from the passive attacker who only impersonates a simple tag without any useful additional information (the case 1 and case 5) to the most strong attacker who can even observe individual authentication state transmitted by the tags (the case 4). Thus, in defining these 5 cases, we already go further than related works (probabilistic tag detection/verification schemes). Usually only Case 1 is covered and there are now some works that start to consider Case 2 (whether a fake tag might have more chance to hide if elects to stay quiet sometimes).

4.3.6 Privacy

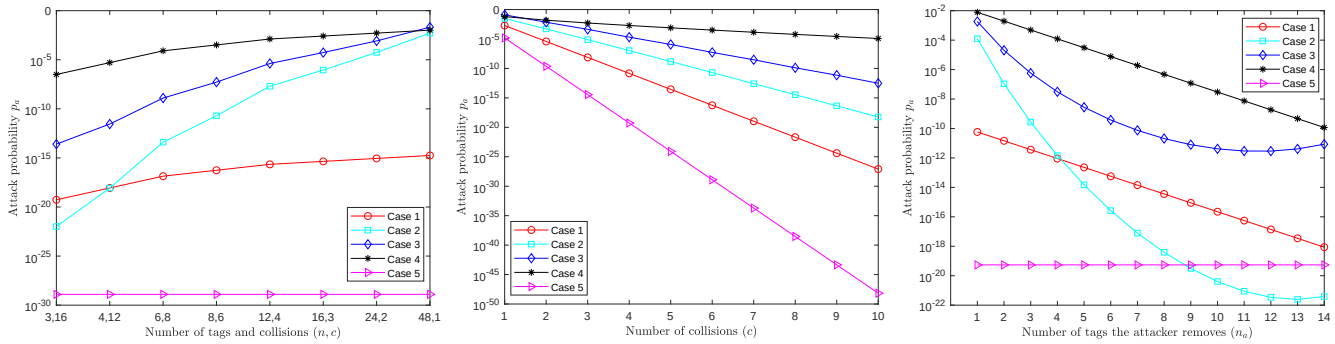
The extended protocol provides the group privacy. Indeed, the index I will be updated after each successful session and the attacker cannot trace the group by the index. In the basic protocol, the attacker could trace the group by another way, that is, it can record S_{new} of the current session and then trace the group by comparing whether $S_{\mathrm{new}} == S'_{\mathrm{current}}$, where S'_{current} is the composite state transmitted by the group tags in next session. However, in the extended protocol, this attack does not work, since each tag updates $s_{i_{\mathrm{current}}}$ after a successful session, which means in next session the first state will not be the same as the last state transmitted in the current session. Thus, if f_1 and f_2 are secure, then the privacy is preserved. The security of f_2 depends on m_2 which is an output of the function f_1. That is, if f_1 is secure then f_2 is secure, so is the protocol.

5 Evaluation and Comparison of Existing RFID Group Authentication Protocols

We first evaluate the security of the proposed schemes. As shown in Figure 7, the attacker's success probability to replace/impersonate tags is throughly evaluated. Figures 7(a), 7(b) and 7(c) show the relationship between the attacker's success probability and the size of the group n, the number of collisions that each tag contributes, and the number of tags replaced by the attacker, respectively, among which we evaluate the success probability in Case 4 using Equation 6 because it represents the attacker's best strategy. From these figures, we observe that either the more collisions each tag contributes or the more tags to be replaced, the more difficult for the attacker to succeed, which is in consistent with our intuitive thought. The attackers success rate, especially for the advanced attack in case 4, could be $\frac{1}{10^5}$ for some selected parameters. However, it does not mean the attack would be practical to our protocol. In particular, the attacker succeeds only if the attacker has the opportunity to try more than one option till he finds a guess that works. Fortunately, the proposed protocol does not allow him that luxury. The attacker has only one chance to guess a tag's state in each protocol run. The next protocol run uses a different seq and r so it is not simply a case of trying the next guess (as by now the state could be totally different). It is similar to a PIN - in theory an attacker has a $\frac{1}{10^4}$ chance of guessing a 4-digit numeric pin as the total combinations are $10 \times 10 \times 10 \times 10$. However, as the number of attempts is limited to 3, it cannot be practically guessed with brute force. In our scheme if the attacker knows a tags state he could try and guess the next state - if he gets it wrong the protocol fails. He cannot then rerun the protocol under the same conditions (same seq and r), with all the other tags still containing the previous state. Therefore, our protocol is good to prevent all these five attacks in practice.

