Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8436917, 12 pages
https://doi.org/10.1155/2020/8436917

Research Article

The Lightweight RFID Grouping-Proof Protocols with Identity Authentication and Forward Security

Zhicai Shi, Xiaomei Zhang, and Jin Liu

School of Electronic & Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Correspondence should be addressed to Zhicai Shi; szc1964@163.com

Received 20 July 2019; Accepted 24 December 2019; Published 18 March 2020

Guest Editor: Hasan Ali Khattak

In many fields, multiple RFID tags are often combined into a group to identify an object. An RFID grouping-proof protocol is utilized to prove the simultaneous existence of a group of tags. However, many current grouping-proof protocols cannot simultaneously provide privacy preserving, forward security, and the authentication between reader/verifier and tags, which are vulnerable to trace attack, privacy leakage, and desynchronization attack. To improve the secure performance of the current grouping-proof protocols, we propose two provable lightweight grouping-proof protocols that provide forward security, identity authentication, and privacy preserving. Our protocols involve a trusted reader and an untrusted reader, respectively. In order to avoid verifying some invalid evidences, our protocols complete the authentication of the verifier to the trusted reader and the verified tags before the verifier verifies the grouping-proof evidence. Each tag uses parallel mode to complete its signature to improve the efficiency of the protocols. Moreover, the activate-sleep mechanism and the filtering operation are proposed to effectively reduce the collision probability and computing load of tags. Our protocols complete the authentication to tags twice by a verifier and a trusted reader, respectively. They can resist various attacks such as eavesdropping, replay, trace, and desynchronization. The protocols are proven to be secure, flexible, and efficient. They only utilize some lightweight operations. Therefore, they are very suitable to the low-cost RFID systems.

1. Introduction

As an important sensing method of Internet of Things (IoTs), Radio Frequency IDentification (RFID) has become a pervasive technology and it has been successfully applied to mobile payment, healthcare, supply chain management, transportation, and other fields [1]. A typical RFID deployment is called an RFID system, which has three main components: Radio Frequency (RF) tags, a reader, and a backend server. A backend server is also called a verifier. Tags are some electronic devices and they are usually used to identify some objects. Tags are usually divided into active tags and passive tags. The current popular tags are passive. They are very simple and cheap. They have no internal power source. When these tags communicate with a reader, they are powered with their on-chip antenna coil, which is activated by the RF signal from the reader. Thus, their computation and communication capabilities are very limited. A tag is usually used to identify an object. However, under many circumstances, multiple tags are combined into a group to identify several related objects or different parts of an object. Therefore, it is necessary to read several tags simultaneously and to prove their coexistence.

In 2004, Juels [2] proposed the first application of a group of tags. He combined two tags into a group to identify the container of the medication and the leaflet, respectively. The leaflet describes the side effects of the medication. He proposed a grouping-proof protocol to verify whether each container was dispensed with its leaflet. Another example is that the manufacturer of aircraft equipment uses two tags to identify a certain part and its safety cap. A grouping-proof protocol is utilized to verify whether a part leaves the factory with its safety cap. For the circumstances described above, some grouping-proof protocols have been proposed to prove the coexistence of multiple tags. Due to the hardware resource limitation of tags, the grouping-proof protocols only

use some lightweight cryptographic functions. Hence, the secure level of the current grouping-proof protocols is very limited. The majority of existing protocols do not protect the privacy of the tag and cannot provide forward security [3, 4]. Some grouping-proof protocols usually use serial signature mode so that they need more time to collect the grouping-proof evidence. In order to overcome the flaws above, we propose two novel grouping-proof protocols. These protocols only utilize some lightweight functions to ensure the security and privacy of an RFID system.

The main contributions of our work can be summarized as follows:

(1) We proposed two grouping-proof protocols. These protocols involve a reader and multiple tag groups. The reader may be trusted or untrusted. It is used to collect the grouping-proof evidence. The protocols complete both the mutual authentication between the verifier and the trusted reader and the one-way authentication of the reader/verifier to tags. One of our protocols completes the authentication to tags twice by the verifier and the trusted reader, respectively, which enhances the security level of the protocols.

(2) The protocols ensure the privacy of the RFID system by utilizing some one-way lightweight functions to generate sessions between reader and tags.

(3) The protocols provide forward security by means of secrecy updating. When the secrecy is updated, the old secrecy is reserved in the verifier so that the protocols can resist desynchronization attack.

(4) In order to reduce the collision probability and computation load of tags, a novel activate-sleep mechanism is proposed. The mechanism makes the related tags activated and other tags sleep during the grouping-proof period. When the reader communicates with tags, only the activated tags give their response. Therefore the collision probability and computation load of tags are remarkably reduced.

(5) The protocols utilize the mechanism based on MAC layer protocol of Ethernet. The message broadcasted by a reader is only received by a certain tag and other unrelated tags do not participate in the interaction of the protocols, which is called the filtering operation. Therefore our protocols use a broadcast RF channel to complete the peer-to-peer communication between a reader and a certain tag, which further reduces the computation load of tags and the collision probability between them.

The rest of this paper is organized as follows. In Section 2, we briefly review some typical grouping-proof protocols and analyze their security. In Section 3, we describe the RFID system under the grouping-proof mode and propose its security model. In Section 4, we propose two novel grouping-proof protocols by utilizing parallel communication mode, the activate-sleep mechanism, and the filtering operation. We describe the detail process of the protocols. In Section 5, we prove the security of our protocols. We analyze their security performance and compare them with some typical grouping-proof protocols. Finally, we give the conclusions in Section 6.

2. Some Typical RFID Grouping-Proof Protocols

In this section, we describe some typical and related grouping-proof protocols and discuss their security and vulnerability.

The first grouping-proof protocol is the Yoking-proofs protocol, which is proposed by Juels [2]. This protocol only involves two tags. The protocol gives a proof that a pair of tags has been scanned simultaneously. For the minimalist version of the protocol, the identifiers of the tags are transferred in plaintext. An adversary can intercept these identifiers by eavesdropping the sessions between reader and tags. Then he can get the privacy of the RFID system. Therefore the protocol cannot resist privacy leakage. Saito and Sakurai [5] and Burmester et al. [3] analyzed the Yoking-proofs protocol. They found that it does not resist replay attack and does not check the results from other tags so that some unrelated tags can join the protocol. Another weakness is that a corrupted tag can impersonate a legal tag to generate the valid evidence. Otherwise, the protocol cannot resist interleaving attack [6].

Leng et al. [7] proposed a select-response grouping-proof protocol. Instead of waiting for the computation result from the tags, their protocol allows the reader to actively select the demanded tags. Therefore their protocol can provide collision-free performance and identify the missing tags. But a malicious tag can stop a legitimate proof generation or force creating an invalid proof. So their protocol cannot resist denial of service (DoS) attack. To overcome these problems, they propose an online protocol and the verifier is involved in each step instead of waiting. Therefore, the protocol wastes the time of the verifier.

Huang and Ku [8] proposed a grouping-proof protocol conforming to the Class-1 Gen-2 standard. Their protocol is used to check the correlation of drug and patient so as to enhance medication safety. Peris-Lopez et al. [4] found that the protocol uses CRC functions. These functions are some algorithms based on polynomial arithmetic in F_2. They found that an attacker can exploit the linearity property of CRC functions, such as $\mathrm{CRC}(a \oplus b) = \mathrm{CRC}(a) \oplus \mathrm{CRC}(b)$ to get the private information of the tag. Then he can impersonate this tag in the future grouping-proof protocol. Otherwise, for the protocol proposed by Huang H-H et al., the target tag updates its *pin* once it is interrogated by an attacker. But the verifier does not know that the target tag has been interrogated and the verifier does not update its pin. Therefore the verifier and the tags own different pin and they lose their synchronization. So the protocol cannot resist desynchronization attack.

Chien et al. [9] proposed two grouping-proof protocols for the EPC C1-G2 tags. Their protocols only utilize a 16-bit pseudorandom number generator and bitwise XOR

operation. Peris-Lopez et al. [4] analyzed the online protocol and found a vulnerability. If an adversary detects that the tag and the reader generate the same random number he can generate a fixed session unrelated to the random number. Later he can use the session to impersonate a target tag. Therefore, the protocol cannot resist forgery attack and subset replay attack. To overcome the shortcoming of the online protocol, Chien et al. proposed an offline protocol. But their offline protocol cannot also resist subset replay attack. In addition, their protocol cannot be applied to some special scenarios where the number and type of tags are not known in advance.

