# An Anti-counterfeit Complete RFID Tag Grouping Proof Generation Protocol

Gao-Feng Shen[1], Shu-Min Gu[2], and Dao-Wei Liu[3]
*(Corresponding author: Gao-Feng Shen)*

Department of Computer Science, Zhengzhou University of Light Industry[1]
Zhengzhou 450002, China
(Email: shgf_123@163.com)
Department of Basic Course, College of Information and Business, Zhongyuan University of Technology[2]
Department of Computer Science and Engineering, Guangzhou College of Technology and Business[3]

*(Received May 27, 2018; Revised and Accepted Oct. 18, 2018; First Online June 13, 2019)*

## Abstract

Most existing radio frequency identification group tags prove that the generation protocol does not meet the lightweight Gen-2 standard and there are existing security issues such as rapid brute-force cracking, proof forgery, and incomplete authentication. To tackle with these problems, an improved anti-counterfeit complete RFID tag grouping proof generation protocol was designed. The protocol adopted a one-way pseudo-random function that conformed to the low-cost Gen-2 standard as the basic encryption algorithm to implement complete tripartite authentication, label group proof generation and verification. The encryption authorization Mark identification was used by the database and the cipher-text transmission mechanism was ures to avoid multiple malicious attacks on the protocol. Finally, the feasibility of the protocol was approved by the GNY logic, and the security attack description indicated that the improved protocol met the security standards. Performance comparison analysis showed that the protocol was in line with low-cost applications.

*Keywords: Cipher-text Transmission; Mark Identity; Proof Generation; RFID; The Gen-2 Standard*

## 1 Introduction

Radio frequency identification (RFID) is a technology for sensing and recognizing a designated object through radio signals to process related data. With the continuous application and development of Internet of Things technology, RFID technology has been known and applied widely [9]. There into, the RFID tag grouping proof mechanism, that is, the proof mechanism that provides a group of RFID tags coexisting at a certain time, is getting more and more attention. This mechanism is playing a very important role in ensuring the security and integrity of the tag's entity and can be applied to a variety of scenarios. For example, in the field of medicine, it should be ensured that patients' multiple medications are delivered at exactly the same time, or a certain drug and its instructions for usage must be enclosed at the same time as it was sold. Similarly, in the field of equipment production, manufacturers need to assure the purchasing organization that all parts and components of a certain equipment are delivered at the same time and that the parts have indeed left the factory along with their safety shields [5,11,13,17]. In these scenarios, it is not adequate just to ensure the security of a single-tag entity, but to uniformly verify the entity of the multiple tags in a group. Therefore, designing a RFID tag grouping proof protocol with high security and suitability for low-cost tag applications is currently one of the hottest research topics. Reference [9] proposed the concept of conjugate proof of RFID tags for the first time, and the simultaneous existence proof showed that the system could generate two tags at the same time. Reference [9] further gave the corresponding proof for the generation protocol. With the introduction of the concept of label conjugation proof, many protocol researchers had been furtherly developed the grouping proof schemes for multiple tags within a group, in which the simultaneous existence of a group of tags were certified by the reader and group tags could be generated in a predetermined time interval.

The related grouping proof documents are as follows: Reference [1] proposed an ultra-lightweight tag grouping proof protocol based on simple bit operations. The protocol was simple and the cost was low for large-scale production applications. But it was also due to this simplicity and lightweight of the protocols, they were vulnerable to forgery and counterfeit attacks. Later, Reference [8] proposed a more secure tag grouping proof protocol based on heavyweight public key cryptography. However, when the protocol was executed, the calculation process was complex, and the tag computation cost was high, which was not suitable for practical applications of low-cost passive tags. Reference [18] proposed a tag grouping proof protocol based on general combination security [19]. This protocol did not require a trusted third party, but only needed a pseudo-random number generator which would perform bit manipulation, and to a certain extent, could reduce the system cost. However, being analyzed in [10], the protocol was vulnerable to message integrity attacks. Reference [2] proposed an improved RFID tag grouping proof protocol based on the message verification code function, but the article failed to demonstrate the theory explicitly, and its security still needs to be systematically proved. In the follow-up, researchers proposed a more secure symmetric cryptographic tag grouping proof protocol based on hash function and one-way pseudo-random function to meet the requirement of application of lightweight cost [3, 6]. Reference [6] proposed a tag grouping proof protocol based on the hash function. There was no dependency between the tags and they had high reliability. However, it was found in this study that the tag grouping before the reader authentication processes the generation was proved to be slightly duplicated and not concise; and the agreement could not achieve full authentication where there existed the threat of counterfeit attacks. Reference [3] was based on a one-way pseudo-random function and proposed a lightweight group privacy protection protocol. However, through the research of this paper, it was found that the Reference [3] showed inadequate resistance to brute force attack and proof forgery threat.