Then, we compare our proposed scheme with existing authentication protocols which aim to achieve batch authentication from several aspects: time cost of authenticating the whole group of tags, the whole communication cost and privacy protection for the tags. The time cost is computed as the number of rounds required to authenticate the group of tags multiplied by the time cost of each round where we follow the definition of a round as the exchange of a verifier message and relative tag response from [32]. The communication cost is denoted by the number of exchanged messages including the information sent from both the verifier and the group of tags.

As shown in Table 1, we choose existing typical different types of protocols to compare. A grouping proof protocol is essentially like a per-tag authentication, with each tag waiting for last tag's output to generate its own proof. Both the time and the communication cost are the same as per-tag authentication, i.e., $n \times (t_c + t_r)$ and $n \times (c + r)$ respectively. In tree-based protocols, in each round, the verifier sends a query and receiving multiple tags' responses at the same time and thus the time cost of each round is $t_c + t_r$. While it has been proved that the tree-based protocol requires at least $2.66n$ rounds to authenticate n tags, the total cost of time is thus $2.66n \times (t_c + t_r)$. Aloha-based protocol needs $t_c + s \times t_r$ time in each round, where s is the number of slots in a frame. Compared with tree-based protocol, aloha-based protocol needs less rounds but each round needs more time. BSTSA is the combination of tree-based and aloha-based protocol, with the strengths of both. It does not need to know the number of tags n a priori to achieve maximum efficiency, but this is not a concern in our paper, since we assume that we already know the number of tags in advance. Chen *et al.*'s protocol considers a reader sends a query to the group tags and collects all the responses from the tag to generate an aggregated response. However, they do not mention how the group tags' responses are collected simultaneously without collisions. Their protocol can adopt either tree-based or aloha-based technique to reduce the collisions. Therefore, the time cost of their protocol is $O(n) \times (t_c + t_r)$. Finally, in our protocol, the reader sends a query and all the tags send individual responses at the same time. These individual responses will turn into a combined response,



(a) Replace a single tag ($n_a = 1$) in the case of a constant length of the authentication state $2cn = 96$.

(b) Replace a single tag ($n_a = 1$) in the case of $n = 8$ with different number of collisions that each tag contributes.

(c) Replace multiple tags from a group with 16 tags and each tag contributes 2 bit collisions $(n = 16, c = 2)$.

Fig. 7. The attacker's success probability to replace/impersonate tags.

TABLE 1
Comparison of RFID Group Authentication Protocols.

Protocols	Time Cost	Communication Cost	Tag Privacy
Grouping Proof [7], [9]	$n \times (t_c + t_r)$	$n \times (c + r)$	$\surd$
Tree-based [31], [32]	$2.66n \times (t_c + t_r)$	$O(n \log n) \times (c + r)$	$\times$
Aloha-based [30], [32]	$1.22n \times (t_c + s \times t_r)$	$O(n) \times (c + r)$	$\times$
BSTSA [32]	$1.25n \times (t_c + 4.4s \times t_r)$	$O(n) \times (c + r)$	$\times$
Chen *et al.* [12]	$O(n) \times (t_c + t_r)$	$O(n) \times (c + r)$	$\surd$
Dhailah *et al.* [22]	$n \times (t_c + t_r)$	$n \times (c + r)$	$\times$
Ours	$t_c + t_r$	$c + r$	$\surd$

n is the number of tags in a group.
t_c, t_r are the time cost of sending a challenge/query and a response.
s is the length of the frame used in Aloha based protocol, namely, the number of time slots in a frame.
c, r are the length of a challenge and a response respectively.

which reduces the communication cost to $c + r$. This makes our protocol much more efficient than existing protocols not only in time cost but also in communication cost.

In addition, we made experiments on the UMich MOO design [47], which is based on Intel's Wireless Identification and Sensing Platform (WISP) UHF tags. Specifically, the MOO is a passive UHF device that is composed of an MSP430 micro-controller unit (MCU), an ultra-low power MCU with a 16-bit instruction set, which operates from a 2-volt supply with an internal clock of 1.075 MHz. We realize the function f with a 128-bit-key Advanced Encryption Standard (AES) encryption algorithm, while the RFID tag has an identity with length of 96 bits. The tested running time of an encryption is about 12ms and it takes the tag about 40ms and 60ms to run the basic and extended protocols, respectively, which is very efficient.

