IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 December 2022, accepted 3 January 2023, date of publication 9 January 2023, date of current version 17 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3234959

RESEARCH ARTICLE

Efficient Mobile RFID Authentication Protocol for Smart Logistics Targets Tracking

CONG XU, WENXUE WEI, AND SHUANGSHUANG ZHENG
Department of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266590, China
Corresponding author: Wenxue Wei (wwxjyh@163.com)

ABSTRACT Target tracking is one of the problems existing in the supply chain management. The use of radio frequency identification (RFID) in target tracking helps improve the monitoring accuracy and status visibility of the tracked target. For mobile RFID system, its three entities have to authenticate each other's identity in order to guarantee the data transmission security. The mobile RFID authentication protocol cannot achieve both high security and low complexity at the same time. For this problem, a new efficiency mobile RFID authentication protocol is proposed in this paper, which implements secure authentication among different communication entities by different operation modes. For example, the protocol adopts Hash Function between reader and cloud server, and exchange-cross bitwise operation between tag and cloud server, to achieve low computing cost at tag-end while improving the security of mobile communication data. At the cloud server end, the protocol proposed in this paper adopts index data table as the storage mode, which further improves the could server efficiency in retrieving the authentication of tags and readers, and reduces the risks of sensitive data disclosure. According to the security analysis, this protocol can resist impersonation attack, replay attack, trace attack and other attacks launched by attackers. Its security performance is further proved by BAN logic, proverif tool and random oracle model. On the other hand, the simple operation at the tag-end of the protocol lowers the tag cost to a larger extent.

INDEX TERMS Authentication protocol, BAN logic, mobile system, RFID, target tracking.

I. INTRODUCTION

Traditional target tracking in the logistics management system is mainly to track the location of the cargo, not the status of the goods. Therefore traditional target tracking is not applicable for cold chain and pharmaceutical logistics processes. During recent years, a target tracking system based on RFID sensor network has been proposed, which achieves position and property tracking of the mobile targets by RFID and sensor technologies. It enables legal users to completely and visually master the cargo status, thereby delivering cargo in accurate amount and appropriate conditions at specific site [1]. In the target tracking system based on RFID sensor network, sensors are responsible for searching information around the target and write into RFID tag. Then the RFID reader inside the smart phone of the driver sends the private data collected by the sensors to the cloud server. RFID, which is featured in non-contact recognition, satisfactory applicability in various environments, and large data capacity [2], [3], improves the visibility of object status in the tracking system, and greatly enhances the performance of the target tracking system [4]. The market scale of using RFID in smart logistics system in China had been expanded from 68 billion in 2018 to 100 billion in 2020.

With the increase of use, data transmitted in RFID system has been expanding day by day, which highlights the urgent demands on data security and privacy protection [5], [6]. Impersonated tags or the interception of tag information may lead to cargo data disclosure, threatening user data security and endangering economic benefits [7]. To improve the data transmission security, identities of all related communication entities in RFID system must be authenticated to achieve mutual trust among communication entities [8]. Most identity authentication protocols are based on an assumption that the

The associate editor coordinating the review of this manuscript and approving it for publication was Giorgio Montisci.

communication channel between reader and server is private and secure. Therefore, the mutual authentication is only achieved between two entities of reader and tag, such as the EPC gen2 protocol [9]. However in the target tracking system, data between RFID reader and cloud server are transmitted by wireless network, for which, the channel is not secure. In this case, a protocol that can achieve mutual authentication a mong the three communication entities, namely tag, reader, and cloud server, is required.

The authentication protocol for mobile RFID system is a protocol to achieve mutual authentication among three entities in the system. It is a security measure to prevent fake entity from passing the RFID target detection, which is significantly important for protecting RFID system security and data privacy. According to the computing costs, the authentication protocols can be divided into three types: heavy-weight protocol, light-weight protocol, and ultra-light-weight protocol [10]. The heavy-weight protocol has been eliminated from RFID system because of its complicated encryption operation. The light-weight protocol is designed to execute operations at the tag-end, such as the one-way Hash Function, Physical Unclonable Function, and pseudo random number generating. The ultra-light-weight protocol is designed to run simple bitwise operation at the tag-end, such as ''and'', ''xor'', ''bit-replacing'', and ''shift'', etc.

As for the light-weight authentication protocol based on Physical Unclonable Function mentioned in the literature [11], although the key generated by the Physical Unclonable Function cannot be copied, the replay of the message intercepted during the communication process can result in inconsistent information between tag and key in cloud server database, making it unable to resist desynchronized attack initiated by the attacker. Information in the light-weight authentication protocol mentioned in the literature [12] are mostly transmitted by plain text, including the generated random numbers and the random numbers used by encryption. Attackers can acquire the private information of the encrypted tags in the authentication message by method of exhaustion, which is actually a loophole for brute force attack. For the light-weight mobile authentication protocol based on bitwise operation mentioned in the literature [13], although the bitwise operation can reduce computing cost and communication cost, the random numbers used for computing the authentication information is transmitted by plain text, so that the attackers can acquire the key information of the tag and the reader-writer, it cannot resist impersonated attack. For the light-weight authentication protocol based on Hash function mentioned in the literature [14], the way of using Hash function to compute the authentication information improves the security of the RFID system, but the complicated Hash operation for the tags also enhances the computing cost.

It has been found from the literature [15], [16], [17], [18] that, SASI protocol [19] frequently uses ''or'' and ''and'' operations when generating secret information, so that its computing results are highly correlated and cannot resist tracked attack, denial of service (DoS) attack, and algebraic attack. It has been pointed out in the literature [20], [21] that, Gossamer protocol [22] cannot resist the DOS attack. And due to its complicated computing and significant power dissipation, it is not suitable for low-cost tag use. In [23], by improving the SASI protocol [19] and making up the security loophole in Gossamer protocol [22], a new ultra-light-weight RFID authentication scheme is put forward, in which, the reader and the background database are not mutually authenticated and can easily be subjected to the impersonated attack of the reader and tag. In [24], a new ultra-light-weight mobile authentication protocol is proposed, which encrypts the transmitted information based on bit rearrangement operation to reduce the protocol computation cost. However, the tag information in the tag identification phase is transmitted in clear text, which is easy to be intercepted by attackers to launch tracking attacks, and the protocol security cannot be guaranteed.A new ultra-light-weight authentication protocol is proposed in literature [25] based on word synthetic operation, which encrypts information by word synthesis. It greatly reduces computational complexity and protocol cost. However the reader of this protocol doesn't authenticate the tag, so the both-way authentication among all communication entities isn't achieved. Literature [26] describes a new ultra-light-weight authentication protocol based on bit replacement, which encrypts the transmitted information by bit replacement. But it cannot guarantee the timeliness of information transmission, and cannot resist replay attack. Moreover, the server bears too much operation loads when verifying reader and tag, so that it could cost a long authentication time if it needs to verify a large number of tags.

For the above problems, an efficient ultra-light-weight mobile authentication protocol is proposed in this paper, which implements secure authentication among different communication entities by different operation modes, and adopts index data table to store ciphertext at cloud server end for authentication purpose. It helps improve the security of the protocol and reduces the tag complexity, and is suitable for being used in low-cost RFID system. This scheme consumes only a little computing and storage resources, satisfies the demands on tag anonymity and both-way entity authentication, and resist impersonated attacks, replay attacks, tracked attacks, and brute force attacks by timestamps and random numbers.

II. DETAILED DESCRIPTION ABOUT THE PROTOCOL

A new ultra-light-weight mobile RFID two-way authentication protocol is proposed in this paper. Similar to other mobile RFID authentication protocols, the protocol in this paper is also designed based on the assumption that the tag, reader, and cloud server communicate via wireless transmission, bearing the risks of being attacked. Both the cloud server and the reader have certain computing capability and large storage space, but the tag is weak in the two aspects [27].

TABLE 1. Notations.

Notation	Description
Query	Inquiry message
$STID$	The pseudo ID of Tag
$SRID$	The pseudo ID of Reader
key_t^{old}	The previous shared key of Tag and Cloud Server
key_t^{new}	The current shared key of Tag and Cloud Server
key_r^{old}	The previous shared key of Reader and Cloud Server
key_r^{new}	The current shared key of Reader and Cloud Server
a	A random number generated by Tag
b	A random number generated by Cloud Server
$Rot(X, Y)$	The operation of left rotation,$x \ll W(Y)$
$Eac(X, Y)$	The operation of exchange-cross
$h()$	Hash Function
$+$	The bitwise AND operation
$\oplus$	The bitwise XOR operation
$\|\|$	The bitwise concatenation operation

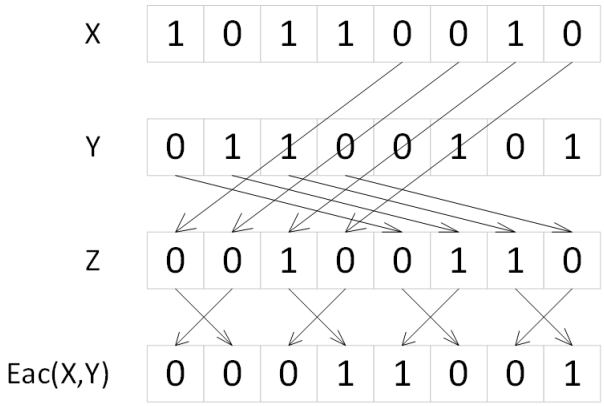

FIGURE 1. Computation of the example.

A. INSTRUCTIONS TO SYMBOLS

This section gives the specific meanings of all symbols used in the protocol, as shown in Tables 1.

As for the protocol proposed in this paper, the exchange-cross bitwise operation is adopted at its tag end. Multiple protocols that have been proposed adopt left-shift operation: suppose the data length is L, when the hamming distance of the data approaches 0 or L, the attacker needs only to move data less than $L/2$ to acquire plain text data. This means great probability of attack success. However the adoption of first-exchange-then-cross operation tackles the above shortcoming. It was firstly proposed by literature [28].