Like the two protocols described above, Peris-Lopez et al. [10] also proposed a grouping-proof protocol to enhance medication safety. For their protocol, the unit-dose packages can automatically match the inpatient to avoid human error. Peris-Lopez et al. claimed that the digital evidence from their protocol could be used for medication tracking and auditing. But Yen et al. [11] found that only the nurse signed the evidence. If a medication dispute occurs, the hospital can counterfeit evidence. In order to overcome the security vulnerability described above, Yen et al. proposed another solution. Their protocol involves four entities: the backend server, the nurse's PDA, the inpatient's wristband, and the unit-dose drug packages. However, their protocol could not resist tracing attack. If the inpatient and the unit-dose tags receive the same challenge from the nurse's PDA many times, they will return the same message. Then an adversary can locate the inpatient and his/her unit-dose package. Therefore, it is easy to leak the privacy of the inpatient. Otherwise, the secret keys of the protocol are not updated after each grouping-proof and the protocol cannot ensure forward security.

Liu et al. [12] analyzed some previous grouping-proof protocols. They found that many protocols only involve a single reader and a group of tags. Then they adopted the distributed authentication mode to propose a grouping-proof protocol. They claimed that their protocol can resist some typical attacks such as forgery, tracking, replay, and denial of proof. Later, Shen et al. [13] proposed an enhanced protocol and claimed that their protocol could preserve the privacy of the RFID system and resist replay attack. However, we found that their protocol uses the plaintext of the identifiers for communication. Moreover, these identifiers are fixed during the grouping-proof period. Hence, their protocol cannot resist trace attack and it seriously leaks the privacy of the RFID system. The grouping-proof evidence V_i of each tag is generated independently and there is not any relationship between V_m and $V_n (m \neq n)$. Their grouping-proof protocol does not have any time limitation. So their grouping-proof evidence does not prove the coexistence of the related tags.

By analyzing some previous grouping-proof protocols, Moriyama [14] utilized parallel signature mode to propose a two-round grouping-proof protocol. The protocol only involves two round sessions. The number of the sessions is independent of the number of tags. But the protocol can only resist impersonation attack. The timestamp is generated by the reader. If it is timeout the verifier cannot judge the validness of the grouping-proof evidence.

Sundaresan et al. [15] analyzed some special requirements for a grouping-proof protocol. Then they proposed a robust grouping-proof protocol for the EPC C1-G2 tags. The protocol provides forward security. It utilizes serial signature mode to collect the grouping-proof evidence from each tag so as to degrade its efficiency. Each tag has to complete a large amount of 128-bit operations, which further reduces the efficiency of the protocol. After the i^{th} tag generates its evidence M_i, it updates its secret VT_S. If a grouping-proof collecting process is stopped or aborted the subsequent tag (e.g., the j^{th} tag, $j > i$) cannot update its secret VT_S. Thus some tags' secrets are updated and other tags' secrets are not updated. Their secrets are not synchronous. Therefore, the protocol cannot resist DoS attack. Otherwise, after a reader is only authenticated it can be authorized to complete the grouping-proof. When there are only some untrusted readers near the verifier they cannot be authorized to complete a grouping-proof.

Huang and Mu [16] proposed a grouping-proof protocol that introduced a new method of the key distribution. The protocol only utilizes some lightweight functions (not hash function) to generate the sessions so as to reduce the computing cost of tags. But the protocol updates the secret key of tags twice for each grouping-proof period. After a tag completes the first updating of its secret key c_i, the reader uses the previous c_i to generate $|c_i - a_i + r_{Ti}|$ and send the result to the tag. The tag cannot authenticate the reader because the secret keys c_i they own are different. Hence desynchronization attack occurs and the protocol cannot resist DoS attack. For the protocol, if an adversary impersonates a reader and repeats to transmit S and a random number r_R to a tag, the tag will reply the same $H(b_i \oplus r_R)$ and k_{l2}/k_{l1}. The protocol also cannot resist tracing attack. k_{l2}/k_{l1} is transferred in plaintext so that the protocol cannot preserve the privacy of the tags. Otherwise, the secret keys of the tags are stored in the reader. So the reader must be trusted. Any untrusted reader cannot be used to complete a grouping-proof.

Shen et al. [17] only used some simple bitwise operations to propose a practical grouping-proof protocol. But their protocol utilizes serial signature mode so that it takes more time to collect a grouping-proof evidence. Otherwise, an adversary can deduce the group's key and the tag's sequence number by eavesdropping the sessions. Hence the protocol cannot preserve the privacy of the system. The protocol does not update the secret keys of the system after each authentication. Therefore the protocol cannot provide forward security.

Hong-yan [18] analyzed the grouping-proof protocol proposed by Batina et al. [19] and he found the protocol has some security vulnerability. Then he utilized ECC mechanism to propose an improved grouping-proof protocol. But his protocol cannot provide forward security and it can only complete the grouping-proof for two tags. So it is not suitable for multiple tags.

Sun and Mu [20] analyzed the protocol proposed by Liu et al. [12] and found that the attacker can easily launch some attacks such as replay, forgery, tracking, and denial of proof. Although Liu et al. claimed their protocol can resist these

well-known attacks, the attacker can effectively compromise all secrets and further impersonate a legal reader or a legal tag.

Zhang et al. [21] proposed a scalable grouping-proof protocol. They use the pruning query tree to reduce the collision between tags. Their protocol supposes that the reader is trusted. Before the reader collects the grouping-proof evidence the verifier firstly updates the secret key of the tag. Then the reader sends r_2 to the tag. After the tag verifies r_2 successfully it updates its secret key. Once r_2 is tampered, the tag cannot update its secret key. But the verifier has updated the secret key of the tag and it does not reserve the old secret key of the tag. So the secret key of the tag stored in the verifier is different from the one stored in the tag. Desynchronization attack occurs. Therefore the protocol cannot resist DoS attack.

Tsai et al. [22] discussed grouping-proof protocols and ownership transfer protocols, respectively. They found that no protocol has been proposed which can achieve both requirements. So they only proposed a novel ownership transfer protocol to ensure that ownership of the cargo is transferred to the new designated owner.

Cherneva and Trahan [23] focus on security, privacy, and efficiency. They proposed a light, improved offline protocol: parallel-dependency grouping-proof protocol. But their protocol does not update any stored secret so as to resist desynchronization attack. So the protocol cannot provide forward security.

As analyzed above, many grouping-proof protocols only involve a group of tags rather than multiple tag groups. Sometimes a tag group only contains two tags. When there only exist some untrusted readers near a verifier the grouping-proof protocol cannot be started. Many grouping-proof protocols use serial signature mode to collect a grouping-proof evidence, which remarkably reduces the efficiency of the protocols. In particular, some grouping-proof protocols cannot provide forward security and they are vulnerable to privacy leakage [24].

3. RFID System under the Grouping-Proof Mode and Its Security Model

Under the grouping-proof mode, an RFID system usually includes multiple tags. These tags are combined into several groups, as shown in Figure 1. A grouping-proof protocol is for a reader to give the evidence that multiple RFID tags exist simultaneously within its broadcast range. There are two classification methods for the grouping-proof protocols:

(1) According to the role of the verifier during the grouping-proof period, the grouping-proof protocols are classified into two different modes: online and offline. For the first mode, the verifier involves the entire grouping-proof process. In contrast, for offline mode, the verifier can only send challenges to the reader and it does not need the persistent presence during the entire grouping-proof period. The efficiency of offline mode is greater than that of online mode. Therefore, many current grouping-proof protocols use offline mode.

