Provable Secure for the Ultra-lightweight RFID Tag Ownership Transfer Protocol in the Context of IoT Commerce

Jia-Qi Wang[1], Yun-Feng Zhang[2], and Dao-Wei Liu[3]
(Corresponding author: Jia-qi Wang)
Information Engineering Institute, Zhujiang College of South China Agricultural University[1]
Guangzhou 510900, China
Continuing Education College, Guangzhou City Construction College[2]
Department of Computer Science and Engineering, Guangzhou College of Technology and Business[3]
(Email: wjqldw@126.com)
(Received July 22, 2018; Revised and Accepted Dec. 28, 2018; First Online June 22, 2019)

Abstract

Under the business environment, the ownership of an (Radio Frequency Identification, RFID) RFID tag embedded item often shifts, and the ownership of the corresponding item must also be transferred, so the privacy of the original owner and the new owner needs to be protected during the transfer of ownership. In order to protect the privacy of tag's ownership during the transfer process, an RFID tag's ownership transfer protocol base on bitwise operation (*PSU-TOTP*) is proposed. The proposed protocol uses bitwise cross-synthesis and cross-connect operations to encrypt the transmitted information and reduce the amount of computation at the tag. The flag *FLAG* is introduced to record the ownership of the current owner. The abstract description of the security model and protocol is given, and the proposed protocol is comprehensively analyzed to meet the corresponding security requirements under the security model. Security analysis shows that the proposed protocol meets the security requirements for the tag ownership transfer. The formalization of GNY logic proves the correctness of the proposed protocol. Performance analysis shows that the proposed protocol can effectively reduce the computational load on the tag side and achieve the goal of reducing the tag's cost. *PSU-TOTP* is suitable for low-cost RFID systems.

Keywords: Cro-Link; Cro-Syn; Index Terms-IoT Business; Ownership Transfer; RFID; Ultra-Lightweight

1 Introduction

RFID is a kind of technology that automatically recognizes and obtains data. By embedding an RFID tag into a specific target, such as embedding an RFID tag in a bus card, the reader can recognize the specific target and read the data without directly contact [15]. The RFID tag has been widely used in manufacturing, transportation, wholesale and retail, and other fields because of its low cost, wide range of read and write, easy to carry, long service life and data encryption [25].

In practical applications, its owner will change frequently during the lifecycle of an RFID tag [12]. For example, an embedded RFID tag product, before it is not shipped, its ownership should be attributed to the manufacturer. When the product is sold by the manufacturer to the wholesaler, the ownership of the product is attributed to the wholesaler at this time. After the wholesaler resells the product to the retailer, the ownership of the product is owned by the retailer [28].

In the transfer process, the ownership of the RFID tag belongs to various owners, so we must protect the privacy of the corresponding owners [24].For example, after the manufacturer wholesales the product to the wholesaler, it must ensure that the manufacturer does not have permission to read the private information stored in the tag and that the wholesaler does not have access to the private information stored by the manufacturer [4]. In the process of the RFID tag ownership transfer, more and more scholars pay more attention to the security of the tag's private information, and also propose many ownership transfer protocols. However, there are more or less certain security flaws or large computations in these protocols [3].In order to solve the above problems, a provably secure and ultra-lightweight protocol for the RFID tag ownership transfer (*PSU-TOTP*) is proposed. The protocol uses bitwise operations to encrypt information so it can achieve the ultra-lightweight level. At the same time, the use of bitwise operation can effectively reduce the computational load of the tag. The introduction of flag *FLAG* can identify the current owner of ownership according to its value. Security and performance analysis

shows that the proposed protocol can meet the security requirements of ownership transfer and reach the goal of reducing the cost of the tag. *PSU-TOTP* can be appropriately used in existing RFID systems.

The first section of this article is the introduction, which tells that the ownership of RFID tag embedded items often changes during its life cycle. To protect the privacy of the tag, the ownership transfer protocol is proposed, which leads to the focus of this paper. The second section introduces some of the classic RFID tag ownership transfer protocol, and points out some of the shortcomings and deficiencies. The third section introduces the mathematical knowledge and symbol meaning used in the design process of *PSU-TOTP*. The fourth section establishes a security model of *PSU-TOTP* for RFID system. The fifth section gives an abstract description of the ownership transfer protocol for the applicable security model. The sixth section systematically describes the design steps of *PSU-TOTP*. The seventh section analyzes the security requirements that *PSU-TOTP* satisfies the transfer of ownership under the security model. The eighth section uses GNY formal logic to rigorously prove *PSU-TOTP*. The ninth section analyzes the performance of *PSU-TOTP* in detail from the aspects of the tag computation and storage space. The tenth section summarizes the whole paper, and gives the next research direction.