6 Practical Considerations

There are a number of practical aspects that should be taken into account when implementing the proposed scheme, which in turn affect possible application scenarios. These aspects include the system architecture, the size of the group that is to be authenticated and the RFID technology used.

RFID is used in a variety of systems, which have different operational objectives. Even if only inventory or supply chain management systems are considered there are still a number of divergent system architectures. In some systems the items are tracked in a large area and maximum operating range is a key objective. In such cases multiple readers are often used to increase reliability in detecting all the tokens within the system's coverage, i.e. multiple readers cover the same area and one reader might detect a token that another has missed. As mentioned in Section 4, the scheme proposed in this paper is intended to operate in a system where a single reader communicates with a group, because the reader has to receive the replies from all the tokens to identify the bit collisions. The proposed scheme will therefore not be practical in a system where one token in a group is read by one reader and another token is read by another reader. However, not all inventory management systems use such an architecture. Some systems instead require to identify items that are located within a small area. For example, a bunch of tagged items move on a conveyor belt. This architecture is also more likely to be used in applications that identify items within a specific container or package, i.e. in systems where items are grouped and could benefit from the proposed scheme. In such systems, token data is aggregated by a single reader, which means that bit collisions could be reliably identified.

In Section 4.1.1 the length of the authentication state is stated to be $2cn$, with c being the number of collisions contributed by each tag and n being the number of tags in the group. In practice, the length of the token's authentication state is restricted by the tag's storage and computation capability. The number of tokens making up a group, with the collisions per token determined by a chosen target attack probability, is therefore limited. In addition, the number of tokens that could be read at the same time by a single reader is constrained by some practical issues, e.g., the receiver architecture of the reader and the reader's communication range.

Therefore, the proposed scheme is more practical in a system that needs to authenticate a large number of groups containing a limited number of tags, in which case it could increase throughput without placing additional burden on the reader or token technology, or achieve the same throughput with simpler and therefore less expensive equipment. For example, a pharmaceutical system could be monitoring containers containing individually tagged blister packs. Under normal circumstances a system that deals with 120 containers per minute (each container contains 15 blister

packs) needs to finish 1800 authentication operations within one minute, which means the reader and each tag has to run the authentication procedure in approximately 33 ms. As a comparison, our proposed protocol only requires the system to run 120 authentication operations per minute, which gives the reader and each tag 500 ms to finish the authentication procedure. Alternatively, the system throughput could be increased to monitor 1800 containers every minute using the same reader and tokens as before. This simple example illustrates that a group does not necessarily have to contain a large number of tags to achieve practical benefits from the proposed scheme.

A number of RFID technologies use the general principle of communication collisions to check if multiple tokens are present during the token selection process. However, as explained in Section 2 the back-channel coding (Manchester) defined in the ISO 14443 and ISO 15693 HF RFID standards can be used without modification to implement our scheme, as any bit period can be distinctly classified as a '1', a '0', or a collision. Although UHF technology is often associated with inventory management there are also numerous such systems utilising HF technology. ISO 15693 tokens are often used for this application, e.g. NXP I-Code and TI Tag-It tokens, and such an ISO 15693 system could achieve an operating range up to 1 m which is comparable to 'near-field' UHF tokens used for item-level tagging in controlled read zone applications. It is therefore feasible that the proposed scheme could be deployed in inventory management systems using existing technology.

The proposed scheme depends on the tokens' capability to generate deterministic bit collisions, by transmitting the calculated token authentication states as described in Section 4.1.1, and the reader's capability to reliably detect such collisions. To test these aspects we implemented a proof-of-concept system using a HF token testbed (ISO 14443/15693) we constructed, shown in Figure 8, and an off-the-shelf multi-ISO HF RFID reader build around a NXP CL RC632 contactless reader IC. The reader IC implements a correlation receiver, basically integrating over half the bit period and comparing the output to a fixed threshold, and could be configured to output intermediate signals within the demodulation/decoding process. We could therefore monitor the output after the correlation step to see if a collision was detected, i.e. if there are sufficient peaks in both halves of the bit period.