The rest of the paper was organized as follows: The second part analyzed the security vulnerabilities in references [6] and [3]. The third part proposed its own improved protocol for the security loopholes in the above references [6] and [3]. The fourth part gave the proof of the formalization of GNY logic of the improved protocol, which demonstrated feasibility and legitimacy of this protocol. The fifth part performed a security analysis of the protocol. It also compared the related documents. A conclusion was drawn that this protocol had higher security. The sixth part gave a comparison of the performance of the protocols and related literature, which showed that the agreement costed less and had high efficiency. The seventh part gave a summary and outlook.

## 2 References [6] and [3] Security Vulnerability Analysis

The security analysis of RFID tag grouping proof protocol based on hash algorithm proposed by Reference [6] was as follows:

Because Reference [6] did not have a complete safety certification process, and only had the mutual authentication process between the reader and the tag, it lacked verifier's verification process for the reader and tag. Therefore, the attacker could directly generate the random number $R_1$ and $R_2$ and even the information $M = MAC[m_1, m_2, m_3 \ldots]$ through the counterfeit reader. The computation factor $m_i$ in $M$ could be obtained by an attacker eavesdropping on $c_t$ using the random number $R_1$ generated by impersonation $m_i = MAC[c_t, R_{i1}]$. Finally, because the protocol lacked the verifier's verification process for the reader, the attacker would forge the grouping proof $P$, $P = (M, R_1, R_2, \ldots)$, which the verifier would still automatically verify. Therefore, Reference [6] presented the security vulnerability of incomplete authentication and the loophole for forgery attack.

The security analysis of RFID tag grouping proof protocol based on one-way pseudo-random function proposed by Reference [3] was as follows:

Reference [3] had the threat from brute force because the security of the Reference [3] depended entirely on the choice of encryption function, and the secret information $N_1$-$N_5$ was transmitted in plain text. In the communication data $r_i = g(PID_i \oplus N_2)$ and $r_R = g(PID_i \oplus N_3)$, $r_i$, $r_R$,$N2$-$N3$, the pseudo-random algorithm $g()$ was disclosed. Only the $PID_i$ was unknown, and the attacker could easily use illegal interception and interception to perform a brute force attack and obtain the tag identification information $PID_i$, so that the tags were maliciously tracked and the brute-force attacks succeed.

Reference [3] had proved flawed because the attacker could quickly decrypt the tag key information $S_i$ according to the public hash algorithm, communication data $m$, $m_i = h(S_i, r_i')$ and the eavesdropped plain text data $r_i'$. Meanwhile, the important component of the protocol identification $P$ was the identifier ticket. Because $S_i$ and $PID_i$ had been obtained, according to the formula $V = E_{K_i}(ticket, PID_i)$, the attacker could crack the session secret identifier ticket, and then falsify tag grouping proof $P$, $P = h(K||ticket||m_1 \oplus \ldots \oplus m_n)$. The forgery attack was thus successful.

In summary, this article aimed to improve the functions of above-mentioned references [3, 6], including the resistance against brute-force attacks, forgery attacks, the loophole of incomplete authentication, and the defect of high system cost and complexity, and to forge a more secure, fully-certified RFID tag grouping proof generation protocol that met the Gen-2 standard. In order to reduce system cost, this protocol used a pseudo-random function and did not use the hash function because the implementation of the Gen-2 standard in the Global Product Electronic Code Center had become the design standard for the RFID tag industry. The Gen-2 standard stipulated that only 2500-5000 circuits in the tag could be used for calculation. The commonly used hash function (MD5 requires 15,000-20,000 circuits) was not suitable for the Gen-2 standard [12, 14]. Pseudo-random functions and some simple bit operations for designing protocols for security standards were attracting more and more attention.

# 3 Improved Tag Grouping Proof Protocol

## 3.1 Prerequisites and Symbols

This agreement did not require trusted third party to support only tags, readers, and databases. The database was usually a trusted entity that was physically secured and difficult to be invaded in RFID system applications. It often stored secret information (eg: keys, identities, *etc.*) needed by the system. The reader wired connection to which the database it was connected could be regarded as a secure communication channel. Assuming that the keys stored in the tag were difficult to steal, side channel attacks [15] and physical cloning attacks [7] were beyond the scope of this article. And for the wireless connection between the reader and the tag: it was precisely due to the openness of this communication, there were a large range of attack behaviors applicable between them, which could be regarded as an insecure communication channel. The basic characteristics of the communication with the general tag grouping proof protocol were the same. It was assumed that this protocol would be completed within a predetermined time interval. The symbols appeared in the agreement were listed in Table 1.