2 Related Works

Molnar *et al.* first proposed the concept of ownership transfer of RFID tags in 2005, and gave a protocol about the transfer of ownership of RFID tags in Reference [13], but it required both the original owner and the new owner to believe the trusted center, which made the protocol limited.

An ownership transfer protocol based on the hash function mechanism is proposed in Reference [14], but the analysis finds that the protocol can‘t resist denial of service attacks.

In Reference [6], a new protocol is proposed. Since the RFID tag returns a hash value of Kp and a random number each time, but the random number is generated by the tag itself, the attacker can reuse the return value, and thus can impersonate the tag, so the protocol can‘t resist impersonate attacks.

In Reference [10], a simple and efficient ownership transfer scheme is proposed, which is based on the existing problems in Reference [14] and can effectively solve the denial of service attacks existing in the original protocol.

The security and privacy protection requirements of the RFID tag ownership transfer protocols are given in Reference [16] and three sub-protocols are also given. However, the analysis shows that they can‘t resist the desynchronization attacks.

Later, Song himself proposed an improved solution to the existing deficiencies in Reference [16] and Reference [17], but the improved scheme still does not resist the desynchronization attacks and can‘t meet the security requirements of backward privacy protection.

Based on SQUASH, Reference [11] gives a scheme of ownership transfer. According to the analysis, the new owner can obtain the public and private keys shared by the original owner and the tag, so that the new owner can access the private information stored by the original owner. Therefore, the protocol does not meet the security requirements of forward privacy protection. An attacker could obtain the information and block the new owner from communicating with the tag. Through the remessage, the shared private key between the tag and the new owner may be out of synchronization, so the protocol can‘t resist the replay attacks and desynchronization attacks.

In Reference [5], a solution of ownership transfer is proposed and a security model is given. However, the analysis shows that the security model has some limitations. This assumption makes the solution unable to provide effective privacy protection in practical application.

In Reference [1], a provable secure RFID tag ownership transfer protocol is proposed. It is found that the transfer of some random number in the ownership transfer protocol is caused by transferring the plain text, which allows the attacker to obtain the random number through wiretapping and then to forcibly crack some of the tag's private information by brute-force means, so the protocol can‘t resist brute-force attack.

In Reference [27], an ownership transfer protocol based on the quadratic residue theorem is proposed. It is found that the protocol does not implement bidirectional authentication between the original owner and the tag, so the protocol can‘t ensure that the transferred tag is the target tag. Therefore, it can‘t resist impersonation attacks.

3 Related Knowledge Introduction

1) Bitwise cross-synthesis operation.

In this paper, to facilitate the use of the symbolic description, we use the symbol *CroSyn(X, Y)* to represent the cross-synthesis operator. The cross-synthesis operation *CroSyn(X, Y)* is defined as follows: Let X, Y, Z be three binary numbers all of which are even l bits, $X = x_1x_2\cdots x_L$, $Y = y_1y_2\cdots y_L$, $Z = z_1z_2\cdots z_L$, where $X \in \{0,1\}^l$, $Y \in \{0,1\}^l$, $Z \in \{0,1\}^l$. We obtain the i-th bit in the binary number X, and simultaneously obtain the $(i+1)$-th bit in the binary number Y. We perform different operations according to the Hamming weight for obtaining two bits, then place them in order and finally synthesize a new binary number Z. If Han-ming weight is odd, bitwise XOR operation is performed; otherwise, bitwise AND operation is performed [9, 19].

Cross-synthesis operation in the tag is implemented in the form of pointers, making it more efficient than the direct use of logic gates. Two pointers are introduced, one for P_X and the other for P_Y; where pointer P_X points to binary number X and pointer P_Y points to binary number Y.When the pointer P_X traverses from the first bit of the binary number X, the pointer P_Y starts traversing from the second bits of the binary number Y at the same time. Based on the traversal, we can get the value of the two bits, and judge their value of Hamming weight (if it's an odd value, perform XOR operation; if it's an even value, perform AND operation), and store the operation result in turn. Finally, calculate *CroSyn(X, Y)* to get a new binary number Z. For example, if $l = 8$, *X=11011001, Y=01100101*, then *CroSyn(X,Y)=11101101*. The specific process may refer to Figure 1.