Figure 9 shows some example output traces from the reader IC. The HF token testbed implements the physical communication layers as specified in the related standard, does clock recovery based on the reader's carrier and contains a 8-bit micro-controller to implement additional functions. It was relatively simple to synchronise multiple tokens' responses to a common 'bit grid' using a data clock signal derived from the reader's carrier and a period counter. The counter was started at the end of a transmission from the reader. This is to be expected taken that this level of synchronisation is already used in current RFID systems, e.g. ISO 14443, for anti-collision. Multiple testbed 'tokens' could therefore be made to participate in the protocol and these were able to construct a verifiable group authentication state by synchronously replying with their authentication state.

We also experimented with groups of different sizes

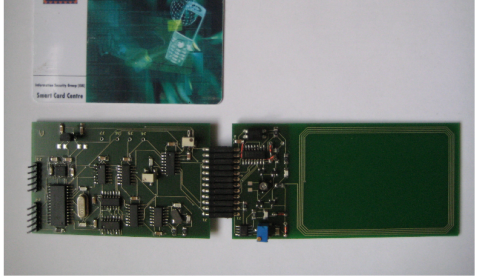


Fig. 8. Constructed HF token emulator

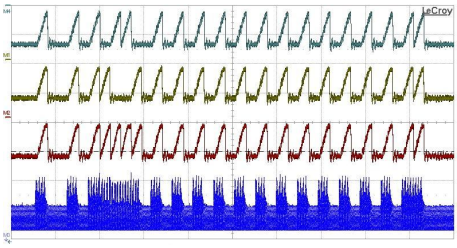


Fig. 9. Intermediate receiver output after correlation step showing a bit collision: 1) bit sequence 1, 2) bit sequence 2, 3) bit sequence 1+2, and 4) the combined bit sequence 1+2 as modulated onto the HF carrier

consisting of x off-the-shelf tokens transmitting the same sequence and 1 off-the-shelf token causing a single bit error. The reader we used allowed for a maximum group size of 11 (10+1), after which the collision was no longer reliably detected. Beyond this threshold the decreased carrier amplitude, resulting from the number of tokens drawing power from the reader and thereby acting as a load on the carrier, effected the operation of the receiver in such a way that the collision was no longer detected. A more powerful reader, with a different receiver architecture might allow for a larger group, although a group of size 11 could still be useful and fits within the practical application scenarios described earlier in this section.

7 Conclusion

In this article a group authentication protocol for RFID-based secure tracking systems is presented. Specially, the proposed protocol enables a verifier to authenticate a group of tags simultaneously based on controlled bit-collision patterns. All the tags can send their individual responses at the same time which results in a composite group response. As a consequence, the verifier can therefore authenticate a group of tags within a limited time period that is comparable to the time taken to perform a single challenge-response authentication. With this advantage, the transaction time to deal with a group of tagged items by a reader is significantly decreased, and thus is quite suitable for practical RFID-based tracking systems in which the available transaction time is constrained due to high tag throughput. The presented protocol employs only some pseudorandom functions and bit operations, so that the computation complexity especially in the tag side is very low which is comparable to the cryptographic primitives required by most grouping proofs and lightweight authentication protocols proposed for the RFID environment. We then give a thorough security analysis of the proposed protocol, where in particular we investigated the attack success probability under five prominent attack scenarios and demonstrated that it is possible

to obtain a low attack probability when appropriate n and c are chosen. In addition, we extend the basic protocol to preserve the privacy for tags. Future work could investigate the possibility of new or existing UHF PHY/MAC layers supporting this scheme, as well as the design of a custom reader to possibly allow for more tokens to be part of the group.

ACKNOWLEDGMENTS

Jian Weng was partially supported by the National Key Research and Development Plan of China under Grant Nos. 2020YFB1005600, National Natural Science Foundation of China (Grant Nos. 61825203, U1736203, 61732021), Major Program of Guangdong Basic and Applied Research Project under Grant No. 2019B030302008, and Guangdong Provincial Science and Technology Project under Grant No. 2017B010111005. Anjia Yang was partially supported by National Natural Science Foundation of China (Grant No. 62072215, 61702222, 61877029, 61932010, 61932011). Gerhard Hancke was supported by the Research Grants Council of Hong Kong under Project CityU 21204716. Xizhao Luo was supported by the National Natural Science Foundation of China under Grant No. 61972454 and Natural Science Foundation of Jiangsu Province under Grant No. BK20201405.