Table 1: Symbol description

| Symbol | Meaning |
|---|---|
| T | Tag |
| R | Reader |
| D | Database |
| $PID_T$ | Tag pseudonym ID |
| $PID_R$ | Reader pseudonym ID |
| $K_i$ | Tag key |
| $K_R$ | Reader key |
| $K_g$ | Group tags Shared Key |
| $N_1, N_2, N_3$ | Removed |
| $A-G$, $P$, $m_i$ | Three-way communication data |
| $Mark$ | Authorization mark |
| $g_k(x,y)$ | Pseudo-random function based on shared key |

## 3.2 Agreement Specific Certification Process

This agreement was divided into five stages, i.e. initialization, authorization, mutual authentication, grouping proof generation and authentication, and key update. Figure 1 shows the specific implementation flow of the improved fully-certified RFID tag grouping proof generation protocol, whose process is explained as follows:

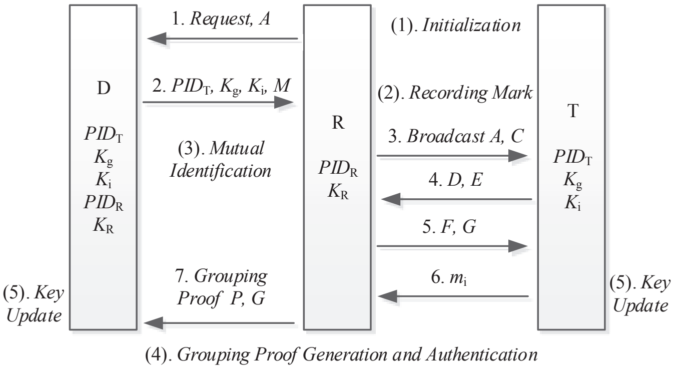

Figure 1: Improved RFID tag grouping proof generation protocol

### 3.2.1 Initialization

The three parties shared a one-way pseudo-random function with low complexity. The single tag in each tag group in the database corresponded to $\{PID_T, K_i, K_g\}$. The corresponding information of the reader was recorded as $\{PID_R, K_R\}$. The group single label stored its own pseudonym identifier $PID_T$ and key information $K_i$ and group shared key $K_g$; the reader/writer stored its own pseudonym identifier $PID_R$ and key information $K_R$.

### 3.2.2 Authorization

Database pre-authenticated readers, generating authorization identifiers $Mark$, in preparation for subsequent group certification generation. The specific process was as follows:

**Step 1.** The reader first sent a group tag authentication authorization requesting a pre-authentication message A from the database.

$$A = K_{Ri} \oplus PID_{Ri}.$$

**Step 2.** After the database received the request instruction, it calculated and tried to find whether there was a reader record equal to $A$ according to all the reader information stored in it. If it did not exist, it meant the reader might be impersonated and the authorization would be terminated. If it existed, it meant that the reader was legitimate, and the database would send pre-authenticated group tag information $\{(PID_{Ti}, K_i)|1 \leq i \leq n, K_g\}$ and message $M$ to the reader. The message M was composed of the authorization identifier Mark generated by the database random number generator.

$$M = \mathrm{g}_{K_g}(K_g) \oplus Mark.$$

**Step 3.** The reader sequentially stored the group tag information transmitted from the database, and decrypted the authorization flag $Mark$ using the received group key $K_g$, and the reader would obtain the authentication and authorization successfully.

### 3.2.3 Mutual Authentication

The authorized reader started the process of mutual identification and identification with the group tag, which was the basis for generating the grouping proof $P$. Assuming that one tag $PID_{T_i}$ was selected in the group tag, the specific authentication process was as follows:

**Step 1.** The reader used the random number generator to generate a random number $N_1$ and broadcasted the encrypted messages $B$, $C$ to the tag.

$$B = \mathrm{g}_{Kg}(K_g \oplus PID_{Ti}), C = K_g \oplus N_1.$$

**Step 2.** After the tag in the group received the message broadcasted by the reader, each tag in the group was calculated using its own key $K_i$ and pseudonym $PID_{T_i}$. When the calculated message B was found to be equal to the received B, the tag was activated, and the random number $N_1$ ($N_1 = C \oplus K_g$) was decrypted to generate a message to act as authenticated $D$. Subsequently, the activated tag generated a random number $N_2$ and an encrypted message $E$, and finally $D$ and $E$ were sent to the reader.

$$D = \mathrm{g}_{K_{Ti}}(PID_{Ti} \oplus K_{Ti} \oplus N_1), E = K_g \oplus N_2.$$