$Eac(X, Y)$ is defined as below: X, Y are two binary sequences with the same number of bits. The number of bits is even. Put the latter $L/2$ bits of the binary sequence X at the front of thew newly composed sequence Z, and put the first $L/2$ bits of the binary sequence Y at the latter of the sequence Z. This is how the new sequence Z is formed. Then the sequence Z shall be traversed, cross and exchange the number on the odd bit with the number of adjacent even bit of Z to obtain the cross-exchange operation results. For example, if $X = 10110010$, $Y = 01100101$, and $L = 8$, then according to the above-mentioned definition, it can be obtained: $Z = 00100110$, $Eac(X, Y) = 00011001$. Specifically, please refer to the Fig. 1.

The exchange-cross operation is implemented based on per-bit operation, which can meet the requirement of reduced computation while ensuring privacy and information security. Compared with the hash function or mode-square operation used at the tag side in other literatures, the exchange-cross operation is less computationally intensive and can largely reduce the computational overhead of tags.

To crack the exchange-cross operation, the attacker has to be able to crack the values of the two numbers involved in the exchange-cross operation. Here, the number of encrypted parameter bits is taken as L=128 bits for cracking analysis. According to the protocol in the text, it is known that: the protocol of the tag, the key and other information are sent in cipher text, that is, it is impossible for the attacker to get the detailed values of these encrypted information. In the premise that the attacker does not obtain the specific values of the encryption parameters, the attacker can only crack according to the known exchange-cross rules, and the correct probability of X and Y obtained after the correct cracking is completed is:

$$P = \frac{1}{2} * \frac{1}{2} * \frac{1}{2} * \ldots\ldots \frac{1}{2} = \frac{1}{2^{128}}$$

For the first 64 bits of X, the probability of correctly breaking each bit is $\frac{1}{2}$, so the probability of getting X correct is $\frac{1}{2^{64}}$, and similarly, the probability of getting the last 64 bits of Y correct is also $\frac{1}{2^{64}}$. In summary, the probability that an attacker wants to correctly break all the bits of the swap-and-cross operation is $\frac{1}{2^{128}}$. If the number of encrypted parameters exceeds 128 bits in the application process, the probability that an attacker can correctly crack it will be smaller than $\frac{1}{2^{64}}$, so the swap-and-cross operation has strong information cracking resistance and can provide the security required for encryption.

B. PROTOCOL DESCRIPTION

This section shows the detailed description about the proposed protocol. It is composed of three stages, including the initial stage, the authentication stage, and the update stage. In the initial stage, the administrator assigns initial values for legal mobile readers and tags; in the authentication stage, mutual authentication is achieved among all three entities of tag, mobile reader and cloud server; and in the update stage, the main task is to update the fake name identifiers and keys for tags,mobile readers and cloud server.

C. INITIAL STAGE

1) TAG

The administrator assigns a pseudonym identifier ($STID$) for each legal tag, and the cloud server generates relevant privacy key_t for it, then calculate message $C = Rot(key_t \oplus STID, key_t)$, $O = ket_t \oplus STID$, C will be used as an index, meanwhile O is stored in the index data table as index content. Through the secure channel, $< STID, ket_t >$ is stored in Tag.

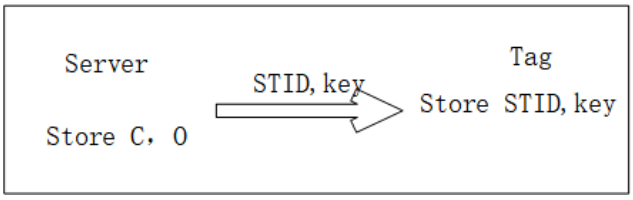

FIGURE 2. Initial phase.

TABLE 2. Index data table.

Index value	Index content
$C_1 = Eac(key_{t1} \oplus STID_1, key_{t1})$	$O_1 = Rot(key_{t1}, STID_1)$
$C_2 = Eac(key_{t2} \oplus STID_2, key_{t2})$	$O_2 = Rot(key_{t2}, STID_2)$
$C_3 = Eac(key_{t3} \oplus STID_3, key_{t3})$	$O_3 = Rot(key_{t3}, STID_3)$
...	...
$C_n = Eac(key_{tn} \oplus STID_n, key_{tn})$	$O_n = Rot(key_{tn}, STID_n)$

Let $STID^{old} = key_t^{old} = C^{old} = O^{old} = 0$ in the server data table.The initialization process is shown in Figure. 2, with the double arrow representing the secure channel.

Since the untrusted cloud server may disclose the stored privacy data, the data in the index data table is stored in form of ciphertext instead of being directly stored as $STID$, key_t. In order to resist synchronous attack, the index value and content of previous round are also stored. The index data table at the cloud server end helps improve the efficiency of data retrieval. And the information stored in form of ciphertext avoids the risk of sensitive data disclosure of the cloud server. Tables 2 shows the detailed information in the index data table, in which, C is used as the index value while the ciphertext O is the index content. In this table, the index value C is selected by the exhaustive search algorithm, while the index content O is effectively and quickly located by the index value C, preventing the cloud server from conducting two exhaustive searches for $STID$. In this process, the search time increases linearly with the increase of the number of RFID tags, which has a certain impact on the scalability of the RFID system. After the ending of each session,key_t and $STID$ need to be updated, which improves the security and ensures accuracy.

2) READER

The administrator assigns a reader pseudonym identifier ($STID$) for each legal mobile reader, and the cloud server generates a relevant privacy key key_r. Similar to the label storage mode, the reader information is also stored by index data table. $h(SRID)$ and $Rot(key_r, SRID)$ are stored in index data table as index value and index content respectively. $< SRID, key_r >$ is stored in reader by secure channel. In the server memory, let $h(SRID^{old}) = Rot(key_r^{old}, SRID^{old}) = 0$.

D. AUTHENTICATION STAGE

The mutual identity authentication process and the communication among tag, reader, and cloud server are introduced in details in this section. The communication this time is firstly initiated by the reader.The detailed authentication process is shown in the Fig. 3.

1) $READER \rightarrow TAG : QUERY, A$

First, the reader generates a random number a. Message A is computed according to the $STID$ stored in reader memory and the generated random number a.Then inquiry message $Query$ and A are sent to the tag.

2) $TAG \rightarrow READER : B, M, t$

After the tag receives the message A, the Hamming weight of the $STID$ stored in the tag is calculated, and a^* is restored from the received message A. Then message $B = Rot(STID, a^* \oplus t)$, $C = Eac(key_t \oplus STID, key_t)$, $M = a^* \oplus C$ is calculated by $STID$, key_t stored in the memory and a^* obtained by restoration. Finally, Message B and M are sent to the reader.

3) $READER \rightarrow CLOUDSERVER : M, N, D, E, F, T_R$

After receiving Message B, since the Hamming weight of $a \oplus t$ is known, the $STID^*$ can be obtained according to the message B.Then it will look for $STID^{new} = STID^*$ in memory. If there is no $STID^{new} = STID^*$, it will keep searching if there's $STID^{old} = STID^*$. If neither exists, it will stop authentication. If there's the required data, it means successful authentication of reader to tag. Then, Message $D = a \oplus h(T_R)$, $E = Rot(h(SRID), a \oplus T_R)$, $F = h(Rot(SRID, key_r) \oplus a)$ will be calculated by $SRID$ and key_r in the memory, the current time T_R, and random number a. Finally, Message M and N of the tag, Message D, E, and F calculated by the reader, and the timestamp T_R of the reader are sent to the server.

4) $CLOUDSERVER \rightarrow READER : G, H, I, T_S$

Once the server receives messages M, N, D, E, F, and T_R, it will check T_R first to see if it satisfies the conditions of $t \leq T_R \leq 2t$ or not. If it satisfies, the server will perform authentication to the reader. The first thing to do is to restore $a^* = D \oplus h(T_R)$. Since the Hamming weight of $a^* \oplus T_R$ is known, the $h(SRID^*)$ can be obtained according to the message E. Then it will look for $h(SRID^{new}) = h(SRID^*)$ in the cloud server database, if there is such data, the $Rot(SRID, key_r)$ that the index corresponds to can be obtained. Then it will compare if the calculated Message $F^* = h(Rot(SRID, key_r) \oplus a)$ is consistent with the received Message F or not. If the two are consistent, the authentication of cloud server to reader passes, otherwise, the authentication fails, and the authentication process ends. If there's no $h(SRID^{new}) = h(SRID^*)$, it will search for $h(SRID^{old}) = h(SRID^*)$ in the database. If the data exists, it will judge if the F^* calculated by index content and the Message F are the same or not. If they are the same, the authentication of cloud server to reader passes and enter the next step of tag authentication, otherwise the authentication fails and ends.

When the server authenticates the tag, the server calculates C^* according to the received Message M and the restored a^*,$C^* = a^* \oplus M$, and looks for $C^{new} = C^*$ in server database.