REFERENCES

[1] C. Su, B. Santoso, Y. Li, R. H. Deng, and X. Huang, "Universally composable rfid mutual authentication," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, pp. 83–94, Jan 2017.

[2] P. Gope, J. Lee, and T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for rfid systems using physically unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 2831–2843, Nov 2018.

[3] M. Chen, S. Chen, and Y. Fang, "Lightweight anonymous authentication protocols for rfid systems," *IEEE/ACM Transactions on Networking*, vol. 25, pp. 1475–1488, June 2017.

[4] A. Yang, Pagnin, A. Mitrokotsa, G. P. Hancke, and D. S. Wong, "Two-hop distance-bounding protocols: Keep your friends close," *IEEE Transactions on Mobile Computing*, vol. 17, pp. 1723–1736, July 2018.

[5] A. Juels, ""yoking-proofs" for rfid tags," in *IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pp. 138–143, March 2004.

[6] Y. Lien, X. Leng, K. Mayes, and J.-H. Chiu, "Reading order independent grouping proof for rfid tags," in *IEEE International Conference on Intelligence and Security Informatics*, pp. 128–136, June 2008.

[7] S. Sundaresan, R. Doss, S. Piramuthu, and W. Zhou, "A robust grouping proof protocol for rfid epc c1g2 tags," *IEEE Transactions on Information Forensics and Security*, vol. 9, pp. 961–975, June 2014.

[8] Z. Zhou, P. Liu, Q. Liu, and G. Wang, "An anonymous offline rfid grouping-proof protocol," *Future Internet*, vol. 10, no. 1, pp. 1–15, 2018.

[9] B. Yuan and J. Liu, "A universally composable secure grouping-proof protocol for rfid tags," *Concurrency and Computation: Practice and Experience*, vol. 28, pp. 1872–1883, 2016.

[10] S. Abughazalah, K. Markantonakis, and K. Mayes, "Two rounds rfid grouping-proof protocol," in *2016 IEEE International Conference on RFID*, pp. 1–14, May 2016.

[11] H. Liu, H. Ning, Y. Zhang, D. He, Q. Xiong, and L. T. Yang, "Grouping-proofs-based authentication protocol for distributed rfid systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, pp. 1321–1330, July 2013.

[12] J. Chen, A. Miyaji, and C. Su, "A provable secure batch authentication scheme for epcgen2 tags," in *Proceedings of 8th International Conference on Provable Security*, ProvSec'14, pp. 103–116, Springer, 2014.

[13] S. Sundaresan, R. Doss, and W. Zhou, "Zero knowledge grouping proof protocol for rfid epc c1g2 tags," *IEEE Transactions on Computers*, vol. 64, pp. 2994–3008, Oct 2015.

[14] S. Cheng, V. Varadharajan, Y. Mu, and W. Susilo, "An efficient and provably secure rfid grouping proof protocol," in *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '17, (New York, NY, USA), pp. 71:1–71:7, ACM, 2017.

[15] W. Zhang, S. Qin, S. Wang, L. Wu, and B. Yi, "A new scalable lightweight grouping proof protocol for rfid systems," *Wireless Personal Communications*, vol. 103, pp. 133–143, 2018.

[16] D.-Z. Sun and Y. Mu, "Security of grouping-proof authentication protocol for distributed rfid systems," *IEEE Wireless Communications Letters*, vol. 7, pp. 254–257, April 2018.

[17] X. Leng, G. P. Hancke, K. E. Mayes, and K. Markantonakis, "Tag group authentication using bit-collisions," in *2012 Information Security for South Africa*, pp. 1–8, Aug 2012.

[18] Y. Tian, G. Yang, and Y. Mu, "Privacy-preserving yoking proof with key exchange in the three-party setting," *Wireless Personal Communications*, vol. 94, pp. 1017–1034, 2017.

[19] D. Sun and G. Xu, "One-round provably secure yoking-proof for rfid applications," in *2017 IEEE Trustcom/BigDataSE/ICESS*, pp. 315–322, 2017.

[20] J. Shen, H. Tan, Y. Zhang, X. Sun, and Y. Xiang, "A new lightweight rfid grouping authentication protocol for multiple tags in mobile environment," *Multimedia Tools and Applications*, vol. 76, pp. 22761–22783, Nov 2017.