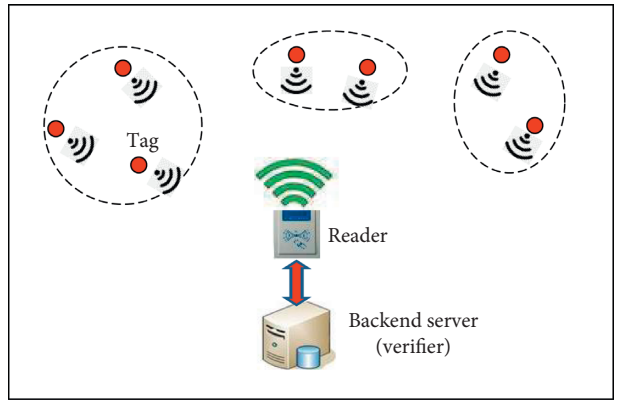

Figure 1: The components of an RFID system under the grouping-proof mode.

(2) According to the sequence for tags to complete their signature, the grouping-proof protocols are classified into two types: serial mode and parallel mode. For the first mode, after one tag finishes its signature another tag begins to sign for generating their grouping-proof evidence. For parallel mode, all tags finish their signatures almost simultaneously. So the grouping-proof protocols under parallel mode are more efficient than those under serial mode.

For an RFID system under the grouping-proof mode, some passive tags are usually used. These tags can only perform some basic cryptographic functions such as pseudorandom number generation and hash operation. We suppose that the verifier is a unique trusted entity and it shares some secrecy with tags. The reader is a potential untrusted entity and it is used to interrogate tags to generate the grouping-proof evidence. Otherwise, we also suppose that the channel between verifier and reader is secure and the channel between reader and tags is insecure. Suppose the verifier and the reader have enough computing and storing resources to complete some advanced cryptographic operations such as asymmetric encryption. For an RFID system under the grouping-proof mode, it should ensure anonymity, confidentiality, and forward security. It can effectively resist privacy leakage, eavesdropping, trace, replay, and desynchronization attack [24].

4. Grouping-Proof Protocols with Identity Authentication and Forward Security

As described above, an RFID system under the grouping-proof mode includes three kinds of entities: verifier, reader, and tag. Generally, we suppose that there are a verifier, a reader, and many tags. These tags are divided into several different groups. Each tag group is only identified by its group identifier. Each tag could be represented by $\{t_{mn}/m \in \{1, 2, \ldots, p\}, n \in \{1, 2, \ldots, q\}\}$, where t_{mn} represents that the tag is the n^{th} tag of the m^{th} group. When we analyze the security of the protocol an adversary must be introduced. It is usually assumed that an adversary is a probabilistic polynomial time algorithm. An adversary can

control each communication channel between reader and tags. He can eavesdrop, intercept, tamper, counterfeit, and replay each session between reader and tags. His main attack goal is to counterfeit a grouping-proof evidence that is verified to be valid by the verifier or to gain the secrecy of the RFID system, such as the secret key and identifier of the tag.

The reader is a potential untrusted entity. It is trusted or untrusted. Now two protocols are proposed for the reader with different security level. They utilize parallel signature mode and they are independent of the sequence accessing to tags. So they are very efficient. For the first protocol, we assume that the reader is untrusted. The reader does not know any secret about tags. So the reader cannot authenticate tags. It only collects the grouping-proof evidence and sends the evidence to the verifier. For the second protocol, the reader is assumed to be trusted and it shares some secrets with the verifier and tags. After a reader is authenticated and authorized by the verifier it can begin to collect the grouping-proof evidence. Then it sends the evidence to the verifier.

For our proposed protocols, each tag stores its current secret key tk_i^{new}, its current identifier tid_i^{new}, and its group identifier gid. A trusted reader stores its identifier rid and its secret key rk. $\{rid, rk\}$ and $\{tk_i^{\text{new}}, tid_i^{\text{new}}, tk_i^{\text{old}}, tid_i^{\text{old}}, gid\}$ are stored in the verifier. tk_i^{old} and tid_i^{old} are the last round secret key and identifier of the i^{th} tag. Let $hash(): \{0,1\}^d \longrightarrow \{0,1\}^d$ be a hash function. Let $prng(): \{0,1\}^d \longrightarrow \{0,1\}^d$ be a pseudorandom number generator. $d \geq 32$ and it is the bit number of the secret key and the identifier. The verified process is started by the reader. The symbols used in our protocols are shown as Table 1.

Table 1: The symbols used in our protocols.

Symbols	Description
gid	The identifier of a tag group
tk_i^{new}, tk_i^{old}	The current and last round secret key of the tag
tid_i^{new}, tid_i^{old}	The current and last round identifier of the tag
rk, rid	The secret key and identifier of the trusted reader
$hash()$	A secure hash function
$prng()$	A pseudorandom number generator
rv, rr, rt_i	Some pseudorandom numbers generated by the different entities
t	The timestamp of the verifier
d	The bit number of the secret key and the identifier
$\oplus$	Bitwise XOR operation

4.1. Grouping-Proof Protocol with the Untrusted Reader. For this protocol, an untrusted reader is used to collect a grouping-proof evidence. When the protocol starts a grouping-proof process, the reader first sends "hello" to the verifier. The verifier sends a message to the reader and the message includes the blinded identifier of the verified tag group. Then the reader collects the coexistence evidence of the tag group and sends the evidence to the verifier. At last, the verifier verifies the validness of the evidence. Because all messages that the reader receives are blinded or encrypted, the reader does not know any secret about the tags and the tag group during the entire grouping-proof period.

The protocol includes four steps as follows:

(1) A reader notifies the verifier that it will start a grouping-proof process.

(2) The verifier starts a timestamp and sends the blinded identifier of the verified tag group to the reader.

(3) The reader collects a grouping-proof evidence and sends the evidence to the verifier.

(4) If it is not timeout the verifier completes the authentication to the tags and verifies the grouping-proof evidence.

The protocol is shown in Figure 2 and is described as follows:

(1) The reader sends "hello" to the verifier.

(2) The verifier stores its current clock to t and starts a timestamp. It generates a pseudorandom number $rv = prng(t)$. Then it uses the verified group's identifier gid to generate the message $m1 = hash(gid \oplus rv)$ and sends $m1 \| rv$ to the reader.

(3) After the reader receives $m1 \| rv$, it broadcasts $m1 \| rv$ to all tags near it.

(4) After each tag receives $m1 \| rv$, it uses its gid to compute $tm1 = hash(gid \oplus rv)$. If $tm1 = m1$ holds, it becomes active. Otherwise, it becomes sleep. The process described above is called the activate-sleep mechanism. Later, only the active tags respond to the reader.

(5) For the i^{th} active tag, it firstly generates a pseudorandom number $rt_i = prng(rv \oplus tk_i)$. Then it uses its tk_i and tid_i to generate $m2_i = hash(tk_i \oplus rt_i)$ and $m3_i = hash(tid_i \oplus rt_i)$, and it sends $m2_i \| m3_i \| rt_i$ to the reader.

(6) After the reader receives $m2_i \| m3_i \| rt_i$ $(i \in \{1, 2, \ldots, k\})$ from each active tag, it calculates $mp = hash(m2_1 \oplus m2_2 \oplus \ldots \oplus m2_k)$ and broadcasts mp to each active tag. k is the total number of the active tags.

(7) After each active tag receives mp, it signs mp with its secret key tk_i and generates $tp_i = hash(tk_i \oplus mp)$. Then it sends tp_i to the reader.

(8) After the reader receives $tp_i (i \in \{1, 2, \ldots, k\})$ from each active tag, it calculates $p = hash(tp_1 \oplus tp_2 \oplus \ldots \oplus tp_k)$. Then it generates the grouping-proof evidence $gp = (rt_1, m2_1, m3_1, rt_2, m2_2, m3_2, \ldots, rt_k, m2_k, m3_k, mp, p)$ and sends gp to the verifier.

(9) After the verifier receives gp, it firstly judges whether it is timeout. If it is timeout, the protocol exits. Otherwise, the protocol goes to the next step.