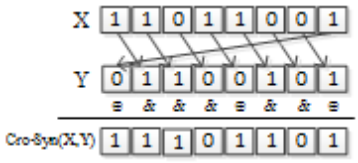

Figure 1: Flow chart of bitwise cross-synthesis operation

2) Bitwise cross-link operation.

In this paper, to facilitate the use of the symbolic description, we use the symbol *CroLink(X, Y)* to represent the cross-link operator. The cross-link operation *CroLink(X, Y)* is defined as follows: Let X, Y, Z be three binary numbers all of which are even l bits, $X = x_1x_2\cdots x_L$, $Y = y_1y_2\cdots y_L$, $Z = z_1z_2\cdots z_L$, where $X \in \{0,1\}^l$, $Y \in \{0,1\}^l$, $Z \in \{0,1\}^l$. We obtain the i-th bit and the $(i+1)$-th bit in the binary number X, and simultaneously obtain the i-th bit and the $(i+1)$-th bit in the binary number Y. According to the obtained four bits' Hamming weight we perform different operations, and then place them from low to high position to get a new left half of the binary Z. Similarly, place them from high to low position to get a new right half of the binary Z. Finally, the left and right half can be linked to get the binary number Z with the length of even l bits. If the Hamming weight is an odd value, perform AND operation; if it's an even value, perform XOR operation [26].

Cross-link operation in the tag is implemented as described below. Two pointers are introduced, one for P_1 and the other for P_2; where pointer P_1 points to the beginning of binary number X and pointer P_2 points to the beginning of binary number Y. When the pointer P_1 traverses from the beginning of the binary number X, the pointer P_2 starts traversing from the beginning of the binary number Y at the same time. We obtain the i-th bit and the $(i+1)$-th bit in the binary number X, and simultaneously obtain the i-th bit and the $(i+1)$-th bit in the binary number Y. Then judge the Hamming weight of the four bits. If it's an odd value, perform AND operation; if it's an even value, perform XOR operation. The calculated value is placed on the left half and the right half of the binary number, respectively, of which the left half is placed from low to high position and the right half is placed from high position to low position. Finally, link the left half and the right half and we can obtain the binary number Z with the length of even l bits.

Cross-link operation only needs traversal, bitwise OR operation, bitwise AND operation and the final link operation, which reduces the amount of system computing and storage, and achieves the ultra-lightweight level. For example, if $l = 8$, $X = \{11011001\}$, $Y = \{01100101\}$, then CroLink(X,Y)= $\{01111110\}$. The specific process may refer to Figure 2.

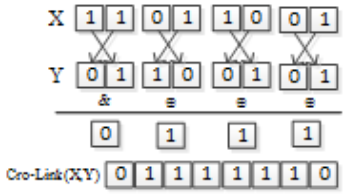

Figure 2: Flow chart of bitwise cross-link operation

4 Security Model

4.1 Communication Model

The existing RFID system generally consists of three parts: The tag T, the reader R and the database DB. The computing power of the tag T is limited, and its storage space is small, but the database DB has a strong data processing capabilities (such as data calculation, data query). The reader R is located between the tag T and the database DB. The reader R completes the communication process by forwarding the information of the tag T to the background database DB or forwarding the information of the database DB to the tag T [20–23]. In the current research, the RFID system is generally suitable for the following assumptions: The communication link between the tag T and the reader R is not secure, and the communication link between the reader R and the database DB is secure. The general communication process of RFID system is shown in Figure 3.

The tag ownership refers to the ability to identify the tag and be able to control all the information associated with the tag. The tag ownership transfer refers that the original owner no longer have the ownership of the tag, and the new owner has the control of the tag. Because the communication link between the reader and the database is safe and reliable, we can the both as a whole. In this paper, there are mainly three entities involved in the final

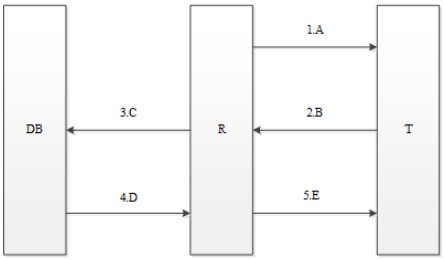

Figure 3: Flow chart of the communication process of RFID system

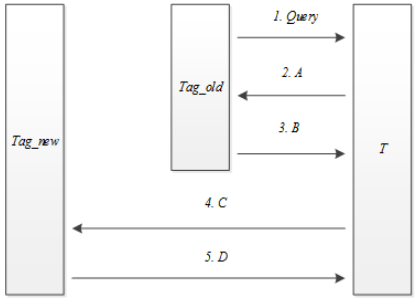

Figure 4: The flow chart of the abstract description of PSU-TOTP protocol

protocol communication by simplifying the research object in the process: The tag's original owner Tag_{old}, the tag's new owner Tag_{new} and the tag Tag.