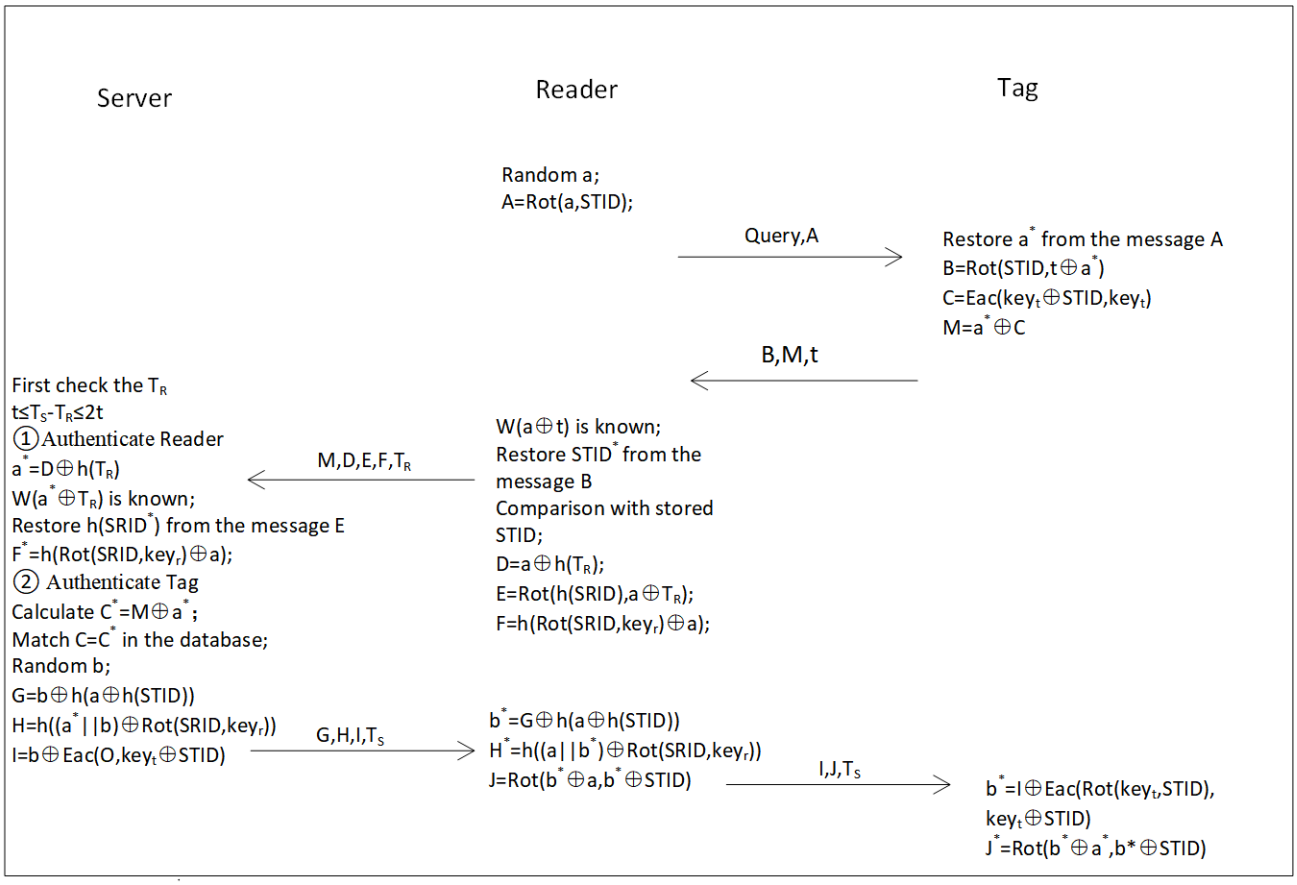

FIGURE 3. Authentication phase.

If there is such data, the server's authentication to tag passes, otherwise, it will search for $C^{old} = C^*$ in the database, if this data exists, the cloud server authentication to tag passes, and let $C^{new} = C^{old}$, $O^{new} = O^{old}$. Otherwise it indicates failure of tag authentication, and the authentication process ends.

Once the cloud server's authentication to tag and reader passes, it will generate a random number b, and calculate Message $G = b \oplus h(a \oplus h(STID))$, $H = h((a^* \parallel b) \oplus Rot(SRID, key_r))$, $I = b \oplus Eac(O, key_t \oplus STID)$, and then send Message G, H, I, and timestamp T_s to the reader.

5) *READER → TAG : I, J, T_S*

After the reader receives the message G, H, it will restore b^* according to the received Message G,$b^* = G \oplus h(a \oplus h(STID))$,then calculate H^* by the key_r stored in the memory, the generated random number b and the restored b^*,$H^* = h((a^* \parallel b^*) \oplus Rot(SRID, key_r))$, and make comparison on the calculated H^* and the received H to see if the two are the same or not. If the two are equal, it means the reader's authentication to server passes. Then calculate $J = Rot(b^* \oplus a, b^* \oplus STID)$, and send Message I, J and T_s to the tag. Otherwise the server authentication fails and the authentication process ends.

6) *TAG*

When the tag receives the Message I, J, it will restore b^* according to the received Message I, $b^* = I \oplus Eac(Rot(key_t \oplus STID), key_t \oplus STID)$,calculate J^* by the stored $STID$ and restored a^* and b^*,$J^* = Rot(b^* \oplus a^*, b^* \oplus STID)$,then compare the calculated J^* and the Message J to see if the two are equal. If they are the same, it means successful authentication of server and reader at tag end, and the authentication process completes. If they are not the same, it means either the reader or the server, or both the two fail the authentication, and the authentication process ends.

E. UPDATE STAGE

The detailed update process is shown in the Fig. 4.

1) *TAG → READER : X_L*

Tag calculate $key_t^{new} = key_t \oplus b^*$, $STID^new = STID + T_s$, $X = Eac(STID^{new} \oplus key_t^{new}, key_t^{new})$,then send the left half of the message X to the Reader for update consistency verification.

2) *READER → CLOUDSERVER : X_L, Y_L*

After the Reader receives the message X_L,it will calculate $SRID^{new} = SRID + T_s$, $STID^{new} = STID + T_s$, $key_r^{new} = key_r \oplus a$, $Y = h(Rot(SRID^{new}, key_r^{new}) \oplus a)$.Finally,Message

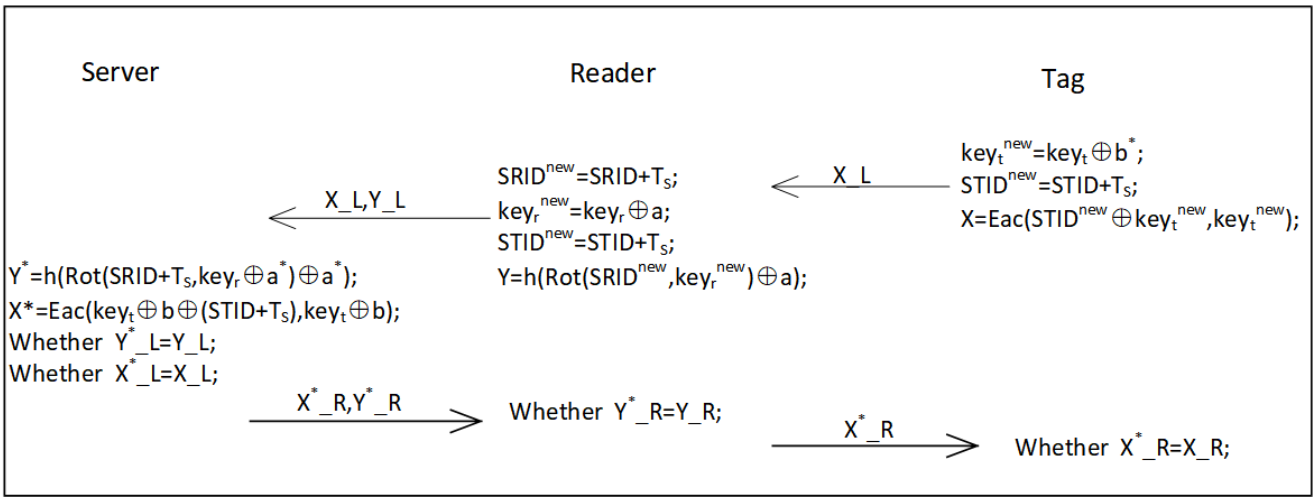

FIGURE 4. Update phase.

X_L of the Tag,the left half of the Message L calculated by the reader are sent to the cloud server.

3) *CLOUDSERVER → READER : X*_R, Y*_R*

After the cloud server receives the message X_L, Y_L, it will calculate $Y^* = h(Rot(SRID + T_s, key_r \oplus a) \oplus a), X^* = Eac(key_t \oplus b^* \oplus (STID + T_s), key_t \oplus b)$,then it will compare if the calculated Message X^*_L, Y^*_L is consistent with the received Message X_L, Y_L or not.If the two are consistent,the cloud server will update the $h(SRID^{new}, Rot(SRID^{new}, key_r^{new}))$, $h(SRID^{old})$, $Rot(SRID^{old}, key_r^{old})$, C^{new}, O^{new}, C^{old}, O^{old} in database, otherwise not updated.

4) *READER → TAG : X*_R*

After receiving the message Y^*_R,the received Y^*_R and the Y_R calculated by the reader are compared,and if they are equal,the server is proved to be consistent with its updated content,then the reader updates the $SRID$, $STID^{new}$, $STID^{old}$, key_r in memory,otherwise it is not updated.

5) *TAG*

After the tag receives the message X^*_R,it will compare the received X^*_R with the X_R calculated by the tag,and if the two are equal,proving that the server is in agreement with its updated content,the tag updates the $STID$, key_t in memory,otherwise it does not update.

III. NON-FORMAL SECURITY ANALYSIS

A. TAG ANONYMITY

The anonymity of the tag is the basis for the RFID system to prevent identity tracking. In the protocol proposed in paper, the secret data of the tag are $STID$ and key_t. In the process of mutual authentication, both the two are encrypted before being transmitted. If an attacker wants to get $STID$, it must get the random number a generated by the reader, but the random number a is transmitted together with $STID$ in encrypted form, so that the attacker cannot acquire $STID$. If the attacker wants to get key_t, it needs to get the random number a generated by the reader and the $STID$, or get the random number b generated by the server and the $STID$. This is obviously impossible. Therefore, tag anonymity can be achieved in the protocol.

B. IMPERSONATION ATTACK

The attacker may initiate impersonation attack in three ways: attacker impersonates tag, reader, or server. In the first case of impersonated tag, information sent with tag every time contains random number generated by the reader, therefore the sent Message B and M are featured in timeliness and cannot be used to impersonate tag by replay. Another impersonation method is to impersonate information. But the impersonated Message B and M contain no correct $STID$ and key_t, so that the reader can identify the fake tag by simple calculation after receiving the fake information. This is how the protocol resists impersonation attack. In the second case of impersonated reader, if the attack impersonates reader by intercepting message and transmitting, since the message contains timestamp T_R, it cannot pass even the first step verification of the server, so that the attacker can not impersonate reader by replaying message. And if the attacker impersonates reader by making fake information, due to the lack of correct $SRID$ and key_r, the fake information sent to the server can be easily figured out. Therefore, it is impossible to impersonate reader by fake information. The protocol of this paper can resist attack of impersonated reader. And in the third case of impersonated server, the fake server must restore a^* in Message D, and find the correct privacy information from the database to calculate Message G, H, and I. But the attacker can neither restore the information, nor acquire the database information, therefore the protocol of this paper can resist

attack from fake server. In a word, the protocol proposed in this paper can resist impersonation attack.