[21] J. M. de Fuentes, P. Peris-Lopez, J. E. Tapiador, and S. Pastrana, "Probabilistic yoking proofs for large scale iot systems," *Ad Hoc Networks*, vol. 32, pp. 43 – 52, 2015.

[22] H. A. Dhailah, E. Taqieddin, and A. Alma'aitah, "An enhanced and resource-aware rfid multitag grouping protocol," *Security and Communication Networks*, vol. 2019, pp. 1–15, 2019.

[23] "Iso/iec 14443. identification cards – contactless integrated circuit cards – proximity cards,"

[24] "Iso/iec 15693. identification cards – contactless integrated circuit cards – vicinity cards,"

[25] "Epc class-1 generation-2 uhf rfid conformance requirements specification v. 1.0.2,"

[26] M. R. Rieback, G. N. Gaydadjiev, B. Crispo, and A. S. Tanenbaum, "A platform for rfid security and privacy administration," in *Proceedings of Large Installation System Administration Conference*, pp. 89–102, 2006.

[27] R. Jin, X. Du, Z. Deng, K. Zeng, and J. Xu, "Practical secret key agreement for full-duplex near field communications," *IEEE Transactions on Mobile Computing*, vol. 15, pp. 938–951, April 2016.

[28] Y. Zhang, Y. Xiang, T. Wang, W. Wu, and J. Shen, "An over-the-air key establishment protocol using keyless cryptography," *Future Generation Computer Systems*, vol. 79, pp. 284 – 294, 2018.

[29] G. Hancke, "Modulating a noisy carrier signal for eavesdropping-resistant hf rfid," *e&i Elektrotechnik und Informationstechnik*, vol. 124, pp. 404–408, Nov 2007.

[30] N. Abramson, "The aloha system: Another alternative for computer communications," in *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, AFIPS '70, (New York, NY, USA), pp. 281–285, ACM, 1970.

[31] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory*, vol. 25, pp. 505–515, Sep 1979.

[32] T. F. L. Porta, G. Maselli, and C. Petrioli, "Anticollision protocols for single-reader rfid systems: Temporal analysis and optimization," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 267–279, Feb 2011.

[33] L. Yang, J. Han, Y. Qi, and Y. Liu, "Identification-free batch authentication for rfid tags," in *Proceedings of the The 18th IEEE International Conference on Network Protocols*, ICNP '10, (Washington, DC, USA), pp. 154–163, IEEE Computer Society, 2010.

[34] Q. Lin, L. Yang, and Y. Guo, "Proactive batch authentication: Fishing counterfeit rfid tags in muddy waters," *IEEE Internet of Things Journal*, vol. 6, pp. 568–579, Feb 2019.

[35] E. Pagnin, A. Yang, Q. Hu, G. Hancke, and A. Mitrokotsa, "Hb+db: Distance bounding meets human based authentication," *Future Generation Computer Systems*, vol. 80, pp. 627–639, 2018.

[36] Y. Zhang, C. Xu, H. Li, K. Yang, N. Cheng, and X. Shen, "Protect: Efficient password-based threshold single-sign-on authentication for mobile users against perpetual leakage," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[37] S. Wang, J. Wang, and Z. Yu, "Privacy-preserving authentication in wireless iot: Applications, approaches, and challenges," *IEEE Wireless Communications*, vol. 25, pp. 60–67, December 2018.

[38] A. Yang, J. Xu, J. Weng, J. Zhou, and D. S. Wong, "Lightweight and privacy-preserving delegatable proofs of storage with data dynamics in cloud storage," *IEEE Transactions on Cloud Computing, DOI: 10.1109/TCC.2018.2851256*, to appear.

[39] Q. Hu, L. M. Dinca, A. Yang, and G. Hancke, "Practical limitation of co-operative rfid jamming methods in environments without accurate signal synchronization," *Computer Networks*, vol. 105, pp. 224 – 236, 2016.

[40] Y. Zhang, C. Xu, C. Nan, H. Li, H. Yang, and X. Shen, "Chronos+: An accurate blockchain-based time-stamping scheme for cloud storage," *IEEE Trans. Services Computing*, vol. 13, no. 2, pp. 216–229, 2020.

[41] J. Liang, Z. Qin, S. Xiao, J. Zhang, H. Yin, and K. Li, "Privacy-preserving range query over multi-source electronic health records in public clouds," *Journal of Parallel and Distributed Computing*, vol. 135, p. 127139, 2020.