**Step 3.** When the reader received the replying message from the activated tag, it would verify whether $D' = D$ or not. If they were not equal, the label was illegal and the agreement was terminated. If they were equal, the reader would authenticate the tag as a legal tag and the protocol would continue. The reader would decrypt and obtain the random number $N_2$ ($N_2 = E \oplus K_g$ ), and generate the information authenticated $F$, and continue to generate the random number $N_3$ and the encrypted message G and send $F$ and $G$ to the tag.

$$F = \mathrm{g}_{K_{Ti}}(PID_{Ti} \oplus K_{Ti} \oplus N_2), G = K_g \oplus N_3.$$

**Step 4.** After the activated tag received the reader's message, it used its own $PID_T$, $K_i$ message to verify whether $F' = F$ or not. If not, the tag verification reader failed and the authentication terminated; if they were equal, the tag was successfully authenticated and the reader performed subsequent calculations. The tag decrypted the random number $N_3$ ($N_3 = G \oplus K_g$), updating the pseudonym identifier $PID_T$ ( $PID_{Ti} = \mathrm{g}_{K_{Ti}}(PID_{Ti} \oplus K_{Ti}) \oplus N_3$) and the key information $K_i$ ($K_{\mathrm{T}i} = K_{Ti} \oplus N_3$ ), and generating the group authentication and identification factor $m_i$ and sent it to the reader.

$$m_i = \mathrm{g}_{K_{Ti}}(PID_{Ti} \oplus K_{Ti} \oplus N_3).$$

## 3.3 Grouping Proof Generation and Authentication

**Step 1.** All tags in the group repeated the mutual identification process; activating all tags in the group, the reader would receive all tag authentication identification factors $m_i$ within a specified time. Generating group certification message $P$ after successful reception, the reader would finally send $P$ to the database.

$$P = \mathrm{g}_{K_g}(K_g || Mark || m_1 \oplus m_2 \ldots \oplus m_n).$$

**Step 2.** After the database received the grouping proof message, it first calculated the grouping proof $P$ to determine whether it was equal to the grouping proof value $P$ from the reader. If they were equal, the database verified the grouping proof to be successful and proved that the group tag existed at the same time. Afterwards, by decrypting the random number $N_3$ ($N_3 = G \oplus K_g$ ), the tag pseudonym identifier $PID_T$ and the key information $K_i$ were updated. At this point, the protocol was completed; if they were not equal, the verification failed and the protocol terminated.

$$\begin{aligned} PID_{Ti} &= \mathrm{g}_{K_{Ti}}(PID_{Ti} \oplus K_{Ti}) \oplus N_3, \\ K_{\mathrm{T}i} &= K_{Ti} \oplus N_3. \end{aligned}$$

## 3.4 Key Update

This process usually refers to the updating of the secret information (pseudonym, key) in the tag and database, which had already been described in the mutual authentication and grouping proof generation and authentication phases, and would not be repeated here.

# 4 Improved Protocol GNY Logic Proof

GNY logic was a proof of security logic form proposed by L. Gong, R. Needham and R. Yahalom *et al.* It was the most direct and simplest method of analysis. In the field of formal verification of RFID algorithms it had a more widespread application.

Its basic idea: Firstly, the algorithm operating environment and the entities involved in communication should be initially idealized; Secondly, it was necessary to define a reasonable and safe proof target according to the algorithm application requirements. In addition, the entire algorithm process was simulated using GNY logic language rules; finally, a number of GNY axioms and rules were used to derive the correct target from the algorithm process [16]. The GNY logic had a total of nearly 50 inference rules [4]. The relevant rules used in this paper are described as following:

Freshness rules $F_1$: $\frac{P|\equiv \#X}{P|\equiv \#(X,Y), P|\equiv \#F(X)}$.

Message interpretation rules $I_1$:

$$\frac{P \lhd *(X)_K, P \in K, P|\equiv P \overset{K}{\longleftrightarrow} Q, P|\equiv \phi(X), P|\equiv \#(X,K)}{P|\equiv Q|\sim X, P|\equiv Q|\sim (X)_K, P|\equiv Q \in K}.$$

Have rules $P_1$: $\frac{P \lhd X}{P \ni X}$.

Identifiable rules $R_6$: $\frac{P \ni H(X)}{P|\equiv \phi(X)}$.