C. REPLAY ATTACK

Replay attack refers to that the attacker replays the intercepted information and sends to one party of the communication, attempting to pass the verification and acquire privacy information. In this protocol, both the reader and the server generate one new random number respectively in each authentication period. The authentication information of each round would be operated by the new random number of current round. Even if the attacker intercepts data successfully, it can not pass the authentication of the reader and server by replay in the next round of authentication. Therefore, it can be deemed that the protocol in this paper can resist the replay attack.

D. DESYNCHRONIZED ATTACK

There are three types of desynchronized attacks: 1: the server updates, but the tag doesn't update;2. The tag updates, but the server doesn't update; 3. Desynchronization occurs in the tag sending channel, and the tag continuously starts two sessions within a short period of time. In the first case, an attacker intercepts the message X_R^* sent by the server at the updating stage, then the tag doesn't update for it receives no message.Since the server updates key_t,the keys at both sizes are different.However, the server stores C and O of previous round of authentication,even the tag uses the updated key_t,it can pass the authentication, resisting the desynchronization attack. In the second case, the attacker cannot obtain the privacy information of the tag and the random number generated by the server, and cannot forge messages to make the tag updated while the server does not update. In the third case, the parameters transmitted in the message $\{B, M\}$ are generated using random numbers and time series, which are different in each session. Suppose the attacker intercepts $\{B1, M1, t1\}\{B2, M2, T2\}$ in two consecutive sessions, since $B1$ and $B2$ are generated using different time series, $B1 \oplus M1! = B2 \oplus M2$, the tag cannot be traced, which satisfies the unlinkability requirement under desynchronization attack. In a word, this protocol can resist desynchronization attacks.

E. UNTRACEABILITY

Attackers obtain tag ids by intercepting status information to track tag traces and violate user privacy.To achieve traceability,the attacker must monitor successive session for a long time, thus finding out relevance among tag information and acquiring tag $STID$ to track. In this protocol, Message B and M are correlated with a, $STID$, key_t. The authentications of the three elements vary at each round, so that the three elements in two rounds of session are non-related. In this case, it is impossible for the attacker to achieve tracked attack. In this protocol, the pseudonym identifier of the tag is different from the tag identifier. The tag identity identifier is unique and unchanged, but the pseudonym identifier of the tag would be updated after each round of authentication, so that the tag can be hardly tracked or located. So the protocol can be considered as untraceable.

F. BRUTE FORCE ATTACK

To acquire privacy information, the attacker sometimes directly uses the method of exhaustion to figure out relative privacy information. In this protocol, the privacy information is encrypted by random numbers before being transmitted among three entities. Each piece of exchanged information is calculated by two or more unknown numbers, so that the attacker cannot acquire any useful privacy information by brute force according to the intercepted information. For example, the calculation of Message $F = h(Rot(SRID, key_r) \oplus a)$ uses three unknown numbers, so that it cannot exhaust its contained privacy information by brute force. Therefore, it can be deemed that the protocol can be used to resist brute force attack.

G. MUTUAL AUTHENTICATION

The protocol proposed in this paper concerns three communication entities. The channels for information transmission are not secure. Therefore Mutual authentication is required in the communication among the three. Authentication Message D, E, F, M, G, H, B contain unique identifiers of reader, tag, and server. Only legal entities can pass the verification of the other party. The server verifies the legitimacy of the reader through Message D, E, F, and verifies the legitimacy of the tag through Message M;The reader verifies the legitimacy of the server through Message G, H, and verifies the legitimacy of the tag through Message B; and the tag verifies the legitimacy of the server and the reader through Message I, J. In summary, the protocol in this paper realizes Mutual authentication.

H. PHYSICAL ATTACK AND CLONE ATTACK

In a physical attack, an attacker which has physical access to a tag can retrieve certain useful information stored in the tag. An attacker may then attempt to trace all previous communications of the flagged user. The information stored by tags in this protocol is updated during each round of authentication, and new random numbers are used to generate messages in each round of authentication, so all previous communications of tags cannot be tracked. Therefore, it can be considered that the protocol can resist physical attacks. Cloning attacks generally occur in RFID systems where a group of tags use the same key for identity authentication. In the scheme proposed in this paper, each tag has its own $\{ID, key\}$. Suppose that the ID of a tag is leaked, since each tag has different secrete parameters, the attacker cannot use the leaked tag information to clone other tags. Therefore, the RFID authentication protocol in this paper can resist clone attack.

In order to facilitate further analysis,we compared the security of this protocol with some proposed protocols and the results are shown in Tables 3, in which "$\surd$" means

TABLE 3. Security performance comparison.

Authentication protocols	Mei et al. [26]	Fan Kai et al. [31]	S.Chiou et al. [33]	Hwang et al. [36]	Gope P et al. [37]	Lee J et al. [38]	our protocol
Tag anonymity	×	√	√	√	√	√	√
Impersonation attack	√	√	√	√	√	√	√
Replay attack	×	√	√	√	√	√	√
Desynchronized attack	√	√	√	×	√	√	√
Tracked attack	√	√	×	√	√	√	√
Brute force attack	√	√	√	×	√	×	√
Mutual authentication	√	×	√	√	√	√	√
Physical and clone attack	×	√	√	√	√	√	√

the corresponding property is satisfied,while "×" means the corresponding property is not satisfied.

IV. FORMAL SECURITY ANALYSIS

A. BAN LOGIC

The BAN logic analysis method [29] is adopted to perform formal analysis and verification for the protocol proposed in this paper. BAN logic is modal logic based on belief. During the reasoning process of BAN logic, the belief of the entities participating in the protocol changes and develops with the information exchanges. When analyzing the protocol by BAN logic, the protocol message is firstly converted into formula of BAN logic, namely conducting the "idealization step" for the protocol. The second step is to perform rational assumption according to the specific situation. Finally it should perform reasoning according to the reasoning rules of the logic to judge if the protocol can achieve the anticipated objective or not. As a formal analysis method, BAN logic has been widely used in authentication protocols, which is featured in visual, simple, and efficient characteristics [30].

1) BASIC NOTATION OF BAN LOGIC

A, B :Represent the subject of communication.
k_a, k_b :Represent the shared key of the communication subject.
k_a^{-1}, k_b^{-1} :Represent the secret key of the communication subject.
N_a, N_b :Represent the viewpoint of the subject of the communication.
P, Q :Represent the subject of communication in a general sense, a concept of scope.
X, Y :Representing statements in a general sense.
K :Represent the encryption key in a general sense, a range of concepts.
(X, Y) :Represent the connection of X and Y.
$P \lhd X$:It indicates that P has seen X, P has received a message containing X, and P can read and repeat X.
$P \mid\sim X$:It means that P has said X and that P has sent a message containing X at some point in time. This assertion contains two meanings: on the one hand, it means that the message X was sent by P, on the other hand, it means that P can confirm the meaning of the message X,it can recognize the message and interpret it correctly.
$P \mid\Rightarrow X$:It indicates that P has control, or jurisdiction, over X.
$\#(X)$:It represents that X is fresh, meaning that it has not been transmitted before the protocol is executed.
$P \overset{k}{\leftrightarrow} Q$:It indicates that P and Q can communicate using a shared key K and that K is a good key. This assertion implies the exclusivity of the key, that is, only P,Q or a trusted third party knows that K.
$\overset{k}{\rightarrow} P$:Represent that K is the public key of P.
$P \overset{x}{\Leftrightarrow} Q$:It represents that X is a shared secret between P and Q and that X is unknown to any subject other than P and Q and the subjects they believe in.
$\{X\}_k$:Represent the result of encrypting X with key k.
$\langle X \rangle_Y$:It represents the combination of X and Y. In practice, it represents a simple cascade of X and Y.

2) REASONING RULES OF BAN LOGIC

There are 21 inference rules in BAN logic, and this paper only lists a few inference rules used in the proof process of this protocol.

$$R_1 : \frac{P \mid\equiv P \overset{k}{\leftrightarrow} Q, P \lhd \{X\}_k}{P \mid\equiv Q \mid\sim X}$$

This rule is a message implication rule, representing that P believes that Q has sent message X if P believes that k is a shared key between P and Q and P receives a message $\{X\}_k$ encrypted with K encrypting X.

$$R_2 : \frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$$

This rule is a temporary value check rule, indicating that P believes X if P believes that X is fresh and P believes that Q has sent X before.

$$R_3 : \frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}$$
$$R_4 : \frac{P \mid\equiv \#(X)}{P \mid\equiv \#(\alpha^X)}$$

These two are freshness rules, representing that if P believes that X is fresh, then P believes that the overall information containing X is also fresh.

$$R_5 : \frac{P \mid\equiv Q \mid\Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$$

This rule is a jurisdictional rule and represents that P believes X when P believes that Q has the right to control X and P believes that Q also believes X.

$$R_6 : \frac{P \mid\equiv \#(k), P \mid\equiv Q \mid\equiv X}{P \mid\equiv P \overset{k}{\leftrightarrow} Q}$$

This rule is the session key rule, where X is a necessary element for computing the key k. If P believes the freshness of k and P believes that Q believes X, then it is possible to determine that P believes that the key between P and Q is k.

$$R_7 : \frac{P \mid\equiv P \overset{Y}{\Leftrightarrow} Q, P \lhd \{X\}_Y}{P \mid\equiv Q \mid\sim X}$$

This rule is a message meaning rule. It means that Y is a shared secret of P and Q. When P receives a message X encrypted with Y, P can determine that Q must have sent X.

$$R_8 : \frac{P \lhd (X, Y)}{P \lhd X}$$

This rule is a receive message rule, which represents that when a subject P receives a formula and that subject knows the associated key, then that subject has received a component of that formula.