4.2 Attack Model

In the RFID tag ownership transfer protocol, the method that an attacker A may use is mainly based on channel. For the channel-based attacks, we assume that the attacker A has full control over the communication channel between the original owner Tag_{old} and the tag Tag, and has complete control over the communication channel between the new tag's owner Tag_{new} and the tag Tag. The connotation of control here is that attacker A can arbitrarily read, tamper, delete, replay any message in the channel, and at the same time, he can initiate any conversation with any participant at any time [8]. Channel-based attacks mainly include replay attacks, man-in-the-middle attacks, privacy attacks, desynchronization attacks, tracking attacks and impersonation attacks.

4.3 Security Requirements

A secure and reliable RFID tag ownership transfer protocol needs to meet the following security requirements [18].

1) Backward privacy protection: After the ownership transfer completes, the tag's original owner Tag_{old} can no longer recognize the tag Tag, and can't access the session information between the Tag. Tag and the tag's new owner Tag_{new}.

2) Forward privacy protection: After the ownership transfer completes, the tag's new owner Tag_{new} can't access the session information between the tag Tag and the tag's original owner Tag_{old}.

3) Mutual authentication: During the transfer process, the ownership transfer can be performed only after the mutual authentications are completed between the tag Tag and the tag's original owner Tag_{old}, and between the Tag Tag and the tag's new owner Tag_{new}.

4) Anti-asynchronous attack: The attacker interrupts the ownership transfer protocol by any attack mode, so that the information between any two no longer maintains the synchronization. The protocol should be able to guarantee the success of the tagged Tag authentication again and realize the resynchronization of the information.

5) Anti-replay attack: An attacker deliberately replays a certain type of intercepted message in an attempt to analyze the private information stored in the tag Tag. The protocol should ensure that the attacker replays the message without affecting the communication among the three entities, and the attacker can't crack any useful privacy information.

5 The Abstract Description of the Ownership Transfer Protocol

The ownership transfer protocol mainly solves the problem that what the ownership transfer protocol belongs to. In order to guarantee the security of the information stored in the tag, after the tag's owner is changed, the tag's ownership must be changed respectively.

In order to make the protocol not lose the generality, the abstract description of the protocol is given in this section firstly, and the implementation steps of the protocol will be given in the next section. The abstract description of *PSU-TOTP* protocol is shown in Figure 4.

The abstract description of the *PSU-TOTP* protocol is as follows.

Step 1: The tag owner Tag_{old} sends a $Query$ command to the tag Tag and initiates a transfer of ownership request.

Step 2: After the tag Tag receives the information, it first looks at the value of the flag $FLAG$, and the current $FLAG = 0$ indicates that the ownership belongs to the original tag owner Tag_{old} and can start the ownership transfer process. Then calculate the value of A, and then send the value of A to the tag owner Tag_{old}.

Step 3: After the original owner Tag_{old} of the tag receives the information, the authenticity of the tag

Tag is determined by verifying the authenticity of A. If true, the original tag owner Tag_{old} calculates the value of B and then sends the value of B to the tag Tag; otherwise, the protocol terminates.

Step 4: After the tag Tag receives the information, the authenticity of the original tag Tag_{old} is identified by verifying that B is true or false. If it is true, the tag Tag calculates the value of C and sends the value of C to the tag's new owner Tag_{new}; otherwise, the protocol terminates.

Step 5: After receiving the information, the tag's new owner Tag_{new} identifies the authenticity of the tag Tag by verifying the authenticity of C. If true, the tag's new owner Tag_{new} calculates the value of D and passes the value of D to the tag Tag; otherwise, the protocol terminates.