(10) The verifier calculates $tm2_i = hash(tk_i^x \oplus rt_i)$ and $tm3_i = hash(tid_i^x \oplus rt_i)$ for $\forall x \in \{\text{new}, \text{old}\}$ and $i \in \{1, 2, \ldots, k\}$. If $tm2_i = m2_i$ and $tm3_i = m3_i$ hold for $\forall i \in \{1, 2, \ldots, k\}$, the verifier completes the

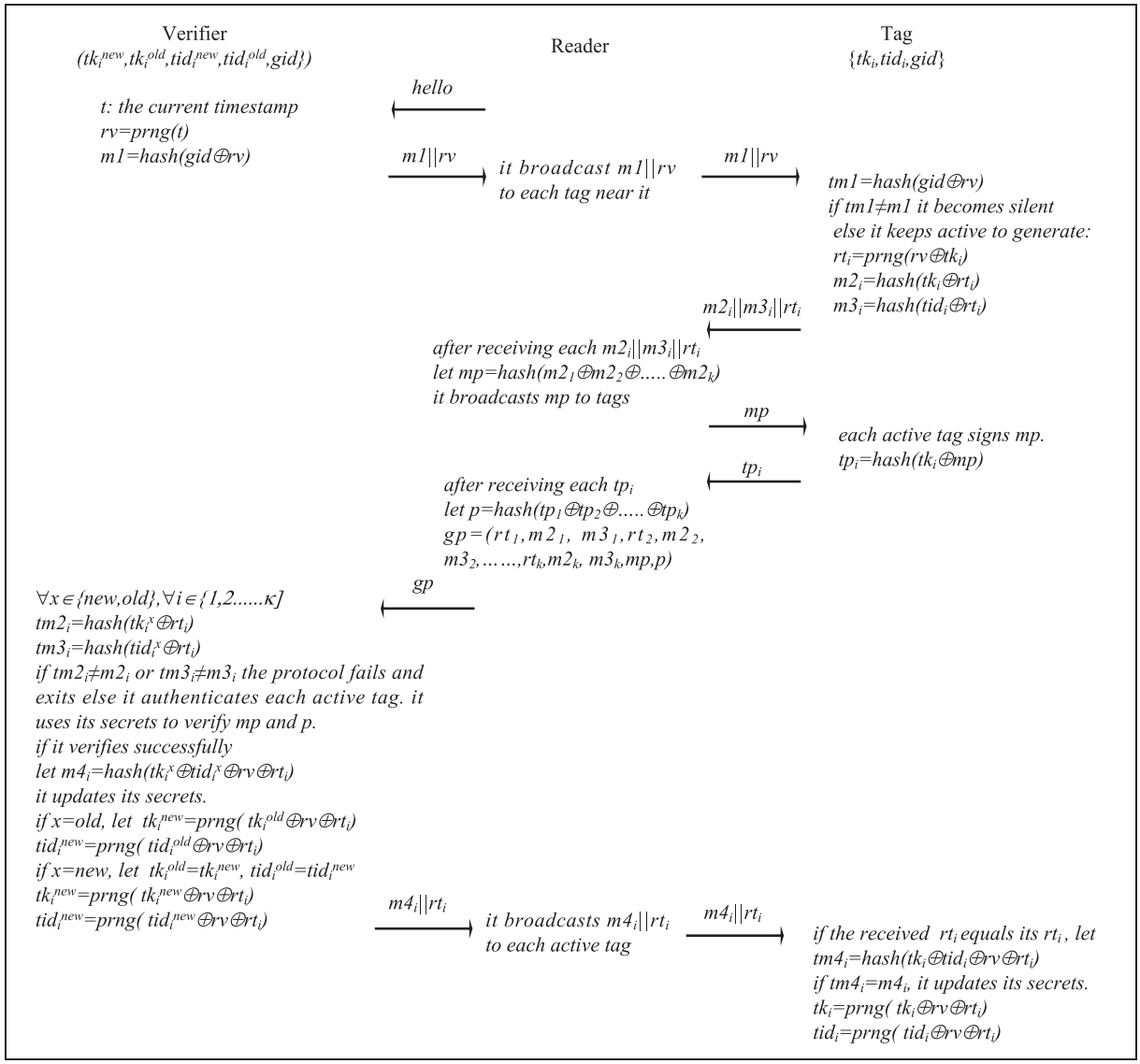

Figure 2: The grouping-proof protocol with the untrusted reader.

authentication to each active tag and begins to verify gp. Otherwise, the protocol fails and exits.

(11) The verifier uses its stored secret information about each active tag and the received rt_i to calculate mp and p. If they equal the received values, the verifier verifies gp successfully and the protocol goes to the next step. Otherwise, the protocol fails and exits.

(12) The verifier generates the message $m4_i = hash(tk_i^x \oplus tid_i^x \oplus rv \oplus rt_i)$. It begins to update its secrets. If x = old holds, let $tk_i^{\text{new}} = prng(tk_i^{\text{old}} \oplus rv \oplus rt_i)$ and $tid_i^{\text{new}} = prng(tid_i^{\text{old}} \oplus rv \oplus rt_i)$. If x = new holds, let $tk_i^{\text{old}} = tk_i^{\text{new}}$ and $tid_i^{\text{old}} = tid_i^{\text{new}}$, $tk_i^{\text{new}} = prng(tk_i^{\text{new}} \oplus rv \oplus rt_i)$, and $tid_i^{\text{new}} = prng(tid_i^{\text{new}} \oplus rv \oplus rt_i)$. Then the verifier broadcasts $m4_i \| rt_i$ to each active tag by the reader. rt_i is used to state that $m4_i \| rt_i$ is to send the i^{th} tag and other tags do not respond to the message, although they receive the message, which is called the filtering operation.

(13) After each active tag receives $m4_i \| rt_i$, it compares its rt_i with the received rt_i. If they are not equal, the tag discards the message. Or the tag calculates $tm4 = hash(tk_i \oplus tid_i \oplus rv \oplus rt_i)$. Then it compares $tm4$ with $m4_i$. If they are equal, it updates its secrets: $tk_i = prng(tk_i \oplus rv \oplus rt_i)$ and $tid_i = prng(tid_i \oplus rv \oplus rt_i)$.

4.2. Grouping-Proof Protocol with the Trusted Reader. For this protocol, a trusted reader is used to collect a grouping-proof evidence. The reader stores its identifier rid and its secret key rk, which are also stored in the verifier. When a trusted reader begins to collect a grouping-proof evidence, it first completes the mutual authentication with the verifier. If the authentication succeeds, the verifier sends the related information of the verified group to the reader and the

information includes the secret key and identifier of the verified tags. Then the reader begins to collect the coexistence evidence of the tags and sends the evidence to the verifier.

The protocol includes the following steps:

(1) The reader notifies the verifier and it will start a grouping-proof.

(2) The verifier completes the mutual authentication with the reader and authorizes it.

(3) The verifier starts a timestamp and sends the related information of the verified tags to the reader.

(4) The reader completes the first authentication to each verified tag, collects a grouping-proof evidence, and sends the evidence to the verifier.

(5) If it is not timeout, the verifier completes the second authentication to each verified tag. Then it begins to verify the grouping-proof evidence and updates its secrets.

(6) The verifier notifies the related tags to update their secrets.

The protocol includes three phases. The first phase completes the authentication and authorization of the verifier to the reader. It is shown in Figure 3 and is described as follows:

(1) The reader sends "hello" to the verifier.

(2) The verifier stores its current clock to t, starts a timestamp, and generates a pseudorandom number $rv = prng(t)$. It sends rv to the reader.

(3) The reader generates a pseudorandom number $rr = prng(rv \oplus rk)$ and a message $m1 = hash(rid \oplus rr)$. It sends $rr \| m1$ to the verifier.

(4) The verifier uses rid, which is stored in its database, to generate $tm1 = hash(rid \oplus rr)$. If $m1 = tm1$ holds, it completes the authentication to the reader. Then it generates $m2 = hash(rk \oplus rv \oplus rr)$ and sends $m2$ to the reader. Otherwise, the protocol fails and exits.

(5) The reader uses its rk to generate $tm2 = hash(rk \oplus rv \oplus rr)$ and compares $tm2$ with $m2$. If they are equal, the reader completes the authentication to the verifier. Then it generates $m3 = hash(rk \oplus rr)$ and sends $m3$ to the verifier. If they are unequal, the protocol fails and exits.

(6) The verifier uses its rk, which is stored in its database, to compute $tm3 = hash(rk \oplus rr)$ and compares $tm3$ with $m3$. If they are equal, the verifier completes the mutual authentication with the reader. If they are unequal, the protocol fails and exits.