3) PROTOCOL ABSTRACTION DESCRIPTION

This subsection describes the authentication process between thesis protocol entities using some formal expressions, where T stands for tag, R stands for reader, and S stands for server.

$R \rightarrow T : Query, A$
$T \rightarrow R : B, M$
$R \rightarrow S : M, D, E, F, T_R$
$S \rightarrow R : G, H, I, T_S$
$R \rightarrow T : I, J, T_S$
$T \rightarrow R : X\_L$
$R \rightarrow S : X\_L, Y\_L$
$S \rightarrow R : X^*\_R, Y^*\_R$
$R \rightarrow T : X^*\_R$

4) PROTOCOL INITIALIZATION ASSUMPTIONS

$P_1 : R \mid\equiv S \mid\equiv R \overset{k_r}{\leftrightarrow} S$
$P_2 : T \mid\equiv S \mid\equiv T \overset{k_t}{\leftrightarrow} S$
$P_3 : R \mid\equiv T \mid\equiv R \overset{STID}{\leftrightarrow} T$
$P_4 : R \mid\equiv T \mid\equiv S \mid\equiv \#(a)$
$P_5 : S \mid\equiv R \mid\equiv T \mid\equiv \#(b)$
$P_6 : T \mid\equiv R \mid\Rightarrow A$
$P_7 : S \mid\equiv R \mid\Rightarrow D$
$P_8 : S \mid\equiv R \mid\Rightarrow E$
$P_9 : S \mid\equiv R \mid\Rightarrow F$
$P_{10} : R \mid\equiv T \mid\Rightarrow B$
$P_{11} : S \mid\equiv T \mid\Rightarrow M$
$P_{12} : R \mid\equiv S \mid\Rightarrow G$
$P_{13} : R \mid\equiv S \mid\Rightarrow H$
$P_{14} : T \mid\equiv S \mid\Rightarrow I$
$P_{15} : T \mid\equiv R \mid\Rightarrow J$

5) PROTOCOL PROOF GOALS

$G_1 : T \mid\equiv A$
$G_2 : T \mid\equiv I$
$G_3 : T \mid\equiv J$
$G_4 : R \mid\equiv B$
$G_5 : R \mid\equiv G$
$G_6 : R \mid\equiv H$
$G_7 : S \mid\equiv M$
$G_8 : S \mid\equiv D$
$G_9 : S \mid\equiv E$
$G_{10} : S \mid\equiv F$
$G_{11} : R \mid\equiv R \overset{key_r^{new}}{\leftrightarrow} S$
$G_{12} : T \mid\equiv T \overset{key_t^{new}}{\leftrightarrow} S$
$G_{13} : S \mid\equiv S \overset{key_r^{new}}{\leftrightarrow} R$
$G_{14} : S \mid\equiv S \overset{key_t^{new}}{\leftrightarrow} T$

6) SPECIFIC PROCESS TO PROVE THE PROTOCOL

The next part shows all details of the formal proof of the protocol. It can be obtained from the protocol abstraction process (1) that:

$$T \lhd \{\{A\}_{STID}, Query\} \tag{1}$$

According to the assumption P_3,$STID$ is the unique key between reader and server.There's no other entity knowing the $STID$ except the reader and the tag. Combing with rule R_8, it can be obtained:

$$T \lhd \{\{A\}_{STID}\} \tag{2}$$

It can be obtained by equation (2) combing with Suppose P_3 and Rule R_1:

$$T \mid\equiv R \mid\sim \{A\}_{STID} \tag{3}$$

Message $A = Rot(a, STID)$ indicates that A is a whole containing random number a.Combining the assumption P_4 and rule R_4, it can be obtained:

$$T \mid\equiv \#\{A\}_{STID} \tag{4}$$

It can be obtained by formula (3)(4) combining with rule R_2:

$$T \mid\equiv R \mid\equiv \{A\}_{STID} \tag{5}$$

It can be obtained by formula (5), assumption P_6 and rule R_5:

$$T \mid\equiv \{A\} \tag{6}$$

Till now, proof for goal G_1 is over.

Similarly, it can be obtained by protocol abstraction process (5) that:

$$T \lhd \{\{I\}_{key_t}, \{J\}_{STID}, T_S\} \tag{7}$$

According to the assumption P_2, it can be obtained that key_t is the unique key between tag and server. Combining with rule R_8, it can be obtained that:

$$T \lhd \{\{I\}_{key_t}\} \tag{8}$$

Equation (9) combined with assumption P_2 and rule R_1 can be obtained:

$$T \mid\equiv S \mid\sim \{I\}_{key_t} \tag{9}$$

Message I is a whole containing random number b,According to assumption P_5 and rule R_4, it can be obtained that:

$$T \mid\equiv \#\{I\}_{key_t} \tag{10}$$

It can be obtained according to equation (10)(11) and rule:

$$T \mid\equiv R \mid\equiv \{I\}_{key_t} \tag{11}$$

It can be obtained by formula (12), assumption P_{14} and rule R_5:

$$T \mid\equiv \{I\} \tag{12}$$

Till now, proof for goal G_2 is over. Similarly, $G_3 - G_{11}$ can also be proved.

It can be obtained according to assumption P_5 and rule R_4:

$$R \mid\equiv \#(key_r^{new}) \tag{13}$$

It can be obtained by combining with the formula (14) and the assumption P_5, as well as the rule R_6:

$$R \mid\equiv R \overset{key_r^{new}}{\leftrightarrow} S \tag{14}$$

Till now, proof for goal G_{12} is over. Similarly,$G_{13} - G_{15}$ can also be proved. In a word, all security objectives of the protocol can be performed with formal proof, which indicates that the protocol proposed in this paper satisfies the logic security requirement.

B. PROVERIF

In this section, the proverif is used for security analysis. Proverif modeling is performed according to the authentication processes for tag, reader and cloud server. And then an identity verification protocol model simulation is built up. The overall process is as follows:

(1)Define the public channel pch and secure channel sch for identity authentication, and define the variables applied in the protocol. They are global variables, but [private] limits and makes them unable to be directly obtained by attacker; next, define string join operation, XOR operation, modular operation, hash function and other functions and equations. A series of related queries are compiled to validate the security requirements. The detailed functional definitions are shown in the Fig. 5 and Fig. 6.

(2)The specific process of the tag is as shown in the Fig. 7.

(3)The specific process of the reader is as shown in the Fig. 8.

(4)The specific process of cloud server is as shown in the Fig. 9.

(5)The Proverif verification results are as shown in the Fig. 10 and Fig. 11. It can be concluded that, *STID*, *SRID*, *T_key*, *R_key* can resist the attacks from attackers, and the proposed protocol passes the proverif verification.

```
(*channel*)
free pch:channel.    (*public channel*)
free sch:channel.    (*secure channel*)

(*constants*)
free STID,T_key,T_s:bitstring[private].
free SRID,R_key,Qr,T_r:bitstring[private].
free a,b,t,T_t:bitstring[private].

(*functions & reductions & equations*)
fun h (bitstring): bitstring [data].    (*hash*)
fun mult(bitstring,bitstring):bitstring.    (*scalar multiplication*)
fun mod(bitstring,bitstring):bitstring.    (*modulus operation*)
fun con(bitstring,bitstring):bitstring.    (*concatenation operation*)
reduc forall m:bitstring,n:bitstring;getmess(con(m,n))=m.
fun add(bitstring,bitstring):bitstring.
fun sub(bitstring,bitstring):bitstring.
fun rot(bitstring,bitstring):bitstring.
fun srot(bitstring,bitstring):bitstring.
fun eac(bitstring,bitstring):bitstring.
fun w(bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.    (*XOR*)
equation forall x:bitstring,y:bitstring;xor(xor(x,y),y)=x.
```

FIGURE 5. Function1.

```
(*queries*)
query attacker(STID).
query attacker(SRID).
query attacker(T_key).
query attacker(R_key).


(*event*)
event beginT(bitstring).
event endT(bitstring).
event beginR(bitstring).
event endR(bitstring).
event beginCS(bitstring).
event endCS(bitstring).

query x:bitstring;inj-event(beginR(x))==>inj-event(beginT(x)).
query x:bitstring;inj-event(beginCS(x))==>inj-event(beginR(x)).
query x:bitstring;inj-event(endR(x))==>inj-event(endCS(x)).
query x:bitstring;inj-event(endT(x))==>inj-event(endR(x)).
```

FIGURE 6. Function2.

```
(*Tag*)
let TAG(T_key:bitstring,STID:bitstring,T_t:bitstring)=
    in(pch,(m1:bitstring,m2:bitstring));
    let a'=srot(m1,w(STID)) in
    let B=rot(STID,xor(a',T_t)) in
    let C=eac(xor(T_key,STID),T_key) in
    let M=xor(a',C) in
    event beginT(M);
    out(pch,(B,M));
    in(pch,(m3:bitstring,m4:bitstring,m5:bitstring));
    let b'=xor(m3,eac(rot(T_key,STID),xor(T_key,STID))) in
    let m4'=rot(xor(b',a'),xor(b',STID)) in
    if m4'=m4 then
        event endT(m4').
```

FIGURE 7. Tag.