### 4.1 Idealized Model

Using $T$ for the label, $R$ for the reader, $D$ for the database, $gk()$ denoted a one-way pseudo-random function encrypted with the key $K_i$, $g_{K_g}()$ denoted a one-way pseudo-random function encrypted with the key $K_g$, the idealized model of this agreement was described as follows:

$M_1 : D \triangleleft [K_R, PID_R]$
$M_2 : R \triangleleft *[PID_T, K_T, K_g], [Mark]$
$M_3 : T \triangleleft *[N_1], g_{K_g}(K_g \oplus PID_T)$
$M_4 : R \triangleleft *[N_2], \mathrm{g}_{K_T}(PID_T \oplus K_T \oplus N_1)$
$M_5 : T \triangleleft *[N_3], \mathrm{g}_{K_T}(PID_T \oplus K_T \oplus N_2)$
$M_6 : R \triangleleft *\mathrm{g}_{K_T}(PID_T \oplus K_T \oplus N_3)$
$M_7 : D \triangleleft *[N_3], \mathrm{g}_{K_g}(K_g||Mark||m_1 \oplus m_2 \oplus \ldots \oplus m_n).$

### 4.2 Initialization Assumption

$X_1 : T \in (PID_T, K_T, K_g), g_{K_T}(), g_{K_g}()$
$X_2 : R \in (PID_R, K_R, K_g), g_{K_g}()$
$X_3 : D \in (PID_T, K_T, K_g), (PID_R, K_R), g_{K_T}(), g_{K_g}()$
$X_4 : R \in N_1, N_3$
$X_5 : T \in N_2$
$X_6 : R \mid\equiv \phi(PID_T, K_g), \phi(PID_T \oplus K_T \oplus N_2),$
$\phi(K_g||Mark||m_1 \oplus m_2 \oplus \ldots \oplus m_n)$
$X_7 : T \mid\equiv \phi(PID_T \oplus K_T \oplus N_1), \phi(PID_T \oplus K_T \oplus N_3).$

### 4.3 Expected Goals

Identification and certification of D-to-R identity information:

$$D_1 : D \mid\equiv \phi(PID_R, K_R).$$

Identification and certification of R-to-T identity information:

$$D_2 : R \mid\equiv T \mid\sim \#g_{K_T}(PID_T \oplus K_T \oplus N_1).$$

Identification and certification of T-to-R identity information:

$$D_3 : T \mid\equiv R \mid\sim \#g_{K_T}(PID_T \oplus K_T \oplus N_2).$$

$D$ authenticated the received group certification:

$$D_4 : D \mid\equiv R \mid\sim \mathrm{g}_{K_g}(K_g||Mark||m_1 \oplus m_2 \oplus \ldots \oplus m_n).$$

That was, if database $D$ calculates $P'$ to be the same as received $P$ (P $=\mathrm{g}_{K_g}(K_g||Mark||m_1 \oplus m_2 \oplus \ldots \oplus m_n)$), then it was believed that tag information existed at the same time. The following used $P$ for $(K_g||Mark||m_1 \oplus m_2 \oplus \ldots \oplus m_n)$.

### 4.4 Reasoning Proof

After receiving the message $M_1$, the database $D$ queries the information to see if there was matching information $\{PID_R, K_R\}$. If the matching was successful, the database recognizes that the reader $R$ was successful, that was $D \mid\equiv \phi(PID_R, K_R)$, the target $D_1$ was implemented.

could be obtained by the message $M_2$, the reader $R$ to obtain the group tag message from the database, available $R \in K_T$ (1), and in accordance with the agreement assumption $X_1(T \in K_T\ )$, we could know $R| \equiv R \xleftrightarrow{K} T$ (2);

From the message $M_4$, we get $R \triangleleft *\mathrm{g}_{K_T}(PID_T \oplus K_T \oplus N_1)$, that was $R \triangleleft *(PID_T \oplus K_T \oplus N_1)_{K_T}$ (3), and the reader believed the freshness of the $\mathrm{g}_{K_T}(PID_T \oplus K_T \oplus N_1)$ message, that was $R| \equiv \#(PID_T \oplus K_T \oplus N_1)$. and then according to the freshness rule $F_1$, we could get $R| \equiv \#((PID_T \oplus K_T \oplus N_1), K_T)$ (4). According to the initialization assumptions $X_4$ $(R \in N_1)$ and $X_6(R| \equiv \phi(PID_T \oplus K_T \oplus N_2))$, we could see $R| \equiv \phi(PID_T \oplus K_T \oplus N_1)$ (5). So by the equations (1)-(5) and the message interpretation rule $I_1$, you could get equation $R| \equiv T \sim (PID_T \oplus K_T \oplus N_1)_{K_T}$, and the transformation could get $R| \equiv T \sim g_{K_T}(PID_T \oplus K_T \oplus N_1)$ (6).