(7) After the verifier completes the mutual authentication with the reader, it transfers (tk_i^x, tid_i^x) of each verified tag and the verified group identifier gid to the reader by a secure channel or a secure cryptographic primitive, where $x \in \{\text{new}, \text{old}\}$; $i \in \{1, 2, \ldots, k\}$, k is the total number of the verified tags. The reader is authorized to collect a grouping-proof evidence.

In the second phase, the reader wakes up the related tags and completes the first authentication to each verified tag. It is shown in Figure 4 and is described as follows:

(1) The reader generates the message $m4 = hash(gid \oplus rr)$ and broadcasts $m4 \| rr \| rv$ to each tag near it.

(2) After a tag receives $m4 \| rr \| rv$, it uses its gid to generate $tm4 = hash(gid \oplus rr)$. If $m4 = tm4$ holds, it remains active. Otherwise, it becomes sleep.

(3) For the i^{th} active tag, it generates a pseudorandom number $rt_i = prng(tk_i \oplus rr)$ and two messages $mk5_i = hash(tk_i \oplus rt_i)$ and $mid5_i = hash(tid_i \oplus rt_i)$ and sends $mk5_i \| mid5_i \| rt_i$ to the reader.

(4) The reader uses the secret information from the verifier and calculates $tmk5_i = hash(tk_i^x \oplus rt_i)$ and $tmid5_i = hash(tid_i^x \oplus rt_i)$, where $x \in \{\text{new}, \text{old}\}$ and $i \in \{1, 2, \ldots, k\}$. If $tmk5_i = mk5_i$ and $tmid5_i = mid5_i$ hold for $\forall i \in \{1, 2, \ldots, k\}$, the reader completes the first authentication to each active tag. Otherwise, the protocol fails and exits.

(5) Once the reader completes the authentication to each active tag, it begins to collect the grouping-proof evidence and enter the verification period.

The third phase completes the collection and verification of the grouping-proof evidence. It is shown in Figure 5 and is described as follows:

(1) The reader calculates $mp = hash(mk5_1 \oplus mk5_2 \oplus \ldots \oplus mk5_k)$ and broadcasts mp to each active tag.

(2) After each active tag receives mp, it signs mp with its secret key tk_i and generates $tp_i = hash(tk_i \oplus mp)$. Then it sends tp_i to the reader.

(3) After the reader receives each tp_i it calculates $p = hash(tp_1 \oplus tp_2 \oplus \ldots \oplus tp_k)$. Then it generates the grouping-proof evidence $gp = (rt_1, mk5_1, mid5_1, \ldots, rt_k, mk5_k, mid5_k, mp, p)$ and sends gp to the verifier.

(4) After the verifier receives gp, it firstly judges whether it is timeout. If it is timeout, the protocol exits. Otherwise, the protocol goes to the next step.

(5) The verifier utilizes its stored secret information about tags and the received rt_i to calculate $vmk_i = hash(tk_i^x \oplus rt_i)$ and $vmid_i = hash(tid_i^x \oplus rt_i)$ for $\forall i \in \{1, 2, \ldots, k\}$. If $vmk_i = mk5_i$ and $vmid_i = mid5_i$ hold for $\forall i \in \{1, 2, \ldots, k\}$, the verifier completes the second authentication to the verified tags and it begins to verify gp. Otherwise, the protocol fails and exits.

(6) The verifier generates $vmp = hash(vmk5_1 \oplus vmk5_2 \oplus \ldots \oplus vmk5_k)$ and calculates $vtp_i = hash(tk_i \oplus vmp)$ for $\forall i \in \{1, 2, \ldots, k\}$. Finally it generates $vp = hash(vtp_1 \oplus vtp_2 \oplus \ldots \oplus vtp_k)$. If $vmp = mp$

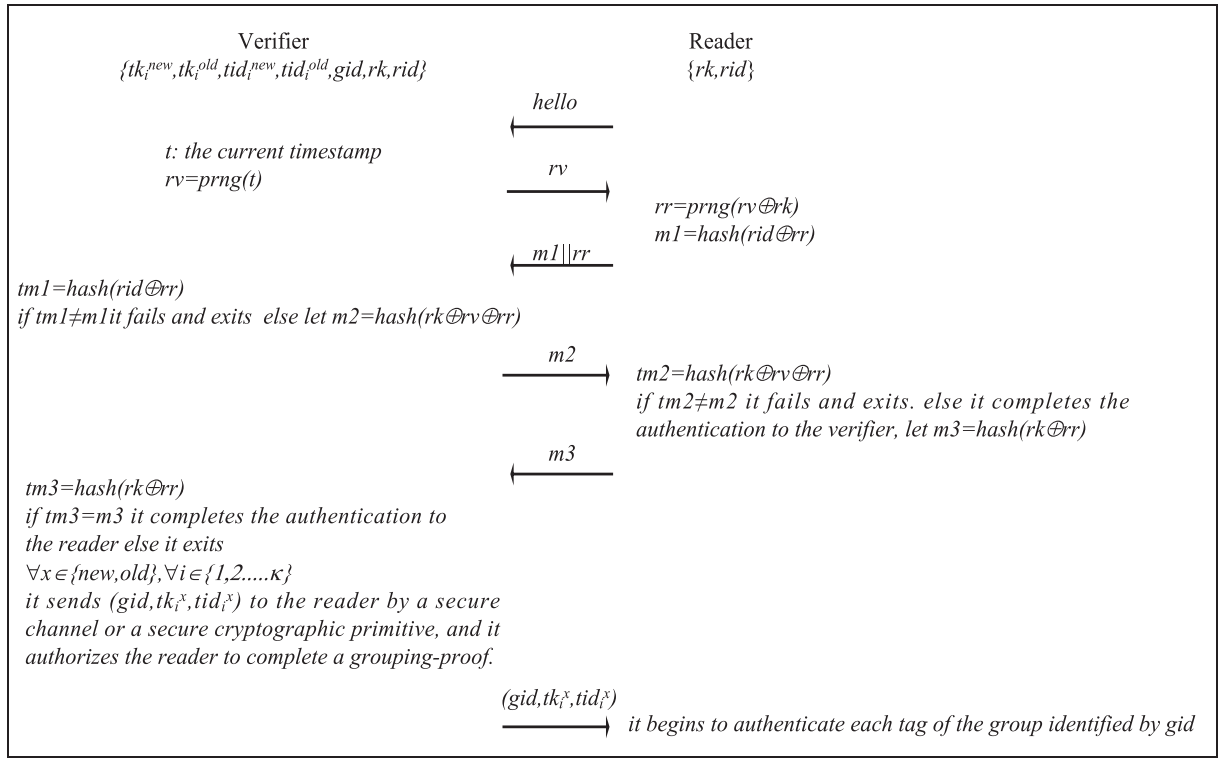

Figure 3: The authentication and authorization of the verifier to the reader.

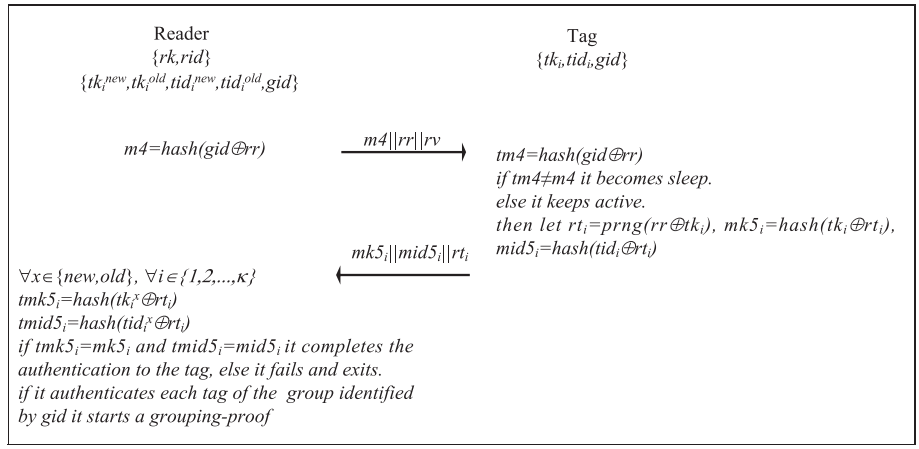

Figure 4: The reader authenticates each verified tag.

and $vp = p$ hold, the verifier gets a valid grouping-proof evidence.