[42] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, "Privacy-preserving cloud-based road condition monitoring with source authentication in vanets," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 1779–1790, July 2019.

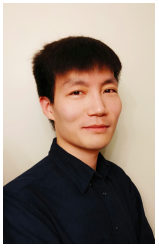
[43] A. Yang, Y. Zhuang, J. Weng, G. Hancke, D. S. Wong, and G. Yang, "Exploring relationship between indistinguishability-based and unpredictability-based rfid privacy models," *Future Generation Computer Systems*, vol. 82, pp. 315 – 326, 2018.

[44] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in internet of things with privacy preserving: Challenges, solutions and opportunities," *IEEE Network*, vol. 32, pp. 144–151, November 2018.

[45] M. L. Das, P. Kumar, and A. Martin, "Secure and privacy-preserving rfid authentication scheme for internet of things applications," *Wireless Personal Communications*, Sep 2019.

[46] Y. Zhang, C. Xu, X. Lin, and X. Shen, "Blockchain-based public integrity verification for cloud storage against procrastinating auditors," *IEEE Trans. Cloud Computing*, pp. 1–15, accepted 2019, to appear. doi: 10.1109/TCC.2019.2908400.

[47] UMich MOO, "A batteryless programmable rfid-scale sensor device," 2014.



Anjia Yang (M'17) received the B.S. degree from Jilin University in 2011 and the Ph.D. degree from the City University of Hong Kong in 2015. He is currently an associate professor in Jinan University, Guangzhou. His research interests include security and privacy in vehicular networks, internet of things, blockchain and cloud computing.

Dutliff Boshoff is an undergraduate student with the Department of Electrical Engineering at City University of Hong Kong. His research interests are system security, physical-layer security and data science for the Internet-of-Things..

Qiao Hu received his B.S. degree from Hunan University in 2011 and his M.Sc degree from Wuhan University in 2013. He obtained his Ph.D. degree from the Department of Computer Science, City University of Hong Kong in 2017. Now he works as an assistant professor in Hunan University. His research interests include RFID security, sensor security and wireless communication security.

Gerhard P. Hancke (S'99-M'07-SM'11) obtained B.Eng. and M. Eng. degrees from the University of Pretoria (South Africa) in 2002 and 2003, and a PhD in Computer Science with the Security Group at the University of Cambridges Computer Laboratory in 2009. He is an Associate Professor with the City University of Hong Kong. Dr. Hanckes research interests are system security, embedded platforms and distributed sensing applications.

Xizhao Luo (M'20) received the B.S. and M.S. degrees from the Xian University of Technology, Xian, China, in 2000 and 2003, respectively, and the Ph.D. degree from Soochow University, China, in 2010. He now is an associate professor at the School of Computer Science and Technology, Soochow University, China. He held a post-doctoral position with the Center of Cryptography and Code, School of Mathematical Science, Soochow University. His main fields of interest are cryptography and computational complexity.

Jian Weng received the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, in 2008. From 2008 to 2010, he held a post-doctoral position with the School of Information Systems, Singapore Management University. He is currently a Professor and the Dean with the College of Information Science and Technology, Jinan University. His research interests include public key cryptography, cloud security, blockchain, etc. He has published over 100 papers in cryptography and security conferences and journals, such as CRYPTO, EUROCRYPT, ASIACRYPT, TCC, PKC, TPAMI, TIFS, and TDSC. He served as a PC co-chairs or PC member for more than 30 international conferences. He also serves as associate editor of IEEE Transactions on Vehicular Technology.

Keith Mayes is a Professor within the Information Security Group at Royal Holloway University of London (UK). His research interests include system, device and communications security, with emphasis on smart cards, RFIDs and embedded security modules. He obtained his BSc and PhD degrees from the University of Bath (UK) in 1983 and 1987.

Konstantinos Markantonakis is a full Professor in Royal Holloway University of London, he is also the Director of the Information Security Group Smart Card and IoT Security Centre (SCC). He obtained his BSc. (Lancaster University), MSc., Ph.D. (London) and his MBA in International Management from Royal Holloway, University of London. His research interests include smart card security and applications, secure cryptographic protocol design, key management, embedded system security, IoT, and trusted execution environments.