Finally, based on Eqn. (6) and the communication message $M_4$, $R| \equiv T \sim \#g_{K_T}(PID_T \oplus K_T \oplus N_1)$ could be obtained and the target $D_2$ was verified.

The target $D_3$ proved that the process was the same as the target $D_2$, and the proof was not repeated.

According to the initial hypothesis $X3$, $D \in K_g$ (7) was available, and because the background database stored all the reader and tag information, $D| \equiv D \xleftrightarrow{K} R$ (8) was obtained.

The message $M_7$ could be obtained $D \triangleleft *(P)_{K_g}$ (9), and according to the belief of the database to the freshness of the message $M_7$, $D| \equiv \#(P)$ could be obtained, and then according to the freshness rule $F_1$, $D| \equiv \#(P, K_g)$ (10) could be obtained.

According to the initial hypothesis $X_3$, $D \in (P)$ was owned by the rule $P_1$ and the message $M_7$, and $D| \equiv \phi(P)$ (11) was obtained according to the recognizable rule $R_6$.By (7)-(11) formula and message interpretation rules $I_1$, finally get $D \mid\equiv R \mid\sim (P)_{K_g}$, target $D_4$ get evidence.

In summary, the four goals of the improvement agreement had been verified.

## 5 Improved Protocol Security Analysis

### 5.1 Mutual Authentication Mechanism

The mutual authentication mechanism between the reader and the tag was the basis. The reader authenticated the message $D$ to verify the legitimacy of the tag, and the tag authenticated the message $F$ to verify the legitimacy of the reader. In addition, the database first authenticated the message $A$ sent from the reader. After the authentication passed, the database generated an authorization identifier $Mark$, which was then transmitted to the reader and the reader decrypted and retained. Once the reader was impersonated, the grouping proof generated would be wrong when the $Mark$ identifier was unacquirable, so the database would fail authentication, and the attacker would fail.

## 5.2 Replay Attack

In most cases, the attackers would unexceptionally disguised as a reader to replay attack on tags, the process was improved as follows: The attacker eavesdroped on the communication channel, and acquired the communication data $B$, $C$, and then masqueraded as a normal reader to replay the intercepted messages $B'$ and $C'$ to the tag.

The tag used its own stored information to calculate whether the comparisons $B$ and $B'$ were equal. If equal, the tags were activated. However, when the tag obtained the random number $N_1'$ sent from the attacker, the generated verification data $D$ and the random number encrypted data $E$ were sent to the attacker.

Since the attacker could not know the secret information $PID_T$, $K_i$, $K_g$ and could not calculate and generate the correct response message $F$, $G$, the tag authentication failed and the protocol terminated.

Therefore, this protocol could resist replay attacks.

## 5.3 Brute-force Attack

Reference [3] relied solely on the selected encryption algorithm to ensure the security of the protocol, which was not rigorous. Therefore, compared with the Reference [3], this protocol encrypted all publicly transmitted random numbers to prevent attackers from eavesdropping on the obtained communication data and performing brute force attacks. Because in general cases, the label group key was already written, and the third party could not know it, and the random numbers in messages $C$, $E$, and $G$ ($C = K_g \oplus N_1$, $E = K_g \oplus N_2$ and $G = K_g \oplus N_3$ ) were encrypted and transmitted by using the group key. The internal algorithms and data in messages $C$, $E$, and $G$ were not available to the attacker. Then the attacker could not obtain the exact value of the random number to decrypt the authenticated messages $D$ and $E$( $D = g_{K_{Ti}}(PID_{Ti} \oplus K_{Ti} \oplus N_1)$, $E = K_g \oplus N_2$). Therefore, this agreement could resist brute force attacks.

## 5.4 Impersonation Attack

An attacker could impersonate a reader or tag to attack. In the case of attacker pretending to be a reader, the reader and the database were wired securely during the authorization phase, so the attacker could not obtain the reader's identity and key information. The database pre-authentication failed, the protocol terminated, and the attack failed. Even if the attacker finally stole message $A$ under secure wired communication and finally passed the authorization, due to the failure of the attacker to obtain $K_g$, it still could not decrypt the authorization $ID$ mark under the encryption condition, so the correct group label authentication message $P$ could not be generated. The authentication failed and the agreement terminated. If the attacker impersonates a legitimate tag, although the communication information $B$, $C$, $F$, and $G$ could be obtained, since all the information was transmitted encrypted, the attacker could not obtain any of the $PID_T$, $K_i$, and $K_g$ data. This meant that the correct random number $N_1$ could not be obtained by decrypting $B$, $C$, $F$, and $G$. In the end, the reader could not be provided with the correct data to be verified $D$, and the legal reader authentication failed and the protocol terminated. Therefore, this protocol could resist replay attacks.