C. RANDOM ORACLE MODEL

In this section, the security of our proposed protocol is formally evaluated by the random oracle model proposed in [34] and [35]. A random prediction is a mathematical function that responds to each query by uniformly selecting random

```
(*Reader*)
let READER(R_STID:bitstring,R_key:bitstring,SRID:bitstring,a:bitstring,Qr:bitstring,T_r:bitstring)=
    let A=rot(a,R_STID) in
    out(pch,(Qr,A));
    in(pch,(R1:bitstring,R2:bitstring,R7:bitstring)) ;
    let RSTID=srot(R1,w(xor(a,R7))) in
    if RSTID=R_STID then
        event beginR(RSTID);
        let D=xor(a,h(T_r)) in
        let E=rot(h(SRID),xor(a,T_r)) in
        let F=h(xor(rot(SRID,T_key),a)) in
        out(pch,(D,E,F,T_r,R2));
        in(pch,(R3:bitstring,R4:bitstring,R5:bitstring,R6:bitstring));
        let d=xor(h(xor(h(R_STID),a)),R3) in
        let suc=h(xor(con(a,d),rot(SRID,T_r))) in
        if suc=R4 then
            let J=rot(xor(d,a),xor(d,R_STID)) in
            out(pch,(J,R5,R6));
            event endR(J).
```

FIGURE 8. Reader.

```
(*CS*)
let CLOUDSERVER(C_STID:bitstring,C_SRID:bitstring,b:bitstring,T_s:bitstring,CR_key:bitstring,CT_key:bit
    in(pch,(C1:bitstring,C2:bitstring,C3:bitstring,C4:bitstring,C5:bitstring));
    (*let v=T_s-C5  in*)
    (*if (T_s-C5)>=t  then *)
        let a'=xor(C2,h(C5)) in
        let M=srot(C3,w(xor(a',C5))) in
        let F'=h(xor(rot(C_SRID,CR_key),a')) in
        if F'=C4 then
            let C'=xor(C1,a') in
            let H=eac(xor(CT_key,C_STID),CT_key) in
            if C'=H then
                event beginCS(H);
                let G=xor(h(xor(h(C_STID),a)),b) in
                let  K=h(xor(con(a',b),rot(C_SRID,CR_key))) in
                let  I=xor(eac(rot(C_STID,CT_key),xor(C_STID,CT_key)),b) in
                out(pch,(G,K,I,T_s));
                event endCS(G).
```

FIGURE 9. Server.

```
-- Query not attacker(STID[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(STID[])
RESULT not attacker(STID[]) is true.
-- Query not attacker(SRID[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(SRID[])
RESULT not attacker(SRID[]) is true.
-- Query not attacker(T_key[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(T_key[])
RESULT not attacker(T_key[]) is true.
-- Query not attacker(R_key[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(R_key[])
RESULT not attacker(R_key[]) is true.
-- Query inj-event(beginR(x)) ==> inj-event(beginT(x)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(beginR(x)) ==> inj-event(beginT(x))
RESULT inj-event(beginR(x)) ==> inj-event(beginT(x)) is true.
-- Query inj-event(beginCS(x)) ==> inj-event(beginR(x)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(beginCS(x)) ==> inj-event(beginR(x))
RESULT inj-event(beginCS(x)) ==> inj-event(beginR(x)) is true.
-- Query inj-event(endR(x)) ==> inj-event(endCS(x)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(endR(x)) ==> inj-event(endCS(x))
RESULT inj-event(endR(x)) ==> inj-event(endCS(x)) is true.
-- Query inj-event(endT(x)) ==> inj-event(endR(x)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(endT(x)) ==> inj-event(endR(x))
RESULT inj-event(endT(x)) ==> inj-event(endR(x)) is true.
```

FIGURE 10. Result1.

responses from a random domain. For the same input, the oracle machine will have the same output every time.

Reveal 1: A one-way hash function with anti-collision properties behaves as a random oracle that passes input x from its corresponding digest $y = h(x)$.

```
-- Query not attacker(STID[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(STID[])
RESULT not attacker(STID[]) is true.
-- Query not attacker(SRID[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(SRID[])
RESULT not attacker(SRID[]) is true.
-- Query not attacker(T_key[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(T_key[])
RESULT not attacker(T_key[]) is true.
-- Query not attacker(R_key[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(R_key[])
RESULT not attacker(R_key[]) is true.
-- Query inj-event(beginR(x)) ==> inj-event(beginT(x)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(beginR(x)) ==> inj-event(beginT(x))
RESULT inj-event(beginR(x)) ==> inj-event(beginT(x)) is true.
-- Query inj-event(beginCS(x)) ==> inj-event(beginR(x)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(beginCS(x)) ==> inj-event(beginR(x))
RESULT inj-event(beginCS(x)) ==> inj-event(beginR(x)) is true.
-- Query inj-event(endR(x)) ==> inj-event(endCS(x)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(endR(x)) ==> inj-event(endCS(x))
RESULT inj-event(endR(x)) ==> inj-event(endCS(x)) is true.
-- Query inj-event(endT(x)) ==> inj-event(endR(x)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(endT(x)) ==> inj-event(endR(x))
RESULT inj-event(endT(x)) ==> inj-event(endR(x)) is true.
```

FIGURE 11. Result2.

Reveal 2: hamming weight is a random oracle model, which can deliver n under the condition of providing $W(n)$.

Proposition 1: Assuming that the one-way hash function and Hamming weight behaviors are almost similar to random oracle, it is proved that the proposed scheme is secure and hard for attackers to launch attacks of extracting reader identity, key and generating random parameters.

Proof: the goal is to construct an attacker for the proposed protocol. The attacker shall be able to extract the reader's ID, key, and secret random number using Reveal oracle 1 and Reveal oracle 2 described in Algorithm 1. The success probability of experiment $EXP1_{A,HRFID}^{Hash,hamming}$ is success1 $= |\Pr[EXP1_{A,HRFID}^{Hash,hamming} = 1]$-1|, and the meaning of $|\Pr[EXP1_{A,HRFID}^{Hash,hamming} = 1]|$ is the probability of experiment results equaling 1. The dominance function of this experiment is Adv1(t1,Q_{r1},Q_{r2})=max(success1), which represents the number of Reveal 1 and Reveal 2 displayed by querying all attackers of Q_{r1} and Q_{r2} within polynomial execution time t1. If and only if Adv1(t1,Q_{r1},Q_{r2}) $\leq \varepsilon$(ε is a sufficiently small value greater than 0), the protocol of this study is certified to be secure and hard for attacker A to illegally acquire privacy data.

Assuming that A can solve the described hamming and invert the one-way hash function, then the above condition does not hold and the attacker can obtain the key and identity ID of the tag and win. However, according to the performance of hash function and the method of calculating hamming weight, it is impossible to export the input x of hash function and obtain 128-bit key by the hamming weight within limited polynomial time, therefore, Adv1(t1,Q_{r1},Q_{r2}) $\leq \varepsilon(\varepsilon \geq 0)$, proving that the protocol proposed in this paper is secure

Algorithm 1 $EXP1_{A,HRFID}^{Hash,hamming}$

Eavesdrop the authentication message$\{M, D, E, F, T_R\}$
Call reveal oracle 1.Let(a') $\Leftarrow reveal1(D)$
Eavesdrop the authentication message$\{Query, A\}$
Call reveal oracle2.Let(STID) $\Leftarrow reveal2(A)$
Calculate h(STID)=$E \gg W(a' \oplus T_R)$
Call reveal oracle 1.Let(SRID') $\Leftarrow reveal1(h(STID))$
Call reveal oracle 1.Let($Rot(SRID, key_r) \oplus a'$) $\Leftarrow reveal1(E)$
Call reveal oracle 2.Let(key_r') $\Leftarrow reveal1(Rot(SRID, key_r))$
Eavesdrop the authentication message$\{G, H, I, T_s\}$
Calculate b'=$G \oplus h(a' \oplus h(STID))$
Calculate H'=$h((a'||b') \oplus Rot(SRID, key_r))$
if $H' = H$ **then**
    Accept key_r as the secret key of the Reader
    Accept a as the secret parameter of the Reader
    Accept SRID as the identity ID of the Reader
    Return 1(success)
else
    Return 0(Failure)
end if

Algorithm 2 $EXP2_{A,HRFID}^{Hash,hamming}$

Eavesdrop the authentication message$\{B, M, t\}$
According to Algorithm 1, calculate STID'=B $\gg W(t \oplus a')$
Eavesdrop the authentication message$\{G, H, I, T_s\}$
Calculate b'=$G \oplus h(a' \oplus h(STID))$
Calculate C'=M$\oplus a'$
Call reveal oracle 2 in C'.Let(key_t'') $\Leftarrow reveal2(C')$
Calculate I'=b'$\oplus$ Eac(Rot(key_t', $STID'$),$key_t'\oplus$ STID')
if $I' = I$ **then**
    Accept key_t as the secret key of the Tag
    Accept STID as the identity ID of the Tag
    Return 1(success)
else
    Return 0(Failure)
end if

when facing any attacker who tries to extract secrete parameters.

Proposition 2: Assuming that the one-way hash function and Hamming weight behavior are random oracle, then it is proved that the proposed scheme is secure and hard for attacker to extract tag identity and key.

Proof: The Proof of Proposition 1 is similar to that of the Proposition 1: it is assumed that the attacker can use Reveal oracle 1 and Reveal oracle 2 which are described in Algorithm 2 to extract the identity and key of the tag. The same as the previous experiment, the success probability of $EXP2_{A,HRFID}^{Hash,hamming}$ is success2 = |Pr[$EXP2_{A,HRFID}^{Hash,hamming} = 1$]-1|. And |Pr[$EXP2_{A,HRFID}^{Hash,hamming} = 1$]-1| means the probability of experimental result equals 1. The dominance function of this experiment is Adv2(t2,Q_{r1},Q_{r2})= max(success2), which represents the number of Reveal 1 and Reveal 2 displayed by querying all attackers of Q_{r1} and Q_{r2} during the polynomial execution time t2. The protocol of this paper is deemed as secure, and hard for attacker A to illegally acquire privacy data. If and only if Adv2(t2,Q_{r1},Q_{r2})$\leq \varepsilon$ (ε is a sufficiently small value greater than 0).

Assuming that A can solve the described hamming and invert the one-way hash function, then the above dominant function inequality condition does not hold and the attacker can obtain the key and identity ID of the tag and win. However, according to the performance of hash function and the method of calculating hamming weight, it is impossible to obtain 128-bit key by hash function input and hamming weight within limited polynomial time, therefore, Adv2(t2,Q_{r1},Q_{r2})$\leq \varepsilon(\varepsilon \geq 0)$,proving that the protocol proposed in this paper is secure when facing any attacker who tries to extract secrete parameters.