(7) After the verifier verifies gp successfully, it generates the message, $m6_i = hash(tk_i^x \oplus tid_i^x \oplus rt_i)$ for $\forall i \in \{1, 2, \ldots, k\}$, and updates its secrets. If x = old holds, let $tk_i^{\text{new}} = prng(tk_i^{\text{old}} \oplus rv \oplus rt_i)$ and $tid_i^{\text{new}} = prng(tid_i^{\text{old}} \oplus rv \oplus rt_i)$. If x = new holds, let $tk_i^{\text{old}} = tk_i^{\text{new}}$, $tid_i^{\text{old}} = tid_i^{\text{new}}$, $tk_i^{\text{new}} = prng(tk_i^{\text{new}} \oplus rv \oplus rt_i)$, and $tid_i^{\text{new}} = prng(tid_i^{\text{new}} \oplus rv \oplus rt_i)$. Then the verifier broadcasts $m6_i \| rt_i$ to each active tag through the reader.

(8) After an active tag receives $m6_i \| rt_i$, it compares its rt_i with the received rt_i. If they are equal, the tag calculates $tm6 = hash(tk_i \oplus tid_i \oplus rt_i)$. If $m6_i = tm6$ holds, it updates its secrets: $tk_i = prng(tk_i \oplus rv \oplus rt_i)$ and $tid_i = prng(tid_i \oplus rv \oplus rt_i)$.

5. Security and Efficiency Analysis of Our Proposed Protocols

For an RFID system under the grouping-proof mode, A is assumed to be a probabilistic polynomial time adversary. He

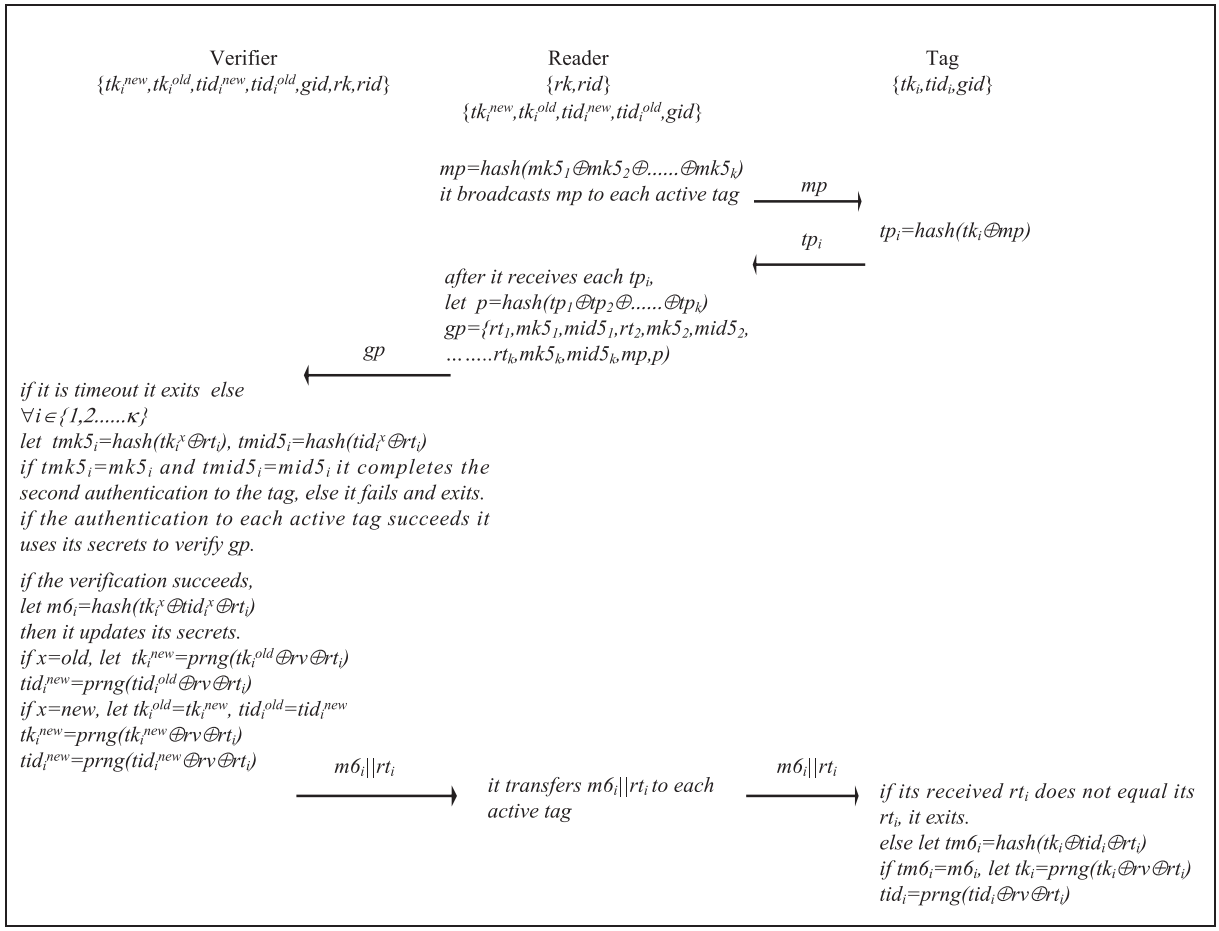

Figure 5: The generation and verification of the grouping-proof evidence.

can eavesdrop, intercept, tamper, counterfeit, and replay each session between reader and tags. He can counterfeit a grouping-proof evidence and transfer it to the verifier in limited time. If the evidence is successfully verified by the verifier, adversary A is considered to win.

Definition 1. Adversary A can continuously issue the oracle queries to $prng()$ and $hash()$. The output of $prng()$ and $hash()$ is d bits. Let σ denote the probability that the adversary guesses successfully the output of the functions. Then we have $\sigma \leq 2^{-d}$.

Definition 2. For a probabilistic polynomial time adversary A, let σ be the probability that he reveals the secret information of an RFID system. If σ is negligible the grouping-proof protocol is considered to be privacy-secure.

Definition 3. For a probabilistic polynomial time adversary A, let σ denote the probability that he distinguishes two different tags. σ is defined as follows:

$$\sigma = 2\Pr\left[tid_m = tid_n\right] - 1 \quad (1)$$

where $m \neq n$. If σ is negligible, the grouping-proof protocol is considered to be indistinguishable-secure.

Definition 4. For a probabilistic polynomial time adversary A, a grouping-proof protocol is defined to be forward-secure if and only if he cannot decrypt any previous session, although he has acquired the current secret key of the RFID system. Let σ denote the probability that he could derive the previous secret key from the current secret key of the protocol. If σ is negligible and the adversary cannot decrypt the previous sessions, the grouping-proof protocol is considered to be forward-secure.

5.1. Security Analysis to the Grouping-Proof Protocol with the Untrusted Reader. For the first grouping-proof protocol proposed by us, an untrusted reader is involved. We assume that an adversary A easily disguises a legal reader to communicate with the verifier or the tags. He can intercept each session from the RFID system, such as $m2_i \| m3_i \| rt_i$, tp_i and $m4_i \| rt_i$. On the one hand, $m2_i$, rt_i, tp_i, and $m4_i$ are four messages that include the secret key tk_i of the i^{th} tag. Suppose adversary A intercepts these messages. Let $\varepsilon 1$ denote the probability that adversary A guesses tk_i from the messages. We have $\varepsilon 1 \leq 2^{-32} \times 4 = 2^{-30}$. On the other hand, $m3_i$ and $m4_i$ are two messages that include the identifier tid_i of the i^{th} tag. Let $\varepsilon 2$ denote the probability that adversary A guesses

tid_i from the messages. We have $\varepsilon 2 \le 2^{-32} \times 2 = 2^{-31}$. It is obvious that $\varepsilon 1$ and $\varepsilon 2$ are negligible. It means that it is very difficult for the adversary to guess any secret information from the intercepted sessions. Therefore the protocol is privacy-secure.