## 5.5 Tracking Attack

In this agreement, although the tag ID was not changed, the tag pseudonym ID was used instead of the real ID for calculation during the entire protocol communication process. And the pseudonym and key information $PID_T$, $K_i$, $K_g$ were updated and stored by the random number $N_3$ after the communication in each round. Each round of authentication of tag and reader also generated random numbers $N_1$ and $N_2$, which made this protocol somewhat fresh and unpredictable; therefore, the attacker could not trace the identity information of the tag only through eavesdropping or interception, and so on.

## 5.6 Proof of Forgery Attack

If an attacker had to forge a valid tag grouping proof to pass the final verification of the database, then it was necessary to:

Forge all valid grouping proof factors, that was, to obtain the pseudonym and private key information ($PID_T$ and $K_i$) for all tags. However, it had been proven in the above attack statement that the probability of obtaining pseudonyms and private key information for all tags was impracticable; at the same time, it was also necessary to obtain the value of the random number $N_3$. However, in this protocol, $N_3$ encrypts the entire transmission, and the attacker could not crack the key without knowing the $K_g$ key.

Get the database authorization logo, but the database and the reader were wired and securely transmitted. Even if it was insecure, the transmission of $Mark$ messages was also performed using multiple bit operations and random number encryption. Therefore, even when the attacker faked the reader, without knowing $Mark$, the attack was not likely to succeed. Therefore, this agreement could resist the proof of forgery attack.

Table 2: Security comparison of related protocols

| Type of attacks | [2] | [3] | [10] | [11] | This article |
|---|---|---|---|---|---|
| No trusted third party | × | × | × | × | ✓ |
| Mutual authentication | ✓ | ✓ | × | ✓ | ✓ |
| Replay attack | × | × | ✓ | ✓ | ✓ |
| Brute-force attack | ✓ | ✓ | ✓ | × | ✓ |
| Impersonation attack | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tracking attack | × | ✓ | ✓ | × | ✓ |
| Proof of forgery attack | ✓ | ✓ | × | × | ✓ |

In summary, Table 2 gave a comparison of the security of this protocol and other RFID tag grouping proof protocols. Among them, ✓ meant that this type of attack could be resisted, and × meant that it could not resist this type of attack.

## 6 Performance Analysis

This section mainly analyzed the performance of this protocol in terms of the amount of tag calculations, storage capacity, and traffic volume. In accordance with the Gen-2 standard, the complexity and overhead of heavyweight encryption algorithms such as hash functions and elliptic ECC functions were significantly higher than those of pseudo-random functions, and the computational overhead required for lightweight MAC operations and pseudo-random encryption algorithms was comparable. [14]. See Table 3 for the performance of related protocols.

Tag computation overhead: The protocol tag side contained only lightweight pseudo-random function operations that satisfied the low-cost requirements of the Gen-2 standard and a simple bit operation (XOR operation), and only $XOR$ operation was performed during the authentication process.

Tag communication overhead: In each round of communication, the single tag in the group conducted mainly two communications of $D$, $E$, and $m_i$.

Tag storage overhead: In the protocol initialization process, the single tag in the group stored tag pseudonyms, keys, and group key information $\{PID_T, K_i, K_g\}$.

Table 3: Performance comparison of related protocol tag

| Related agreements | Computation overhead | Communication overhead | Storage overhead |
|---|---|---|---|
| [2] | $3M()$ | $2I$ | $L$ |
| [3] | $H() + E() + 2X$ | $3I$ | $2L$ |
| [10] | $3H() + 3M() + 3X$ | $2I$ | $L$ |
| [11] | $2H() + 4g()$ | $2I$ | $2L$ |
| This article | $4X + 5g()$ | $3I$ | $2L$ |

In Table 3, $H()$ denoted a hash operation, $X$ denoted an exclusive-OR operation, $g()$ denoted a pseudo-random operation, $M()$ denoted a MAC operation, $E()$ denoted an elliptic curve operation, and the unit length of an ID and a key were both $I$, the unit of single communication overhead was $L$.

## 7 Conclusions

This paper proposed a lightweight, fully-certified RFID tag grouping proof protocol that met the Gen-2 standard. The protocol did not require the verification support of a trusted third party. It only required tag to support one-way pseudo-random function operations and simple XOR operations. Based on the reader and tag authentication, the database authorization identifier Mark was introduced to further encrypt and verify the generated tag grouping proof. GNY formal logic proved that the agreement was feasible and complete; after attack description analysis, the protocol met tag untraceability and could resist tag and reader impersonation attack and message replay attack. The protocol used encrypted random numbers and Mark authorization token to resist the tag proof of forgery attack and rapid brute force attack. Finally, according to the comparison of tag calculation, communication and storage overhead between the related protocols, it was proved that this protocol was better than the other tag grouping proof protocols presented. The next step was to add tag collision to the protocol for further study.

## Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant 51404216.

## References

[1] 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.

[2] Z. S. Da and G. Z. Ze, "Improved RFID yoking proof protocol," *Computer Engineering and Design*, vol. 38, no. 8, pp. 2076–2080, 2017.

[3] Y. M. Guo, S. D. Li, Z. H. Chen, and X. Liu, "A lightweight privacy-preserving grouping proof protocol for RFID systems," *Acta Electronica Sinica*, vol. 43, no. 2, pp. 289–292, 2015.

[4] H. F. Hong and H. L. Tian, *Privacy Protection Security Protocol Study (In Chinese)*, Beijing: Science Press, 2015.

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

[6] Z. Z. Kai, P. Tao, A. Liang, M G, and *et al.*, "High-reliable RFID grouping tag proof protocol," *Computer Engineering and Design*, vol. 39, no. 2, pp. 150–154, 2013.

[7] T. Korak, T. Plos, and A. Zankl, "Minimizing the costs of side-channel analysis resistance evaluations in early design steps," in *Eighth International Conference on Availability, Reliability and Security (ARES'13)*, pp. 169–177, 2013.

[8] Q. Lin and F. Zhang, "Ecc-based grouping-proof RFID for inpatient medication safety," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3527–3531, 2012.

[9] J. Liu, M. Chen, B. Xiao, F. Zhu, S. Chen, and L. Chen, "Efficient RFID grouping protocols," *IEEE/ACM transactions on networking*, no. 5, pp. 3177–3190, 2016.

[10] M. Safkhani, N. Bagheri, M. Hosseinzadeh, M. Eslamnezhad Namin, and S. Rostampour, "On the security of an RFID-based parking lot management system," *International Journal of Communication Systems*, vol. 30, no. 15, p. e3313, 2017.

[11] C. H. Wei, M. S. Hwang, A. Y. H. Chin, "A mutual authentication protocol for RFID", *IEEE IT Professional*, vol. 13, no. 2, pp. 20–24, Mar. 2011.

[12] C. H. Wei, M. S. Hwang, Augustin Y. H. Chin, "An authentication protocol for low-cost RFID tags", *International Journal of Mobile Communications*, vol. 9, no. 2, pp. 208–223, 2011.

[13] C. H. Wei, M. S. Hwang, Augustin Y. H. Chin, "An improved authentication protocol for mobile agent device in RFID," *International Journal of Mobile Communications*, vol. 10, no. 5, pp. 508–520, 2012.

[14] C. H. Wei, M. S. Hwang, and A. Y. h. Chin, "Security analysis of an enhanced mobile agent device for RFID privacy protection," *IETE Technical Review*, vol. 32, no. 3, pp. 183–187, 2015.

[15] C. H. Wei, M. S. Hwang, and A. Y. H. Chin, "A secure privacy and authentication protocol for passive RFID tags," *International Journal of Mobile Communications*, vol. 15, no. 3, pp. 266–277, 2017.

[16] Z. Z. Wen, B. W. Li, F. L. Yi, and *et al.*, *Security Protocol Design and Analysis (In Chinese)*, Beijing: National Defense Industry Press, 2015.

[17] Z. Yang, G. P. Chang, and F. D. Hong, "A disordered and anonymous RFID grouping proof scheme," *Computer Engineering*, vol. 38, no. 20, pp. 85–88, 2012.

[18] B. Yuan and J. Liu, "A universally composable secure grouping-proof protocol for RFID tags," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 6, pp. 1872–1883, 2016.

[19] Q. Zhang, X. Hu, J. Wei, and W. Liu, "Universally composable three-party password authenticated key exchange," in *International Conference on Cloud Computing and Security*, pp. 123–137, 2017.

# Biography

**Gao-feng Shen** received his master degree in computer specialty from Huazhong University of Science and Technology (China) in June 2005. He is a lecturer in computer science in School Of Computer and Communication Engineering, Zhengzhou University of Light Industry. His current research interest fields include algorithm design, database and its application, data mining.

**Shu-min Gu** received her master degree in computation mathematics from Henan Normal University (China) in June 2007. She is an adjunct professor in mathematics in school of information and business, Zhongyuan University of Technology. Her current research interest fields include numerical solution of differential equation.

**Dao-wei Liu** received a master's degree in School of Computers from Guangdong University of Technology (China) in June 2016. He is a teacher in department of computer science and engineering, Guangzhou College of Technology and Business. His current research interest fields include information security.