TABLE 4. Symbols used in performance analysis.

Symbol	Meaning
l_h	Length of the output of hash function (e.g.128 bits)
l_r	Length of random numbers(e.g.128 bits)
l_c	Length of the output of CRC and PRNG (e.g.128 bits)
l_q	Length of message "Query" (e.g.40 bits)
l_{id}	Length of ID and SID (e.g.128 bits)
l_t	Length of time (e.g.64 bits)
l_m	Length of the modulo squaring(e.g.1024 bits)
l_k	Length ofK_t,K_r,k,ACK, n, s_i, t_i, k_i(e.g.1024 bits)

V. PERFORMANCE ANALYSIS

In this section, the protocol of this paper is compared with other similar protocols in terms of performance, including the comparison of communication cost, protocol computation cost and tag cost. The specific comparison results are described as follows.

A. COMPARISON OF COMMUNICATION COST

Tables 4 lists the basic symbols used in the comparison of communication costs, and the numbers in parentheses indicate the number of bits of communication data.

Tables 5 show the specific comparison results of communication cost between the proposed protocol and other similar protocols. Protocol communication cost includes interaction times and communication data length. The protocol proposed in this paper has a medium number of interactions in the authentication phase and a low total length of communication data. The protocol proposed in literature [37] only implements two-party authentication for RFID systems, and the default server and reader are integrated and not applicable to mobile RFID systems, so its communication data length is lower than that of the protocol in this paper. Although the protocol in literature [31] has a slightly lower communication cost than the protocol in this paper, it does not implement two-party authentication between the cloud server, reader,

TABLE 5. Comparison of communication cost.

Authentication protocol	Round	Server	Reader	Tag	Total
Mei et al. [26]	5	$4l_r$	$l_q + l_{id} + 4l_k + 4l_r$	$3l_k + l_r$	8488 bits
Fan Kai et al. [31]	9	$l_t + l_m + 2l_c$	$3l_t + 4l_{id} + 5l_c + l_q$	$2l_c + l_{id}$	3112 bits
S.Chiou et al. [33]	5	$2l_c$	$3l_c + 2l_m + 2l_t$	$l_c + l_m$	3968 bits
Hwang et al. [36]	4	$l_t + 2l_h + 2l_k$	$l_k + l_h + l_{id}$	$2l_h + l_t + l_k$	3968 bits
Gope P et al. [37]	4	None	$l_k + l_r + 3l_h$	$l_h + l_r + l_k + l_{id}$	2944 bits
Lee J et al. [38]	4	$2l_h + l_t + l_k$	$l_h + l_k + l_{id}$	$3l_h + l_t + l_k + l_{id}$	4224 bits
our protocol	5	$2l_h + l_k + l_t$	$l_q + 2l_r + 3l_h + l_t$	$l_{id} + l_k$	3240 bits

TABLE 6. Comparison of computation costs of various protocols.

Authentication protocol	Cloud Server	Reader	Tag	Total	Time(ms)
Mei et al. [26]	Random	Random	Random	3Random	0.375
Fan Kai et al. [31]	4PRNG	8PRNG	5PRNG	17PRNG	1.411
S.Chiou et al. [33]	4msg+4PRNG	2msg+2PRNG	2msg+2PRNG	8msg+8PRNG	1.032
Hwang et al. [36]	7hash	2hash	5hash	14hash	0.91
Gope P et al. [37]	None	Random+5hash	Random+5hash+2PUF	2Random+10hash+2PUF	1.35
Lee J et al. [38]	5hash	2hash	5hash	12hash	0.78
our protocol	Random+5hash	Random+5hash	Bitwise operation,ignored	2Random+10hash	0.9

TABLE 7. Comparison of computation costs of various protocols.

Authentication protocol	computing cost	storage cost
Mei et al. [26]	$Random + 4Bro$	$2l_k + l_{id} = 2176bits$
Fan Kai et al. [31]	$2PRNG + Rot$	$l_k + 2l_{id} = 1280bits$
S.Chiou et al. [33]	$Modulosquaring + 2PRNG$	$2l_k + 2l_{id} = 2304bits$
Hwang et al. [36]	$14hash$	$l_k + 2l_{id} = 1208bits$
Gope P et al. [37]	$2Random + 10hash + 2PUF$	$2l_h + l_{id} = 384bits$
Lee J et al. [38]	$12hash$	$l_k + 3l_{id} = 1408bits$
our protocol	$2Eac + 2Rot$	$l_k + l_{id} = 1152bits$

and tag, and there is a security vulnerability of impersonating a reader. In conclusion, the protocol proposed in this paper actually shows lower communication cost than other similar protocols under the premise of ensuring the security of mobile RFID system.

B. COMPARISON OF COMPUTATION COST AMONG PROTOCOLS

In this section, the computation cost and execution time of RFID tags, RFID readers, and cloud servers are defined, and the difference between the proposed protocol and other similar protocols are showed as well. The computations of this protocol and other protocols are done by hash, XOR, random number, and modular operation, etc. Among all these operations, the "XOR" operation, "and" operation, and "ring shift left" are all bitwise operation, which is actually a type of lightweight computation having little impact on overall computation. Therefore the computation of bitwise operation can be ignored, while focusing more on the dominating operations featured in dense computations in the protocol. In this paper, Random is used to represent the computations of random number, PRNG is used as the computation to create pseudo-random numbers, Hash is used to represent the computation of Hash functions, Msg is used to represent the computation of modulo squared, Bro is used to represent the computation for bit substitution, Cro is applied to represent the computation of bit crossing, Rot is used to represent the computations for ring shift left, puf represents the computation of the physical incompressible function, and eac represents the computation of the exchange and re-crossing. The computation time of a single random, such as random, PRNG, Hash, msg and puf are 0.125, 0.083, 0.065, 0.046 and 0.226, respectively.

It can be seen from the Table 6 that, The protocol proposed in literature [26] shows the lowest total computation cost and the shortest execution time, but this protocol uses a complex hash function on the tag side, has high computation cost on the tag side, the tag id is transmitted in clear text and not updated, cannot guarantee security, and the total length of the communication data is the longest, so even though this protocol has the lowest computation cost and execution time, it is not the best choice. The protocol proposed in literature [38] has a shorter execution time than the protocol in this paper, but its total length of communication data is longer than the protocol in this paper and it is not resistant to brute force attacks. In conclusion, the computational cost and execution time of the protocol proposed in this study are kept low while ensuring security and low communication data length.

C. COMPARISON ON TAG COST

Most complex computation of RFID authentication protocol is carried out on the server. The tag is the most restrained entity with the weakest computing power in the system,

which makes its computation and storage an important concern. Table 7 shows the comparison of the computational and storage costs of the proposed protocol with other protocols on the tag. It can be seen from the table that, the protocol proposed in literature [26] uses random number operation on the tag, the one proposed in literature [31] adopts pseudo-random number operation on the tag, the one proposed in literature [32] uses modular square operation on the tag, and the protocol proposed in literature [35], [36], [37] applies hash function on the tag. The cost of these operations is higher than that of the bit operation used in the protocol of this study. The simple bit operation on the tag consumes less computing cost, which meets the requirements of low-cost tags in the RFID system. Meanwhile, for the protocol proposed in this paper, only STID and are stored on the tag, consuming lower storage cost. Speaking of this storage cost, though it is higher than that required by protocol proposed in literature [36], the protocol of literature [36] is two-party authentication, while the protocol proposed in this paper is three-party authentication, so the protocol in this paper is more competitive than other protocols in terms of tag storage cost among the three-party RFID authentication protocols.

VI. CONCLUSION

An efficient mobile RFID authentication protocol is proposed in this paper. It can be applied in a low-cost RFID system to provide a secure environment for the secure storage and communication of private data in the system, and resist various known attacks. For this protocol, the Hash Function is used at the high-performance reader end to calculate authentication information, and the exchange-cross bitwise operation is used at the performance-restricted tag end to calculate the authentication information. The Hash Function helps improve the security of the authentication information while the exchange-cross bitwise operation guarantees low computation cost at tag end and the tag anonymity. The cloud server stores the encrypted information in form of index data table, which enhances the cloud server's retrieval efficiency during its authentication to the tag and the reader, and reduces the risk of sensitive information disclosure of the cloud server. By doing so, the safe and efficient identity authentication among tag, reader, and server is perfectly achieved. According to the non-formal security analysis, the efficient mobile RFID authentication protocol designed in this paper is featured in enhanced security function and capability in resisting known attacks like impersonation attack, replay attack, and tracked attack, etc. In this paper, the protocol security is further proved by BAN logic formal analysis, proverif tool, and random oracle model, while the low computing cost of the protocol and the low storage cost of the tag-end are also proved by the performance analysis. In a word, this is a safe, efficient, and low-cost RFID mobile authentication protocol applicable to the target tracking system.

The lightweight authentication protocol currently uses security analysis to prove the security, and the subsequent research work is to establish a security model to prove the security of the authentication protocol under the standard model. The protocol proposed in this paper does not support the integration with physical identification systems (e.g., fingerprints) for the time being, and the next research direction is to gradually adjust the protocol to achieve the integration with physical identification systems in practical applications.

REFERENCES

[1] S. Anandhi, R. Anitha, and V. Sureshkumar, "IoT enabled RFID authentication and secure object tracking system for smart logistics," *Wireless Pers. Commun.*, vol. 104, pp. 543–560, Oct. 2018, doi: 10.1007/s11277-018-6033-6.

[2] C.-C. Lee, C.-T. Li, C.-L. Cheng, Y.-M. Lai, and A. V. Vasilakos, "A novel group ownership delegate protocol for RFID systems," *Inf. Syst. Frontiers*, vol. 21, no. 5, pp. 1153–1166, Oct. 2019, doi: 10.1007/s10796-018-9835-x.

[3] Y. Zhong. *Research on Key Technologies of RFID in Intelligent Logistics System*. Shanghai, China: Fudan University, 2014.