For a probabilistic polynomial time adversary A, we assume that he can intercept each session from tags. Suppose the adversary intercepts $m3_m$ and $m3_n$ from the m^{th} and n^{th} tag, where $m, n \in \{1, 2, \ldots, k\}$ and $m \ne n$. If the adversary can distinguish these two tags, his successful probability can be defined as follows [25]:

$$\Pr[tid_m = tid_n] = 2^{-1} + \varepsilon \tag{2}$$

where ε is the probability that the adversary can guess tid_m and tid_n simultaneously. By Definition 1, we have $\varepsilon \le 2^{-d} \times 2^{-d}$. When $d \ge 32$, we have $\varepsilon \le 2^{-64}$. By Definition 3, we have $\sigma = 2\Pr[tid_m = tid_n] - 1 = 2\varepsilon \le 2^{-63}$. Therefore σ is negligible. The grouping-proof protocol is indistinguishable-secure.

For our proposed grouping-proof protocol with an untrusted reader, $m2_i$, rt_i, tp_i, and $m4_i$ are four messages that include the secret key tk_i of the i^{th} tag. Suppose adversary A intercepts these messages. Let $\varepsilon 1$ denote the probability that adversary A guesses tk_i successfully from the messages. We have $\varepsilon 1 \le 2^{-32} \times 4 = 2^{-30}$. After each successful grouping-proof, the secret key of each tag is updated by $tk_i^{\text{new}} = prng(tk_i^{\text{new}} \oplus rv \oplus rt_i)$. If the adversary wants to get the last round secret key it has to issue the oracle query to $prng()$. Suppose the adversary can deduce the last round secret key from the current secret key by querying $prng()$. His successful probability is $\varepsilon 2$. Then we have $\varepsilon 2 \le 2^{-32}$. There are two cases:

(1) The adversary does not corrupt the i^{th} tag and it does not know the current secret key tk_i. Firstly, the adversary has to guess the current secret key from the intercepted sessions. Then he guesses the previous secret key from the guessed current secret key by issuing the random queries to $prng()$. Let $\sigma 1$ be the probability that the adversary guesses the last round secret key. We have $\sigma 1 = \varepsilon 1 \times \varepsilon 2 \le 2^{-30} \times 2^{-32} = 2^{-62}$.

(2) The adversary corrupts the i^{th} tag and it gets the current secret key tk_i; he can guess the last round secret key only by issuing the oracle queries to $prng()$. Let $\sigma 2$ be the probability that the adversary wins. Then we have $\sigma 2 = \varepsilon 2 \le 2^{-32}$.

It is obvious that $\sigma 1$ and $\sigma 2$ are negligible. The adversary cannot guess the last round secret key from the current secret key. So he cannot reveal the previous sessions and the grouping-proof protocol is forward-secure.

5.2. Security Analysis to the Grouping-Proof Protocol with the Trusted Reader. The second grouping-proof protocol proposed by us involves a trusted reader. Under this circumstance, the verifier and the reader can use some complicated cryptographic primitives to ensure the confidential communication between them. So we assume that the communication between verifier and reader is secure. An adversary can only intercept sessions between reader and tags. If adversary A wants to guess the secret key and identifier of tags from the intercepted sessions it has to issue the oracle queries to $hash()$ and $prng()$. On the one hand, rt_i, $mk5_i$, tp_i, and $m6_i$ include the secret key tk_i of the i^{th} tag. Let σ denote the probability that an adversary successfully guesses the secret key of the tag from these sessions and we have $\sigma \le 4 \times 2^{-32} = 2^{-30}$. It is obvious that σ is negligible. On the other hand, only $mid5_i$ and $m6_i$ include the identifier tid_i of the i^{th} tag. Suppose an adversary can guess the identifier by issuing the oracle queries to $hash()$. Let σ be the probability that he wins by querying $mid5_i$ and $m6_i$. Then we have $\sigma \le 2 \times 2^{-32} = 2^{-31}$. It is obvious that σ is also negligible. So our proposed protocol is privacy-secure.

Adversary A can distinguish two different tags by intercepting some sessions that include the identifier of these tags. We assume that adversary A intercepts $mid5_m$ and $mid5_n$ from the m^{th} and n^{th} tag, where $m, n \in \{1, 2, \ldots, k\}$ and $m \ne n$. If A can distinguish these two tags, his successful probability is defined by equation (2). As discussed in the last subsection, we have $\varepsilon \le 2^{-d} \times 2^{-d}$. When $d \ge 32$, we have $\varepsilon \le 2^{-64}$. By Definition 3, we have $\sigma = 2\Pr[tid_m = tid_n] - 1 = 2\varepsilon \le 2^{-63}$. So σ is negligible and our grouping-proof protocol is indistinguishable-secure.

Now we discuss the forward security of the protocol. For our proposed grouping-proof protocol with the trusted reader, rt_i, $mk5_i$, tp_i, and $m6_i$ are some sessions that include the secret key tk_i to the i^{th} tag. Suppose adversary A can intercept these sessions. He can issue any oracle query to $hash()$ and $prng()$. $\varepsilon 1$ is the probability that he can guess tk_i from the messages described above. We have $\varepsilon 1 \le 4 \times 2^{-32} = 2^{-30}$. After each successful grouping-proof, the secret key to each tag is updated by $tk_i^{\text{new}} = prng(tk_i^{\text{new}} \oplus rv \oplus rt_i)$. If the adversary wants to get the last round secret key he has to issue the oracle query to $prng$. Let $\varepsilon 2$ denote the probability that the adversary guesses the last round secret key from the current secret key by issuing the random queries to $prng()$. We have $\varepsilon 2 \le 2^{-32}$. There exist two cases as described in the last subsection. The probability that the adversary gains the last round secret key from the current secret key is negligible. The adversary cannot guess the previous secret key from the current secret key. So he cannot decrypt the previous sessions and the grouping-proof protocol is forward-secure.

5.3. Resistance to Other Attacks. In addition to resisting the attacks described above, our proposed grouping-proof protocols can also resist eavesdropping attack, replay attack, and desynchronized attack.

(i) Eavesdropping: during the grouping-proof period, all session messages, which include the secret information of the RFID system, are generated by $hash$ or randomized by $prng$. An adversary can intercept each session from the protocol. But he cannot reveal any secret information about the tag and the tag group from the intercepted sessions. Eavesdropping to the communication channels is invalid.

Table 2: The comparison of some typical grouping-proof protocols with our protocols.

Protocols	Privacy	Anonymity	Trace attack	Replay attack	Forward security	DoS attack	The number of tag groups
Ref. [2]	×	×	×	×	×	—	One (only two tags)
Ref. [15]	✓	✓	✓	✓	✓	×	One
Ref. [16]	×	×	×	✓	✓	×	Multiple
Ref. [18]	✓	✓	✓	✓	×	—	One (only two tags)
Ref. [19]	✓	✓	×	✓	×	—	One (only two tags)
Ours	✓	✓	✓	✓	✓	✓	Multiple

(ii) Interleaving and replay attack: this type of attack means that an adversary replays the grouping-proof evidence that he intercepted and the replayed evidence can be successfully verified by the verifier. The intercepted evidence may be from the same or different grouping-proof process. In order to prevent interleaving and replay attack, the clock of the verifier is utilized as timestamp and seed to generate some pseudorandom numbers. These pseudorandom numbers are different for different grouping-proof processes and they are utilized to randomize the sessions between reader and tags. On the one hand, the sessions from the same grouping-proof process can be replayed later. But they are timeout and they cannot be verified successfully. So our protocols can resist replay attack. On the other hand, the sessions from the different grouping-proof processes include the different timestamps. So they cannot be combined to construct any valid grouping-proof evidence. Hence our protocols can resist interleaving attack.

(iii) Desynchronization: in order to resist desynchronization attack, our protocols reserve the last round secrecy and the current secrecy in the verifier when the secrecy of the RFID system is updated. An adversary can tamper or block $m6_i||rt_i$ so that the tag cannot update its current secrecy. But the verifier reserves the last round secrecy and it can use this secrecy to communicate with tags. So our protocols can complete the grouping-proof regardless of whether the tag updates its current secrecy. The protocol can avoid desynchronization attack.

5.4. Analysis to the Efficiency of Our Proposed Protocols. In order to avoid the collision between tags and reduce the computing load of the RFID system, the novel activate-sleep mechanism and the special filtering operation are proposed for our grouping-proof protocols.

(i) The activate-sleep mechanism: for our protocols, maybe there exist many tag groups. Each tag group is only identified by its group identifier *gid*. Before our protocols begin to authenticate tags and generate the grouping-proof evidence, the reader sends the message $m1$ or $m4$ to each tag group so that the tags with other group identifiers become sleep. During the later period of the protocol, only the tags with the group identifier *gid* can communicate with the reader. When there exist many tag groups, the collision probability between tags is reduced remarkably. Otherwise, the reader only receives the messages from the objective group and other tag groups do not send any message to it. Its processing load is reduced efficiently.

(ii) The filtering operation: the computing ability of tags is very limited. So it is necessary to reduce the computing load of tags. For our grouping-proof protocols, the reader uses the broadcast channel to communicate with tags. But sometimes the reader sends a message only to one tag (e.g., $m6_i\|rt_i$ in Figure 5). In order to complete the peer-to-peer communication through the RFID broadcast channel, the theorem of the data link layer of Ethernet is utilized. rt_i is defined as MAC address of the i^{th} tag. The message that is only sent to the i^{th} tag is attached with rt_i. After a tag receives the messages, it first recognizes whether the received rt_i equals its stored rt_i. After the tag is sure that the received message is sent to it, it calls $hash()$ to calculate $m6_i$. Therefore the computing load of the tag is reduced remarkably.

The comparison of our proposed protocols with some typical grouping-proof protocols is shown in Table 2.

6. Conclusions

For some RFID applications, multiple tags are often combined together to identify a group of different objects or different parts of an object. Therefore, it is necessary to acquire the coexistence evidence of a group of tags. As an important component of an RFID system, the tags usually are some passive ones and they only have some very limited computing and memory resources. It is difficult for these tags to complete some advanced cryptographic operations. Therefore, we only use some lightweight functions and bitwise operation to propose two grouping-proof protocols. These protocols involve multiple tag groups. They efficiently use the activate-sleep mechanism and the filtering operation to reduce the collision between tags and the computing load of the RFID system. They only utilize a hash function and a pseudorandom number generator to encrypt all sessions transferred between reader and tags. This ensures the confidentiality and privacy of the RFID system. Meanwhile, our protocols use pseudorandom numbers to randomize each session of the protocols so as to resist trace attack and replay attack. After each grouping-proof, the secrecy of tags is updated and the last round secrecy of tags is preserved. Therefore, our proposed protocols provide forward security

and resist desynchronization attack. Otherwise, our protocol can complete a grouping-proof regardless of whether the reader is untrusted or trusted.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61802252 and 61701296) and the Opening Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security (AGK2019004).

References

[1] J. Kang, "Lightweight mutual authentication RFID protocol for secure multi-tag simultaneous authentication in ubiquitous environments," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4529–4542, 2019.

[2] A. Juels, "Yoking-proofs for RFID tags," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pp. 138–143, Orlando, FL, USA, March 2004.

[3] M. Burmester, B. de Medeiros, and R. Motta, "Provably secure grouping-proofs for RFID tags," Smart Card Research and Advanced Applications, In Proceedings of the 8th IFIP WG 8.8/11.2 international conference on Smart Card Research and Advanced Applications-CARDIS'08, Springer, London, UK, 2008, pp. 176–190.

[4] P. Peris-Lopez, A. Orfila, J. C. Hernandez-Castro, and J. C. A. van der Lubbe, "Flaws on RFID grouping-proofs. Guidelines for future sound protocols," *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 833–845, 2011.

[5] J. Satio and K. Sakurai, "Grouping-proof for RFID tags," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pp. 621–624, Taipei, Taiwan, March 2005.

[6] D. Sun and Z. Zhu, "Improved RFID yoking proof protocol," *Computer Engineering and Design*, vol. 38, no. 8, pp. 2076–2080, 2017, in Chinese.

[7] X. Leng, Y. Lien, K. Mayes, K. Markantonakis, and J.-H. Chiu, "Select-response grouping-proof for RFID tags," *Asian Conference on Intelligent Information and Database Systems,* pp. 73–77, IEEE Computer Society, Dong Hoi City, Vietnam, April 2009.

[8] H.-H. Huang and C.-Y. Ku, "A RFID grouping proof protocol for medication safety of inpatient," *Journal of Medical Systems*, vol. 33, no. 6, pp. 467–474, 2009.

[9] H.-Y. Chien, C.-C. Yang, T.-C. Wu, and C.-F. Lee, "Two RFID-based solutions to enhance inpatient medication safety," *Journal of Medical Systems*, vol. 35, no. 3, pp. 369–375, 2011.

[10] P. Peris-Lopez, A. Orfila, A. Mitrokotsa, and J. C. A. van der Lubbe, "A comprehensive RFID solution to enhance inpatient medication safety," *International Journal of Medical Informatics*, vol. 80, no. 1, pp. 13–24, 2011.

[11] Y.-C. Yen, N.-W. Lo, and T.-C. Wu, "Two RFID-based solutions for secure inpatient medication administration," *Journal of Medical Systems*, vol. 36, no. 5, pp. 2769–2778, 2012.

[12] 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, no. 7, pp. 1321–1330, 2013.

[13] J. Shen, H. Tan, Y. Wang, S. Ji, and J. Wang, "An enhanced grouping proof for multiple RFID readers and tag groups," *International Journal of Control and Automation*, vol. 7, no. 12, pp. 239–246, 2014.

[14] D. Moriyama, "Provably secure two-round RFID grouping-proof protocols," in *Proceedings of the 2014 IEEE RFID Technology and Applications (RFID-TA) Conference*, pp. 272–276, Tampere, Finland, September 2014.

[15] 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, no. 6, pp. 961–975, 2014.

[16] P. Huang and H. Mu, "A high-security RFID grouping proof protocol," *International Journal of Security and Its Applications*, vol. 9, no. 1, pp. 35–44, 2015.

[17] J. Shen, H. Tan, Y. Ren, L. Qi, and D. Wang, "A practical FRID grouping authentication protocol in multiple-tag arrangement with adequate security assurance," in *Proceedings of the 2016 18th International Conference on Advanced Communication Technology*, pp. 693–699, Pyeongchang Kwangwoon Do, South Korea, January 2016.

[18] K. Hong-yan, "Analysis and improvement of ECC-based grouping-proof protocol for RFID," *International Journal of Control and Automation*, vol. 9, no. 7, pp. 343–352, 2016.

[19] L. Batina, Y. K. Lee, S. Seys, D. Singelée, and I. Verbauwhede, "Privacy-preserving ECC-based grouping proofs for RFID," *Lecture Notes in Computer Science*, vol. 6531, pp. 159–165, 2011.

[20] D. Sun and Y. Mu, "Security of grouping-proof authentication protocol for distributed RFID systems," *IEEE Wireless Communications Letters*, vol. 7, no. 2, pp. 254–257, 2018.

[21] 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, no. 1, pp. 133–143, 2018.

[22] K.-Y. Tsai, M. Yang, J. Luo, and W.-T. Liew, "Novel designated ownership transfer with grouping proof," *Applied Sciences*, vol. 9, no. 4, p. 724, 2019.

[23] V. Cherneva and J. Trahan, "A secure and efficient parallel-dependency RFID grouping-proof protocol," in *Proceedings of the IEEE International Conference on RFID*, pp. 1–8, Phoenix, AZ, USA, April 2019.

[24] Z. Shi, X. Zhang, and Y. Wang, "A lightweight RFID grouping-proof protocol based on parallel mode and DHCP mechanism," *Information*, vol. 8, no. 3, p. 85, 2017.

[25] Y. Zhou and D. Feng, "Design and analysis of cryptographic protocols for RFID," *Chinese Journal of Computers*, vol. 29, no. 4, pp. 581–589, 2006, in Chinese.