[4] T. Fan, F. Tao, S. Deng, and S. Li, "Impact of RFID technology on supply chain decisions with inventory inaccuracies," *Int. J. Prod. Econ.*, vol. 159, pp. 117–125, Jan. 2015, doi: 10.1016/j.ijpe.2014.10.004.

[5] Z. Sun, Z. Ren, and H. Yan, "Modern tracking technology of logistics information research progress review," *J. Zhejiang Univ. Sci. Technol.*, vol. 17, no. 2, pp. 126–130, 2005, doi: 10.3969/j.issn.1671-8798.2005.02.012.

[6] W. C. Wang, Y. Yona, S. N. Diggavi, and P. Gupta, "Design and analysis of stability-guaranteed PUFs," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 978–992, Apr. 2018, doi: 10.1109/TIFS.2017.2774761.

[7] A. Mitrokotsa, M. R. Rieback, and A. S. Tanenbaum, "Classifying RFID attacks and defenses," *Inf. Syst. Frontiers*, vol. 12, no. 5, pp. 491–505, 2010, doi: 10.1007/s10796-009-9210-z.

[8] D. Liu, J. Ling, and X. Yang, "An improved RFID authentication protocol to meet the backward privacy," *Comput. Sci.*, vol. 43, no. 8, pp. 128–130, 2016, doi: 10.11896/j.issn.1002-137X.2016.8.027.

[9] EPCglobal, "EPC radio-frequency identity protocols generation-2 UHF RFID. Specification for RFID air interface protocol for communications at 860 MHZ-960 MHZ," Milan, Italy, EPCglobal, Tech. Rep., 2013.

[10] M. Shariq, K. Singh, and P. K. Maurya, "URASP: An ultralightweight RFID authentication scheme using permutation operation," *Peer-to-Peer Netw. Appl.*, vol. 44, pp. 1–21, Jul. 2021, doi: 10.1007/s12083-021-01192-5.

[11] S. D. Kaul and A. K. Awasthi, "Privacy model for threshold RFID system based on PUF," *Wireless Pers. Commun.*, vol. 95, no. 3, pp. 2803–2828, 2017, doi: 10.1007/s11277-017-3965-1.

[12] Y. Tao, X. Zhou, Y. Ma, and Z. Fan, "Hash function-based mobile mutual authentication protocol," *J. Comput. Appl.*, vol. 36, no. 3, pp. 657–660, 2016, doi: 10.11772/j.issn.1001-9081.2016.03.657.

[13] K. Fan, W. Jiang, H. Li, and Y. Yang, "Lightweight RFID protocol for medical privacy protection in IoT," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1656–1665, 2018, doi: 10.1109/TII.2018.2794996.

[14] H. Xiao, A. Alshehri, and B. Christianson, "A cloud-based RFID authentication protocol with insecure communication channels," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Tianjin, Aug. 2016, pp. 332–339, doi: 10.1109/TrustCom.2016.0081.

[15] A. W. Phan, "Cryptanalysis of a new ultralightweight RFID authentication protocol—SASI," *IEEE Trans. Dependable Secure Comput.*, vol. 6, no. 4, pp. 316–320, Dec. 2009, doi: 10.1109/TDSC.2008.33.

[16] T. Cao, E. Bertino, and H. Lei, "Security analysis of the SASI protocol," *IEEE Trans. Dependable Secure Comput.*, vol. 6, no. 1, pp. 73–77, Mar. 2009, doi: 10.1109/TDSC.2008.32.

[17] H. M. Sun, W. C. Ting, and K. H. Wang, "On the security of Chien's ultralightweight RFID authentication protocol," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 2, pp. 315–317, 2011, doi: 10.1109/TDSC.2009.26.

[18] P. D'Arco and A. De Santis, "On ultralightweight RFID authentication protocols," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 4, pp. 548–563, Aug. 2011, doi: 10.1109/TDSC.2010.75.

[19] H. Y. Chien, "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity," *IEEE Trans. Dependable Secure Comput.*, vol. 4, no. 4, pp. 337–340, Dec. 2007, doi: 10.1109/TDSC.2007.70226.

[20] P. Peng, Y. M. Zhao, and W. L. Han, "Ultra-lightweight RFID mutual authentication protocol," *Comput. Eng.*, vol. 37, no. 16, pp. 140–142, 2011, doi: 10.3969/j.issn.1000-3428.2011.16.047.
[21] Y. Farzaneh, M. Azizi, and M. Dehkordi, "Vulnerability analysis of two ultra lightweight RFID authentication protocols," *Int. Arab J. Inf. Technol.*, vol. 12, no. 4, pp. 340–345, 2015.
[22] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E. Tapiador, and A. Ribagorda, "Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol," in *Proc. 9th Int. Workshop Inf. Secur. Appl.* Berlin, Germany: Springer, 2008, pp. 56–68.
[23] K. Fan, N. Ge, Y. Gong, H. Li, R. Su, and Y. Yang, "An ultra-lightweight RFID authentication scheme for mobile commerce," *Peer Peer Netw. Appl.*, vol. 10, no. 2, pp. 368–376, 2017, doi: 10.1007/s12083-016-0443-6.
[24] K. Huang, Y. Liu, and X. Yin, "Ultra-lightweight RFID mutual authentication protocol based on regeneration transformation," *J. Comput. Appl.*, vol. 39, no. 1, pp. 118–125, 2019, doi: 10.11772/j.issn.1001-9081.2018071738.
[25] Z. Ma and L. Cheng, "Mobile mutual authentication protocol based on word synthesis operation," *Appl. Res. Comput.*, vol. 33, no. 8, pp. 814–819, 2017, doi: 10.3969/j.issn.1001-3695.2017.08.047.
[26] S. Mei and R. Deng, "An ultra-lightweight mobile RFID authentication protocol with bit-substitution computing," *Comput. Eng. Appl.*, vol. 56, no. 3, pp. 100–105, 2020, doi: 10.3778/j.issn.1002-8331.1905-0234.
[27] B. Zhi and H. Yigang, "Recognition of the anticollision algorithm for RFID systems based on tag grouping," *Int. J. Inf. Comput. Technol.*, vol. 14, no. 1, pp. 81–88, 2019, doi: 10.1504/IJICT.2019.10017022.
[28] Y. Duan, "Ultra-lightweight authentication protocol based on EAC," *Comput. Appl. Softw.*, vol. 38, no. 9, pp. 333–337, 2021, doi: 10.3969/j.issn.1000-386x.2021.09.052.
[29] P. F. Syverson and P. C. van Oorschot, "On unifying some cryptographic protocol logics," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, 1994, pp. 14–28, doi: 10.1109/RISP.1994.296595.
[30] S. Yang, "Analytical study of security protocols and their BAN logic," Guizhou Univ., Guiyang, China, Tech. Rep., 2007, pp. 1–99.
[31] F. Kai, Z. Shanshan, Z. Kuan, and Y. Yang, "A lightweight authentication scheme for cloud-based RFID healthcare systems," *IEEE Netw.*, vol. 33, no. 2, pp. 44–49, Apr. 2019.
[32] P. K. Roy and A. Bhattacharya, "Desynchronization resistant privacy preserving user authentication protocol for location based services," *Peer-to-Peer Netw. Appl.*, vol. 14, pp. 1–15, Jun. 2021, doi: 10.1007/s12083-021-01194-3.
[33] S. Y. Chiou and S. Y. Chang, "An enhanced authentication scheme in mobile RFID system," *Ad Hoc Netw.*, vol. 71, pp. 1–13, Mar. 2018, doi: 10.1016/j.adhoc.2017.12.004.
[34] S. Izza, M. Benssalah, and K. Drouiche, "An enhanced scalable and secure RFID authentication protocol for WBAN within an IoT environment," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102705, doi: 10.1016/j.jisa.2020.102705.
[35] R. Ali and A. Pal, "Cryptanalysis and biometric-based enhancement of a remote user authentication scheme for E-healthcare system," *Arabian J. Sci. Eng.*, vol. 43, pp. 7837–7852, Apr. 2018, doi: 10.1007/s13369-018-3220-4.
[36] P. Gope and T. Hwang, "A realistic lightweight authentication protocol preserving strong anonymity for securing RFID system," *Comput. Secur.*, vol. 55, pp. 271–280, Nov. 2015, doi: 10.1016/j.cose.2015.05.004.
[37] P. Gope, R. Amin, S. K. H. Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment," *Future Gener. Comput. Syst.*, vol. 83 pp. 629–637, Jun. 2017, doi: 10.1016/j.future.2017.06.023.
[38] P. Gope, J. Lee, and T. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2831–2843, Nov. 2018, doi: 10.1109/TIFS.2018.2832849.

CONG XU received the B.S. degree in network engineering from the Shandong University of Science and Technology, in 2020, where she is currently pursuing the M.D. degree in computer science and technology. Since 2020, she has been conducting research of information security with the Data Security Laboratory, School of Computer Science and Technology, Shandong University of Science and Technology. Her research interests include wireless and mobile communications, protocol analysis and model detection, cryptography, and information security.

WENXUE WEI received the Ph.D. degree in network engineering. He teaches the courses which include the Internet of Things technology and application, network security theory and application, data communication and computing network, and network security technology. His research projects include intelligent storage management system based on the Internet of Things, network public opinion collection and analysis systems, and 863 key projects "Digital Mining Key Technology and Software Development." He has published more than 30 papers in important academic journals at home and abroad, including 11 papers included in SCI and EI and one monograph. His main research interests include information security, the Internet of Things engineering, and digital mine.

SHUANGSHUANG ZHENG received the B.S. degree in information and computing science from Taishan University, in 2020. She is currently pursuing the M.D. degree in software engineering with the Shandong University of Science and Technology. Since 2020, she has been working of image processing research with the Data Security Laboratory, School of Computer Science and Technology, Shandong University of Science and Technology. She has published a paper in the journal *Laser and Optoelectronics Progress*, in 2022. Her research interests include information security, image processing, and in-depth learning.