Step 6: After the tag Tag receives the information, the authenticity of the tag's new owner Tag_{new} is discriminated by verifying the authenticity of D. If it is true, the tag Tag sets the value of the flag $FLAG$ to 1, indicating that the ownership transfer is successful and the current ownership belongs to the tag new owner Tag_{new}; otherwise, the protocol terminates.

6 The Design of PSU-TOTP

The reader and the database communicate through a secure link, so this article will see the two as a whole, so there are three communication entities involved in the *PSU-TOTP*: The tag's original owner Tag_{old}, the tag's new owner Tag_{new}, the tag Tag.

6.1 The Symbol Description

The description of the communication entity symbols and operation symbols involved in the *PSU-TOTP* is shown in Table 1 below.

Table 1: Symbol description

Symbol	Description
Tag	tag
Tag_i	the i-th tag
Tag_{old}	the tag's original owner
Tag_{new}	the tag's new owner
ID_{t_i}	the i-th tag's identifier ID
$ID_{t_{i_L}}$	the left half of ID_{i_t}
$ID_{t_{i_R}}$	the right half of ID_{i_t}
$K_{i_{old}}$	the shared private key generated between Tag_{old} and Tag_i
$K_{i_{new}}$	the shared private key generated between Tag_{new} and Tag_i
R_{Tag}	the random number generated by the tag
$R_{Tag_{new}}$	the random number generated by Tag_{new}
$R_{Tag_{old}}$	the random number generated by Tag_{old}
$\oplus$	bitwise XOR operation
$\|\|$	bitwise concatenation operation
&	bitwise AND operation
$CroLink(X, Y)$	cross-link operation
$CroSyn(X, Y)$	cross-synthesis operation
$FLAG$	the flag of the tag's ownership

6.2 Initial Assumptions and Initialization Phase

In order to make the *PSU-TOTP* design not lose the generality, the *PSU-TOTP* design also makes the following assumptions:

1) The communication link between the tag Tag and the tag's new owner Tag_{new} is not secure;

2) The tag Tag and the communication link between the tag's original owner Tag_{old} is not secure;

3) The communication link between the tag's original owner Tag_{old} and the tag's new owner Tag_{new} is secure. The attacker can listen to the communication messages in 1) and 2), and the attacker can't listen to the communication messages in 3). At the same time, it is assumed that the information stored in the tag Tag, the tag's original owner Tag_{old}, and the tag's new owner Tag_{new} is safe and reliable, and the attacker cannot know it in advance.

Before the tag's ownership transfer starts, the tag's original owner Tag_{old} and tag's new owner Tag_{new} both store all the tags' identifiers, because they don't know which specific tag is to be transferred. After the protocol is initialized, the tag Tag stores the following four-tuple data structure:$(ID_{t_{i_L}}, ID_{t_{i_R}}, K_{i_{old}}, K_{i_{new}})$. The tag's new owner Tag_{new} stores the following three-tuple data structure: $(ID_{t_{i_L}}, ID_{t_{i_R}}, K_{i_{new}})$. The tag's original owner Tag_{old} stores the following three-tuple data structure: $(ID_{t_{i_L}}, ID_{t_{i_R}}, K_{i_{old}})$. The flag $FLAG$ has an initial value of 0. When $FLAG = 0$, the ownership of the tag currently belongs to the tag's original owner. When $FLAG = 1$, the ownership of the tag has been transferred and at this time, the ownership belongs to the tag.

6.3 The Protocol Description

The *PSU-TOTP* flow chart is shown in Figure 9. The following describes the specific meanings of the formulas of $M0$ to $M7$ in Figure 9 as shown in Table 2, and then a description of the specific steps of the *PSU-TOTP* is given in conjunction with Figure 9.

The *PSU-TOTP* flow chart is shown in Figure 5.

The detailed steps of the *PSU-TOTP* process are de-

Table 2: Formula Description

Symbol	Description
$M0$	$ID_{t_{i_R}} \oplus R_{Tag}$
$M1$	$CroLink(R_{Tag}, K_{i_{old}})$
$M2$	$R_{Tag_{old}}$
$M3$	$CroSyn(R_{Tag_{old}}, K_{i_{old}})$
$M4$	$R_{Tag} \oplus K_{i_{new}}$
$M5$	$CroLink(R_{Tag}, ID_{t_{i_R}})$
$M6$	$R_{Tag_{new}} \oplus ID_{t_{i_R}}$
$M7$	$CroSyn(R_{Tag_{new}}, K_{i_{new}})$

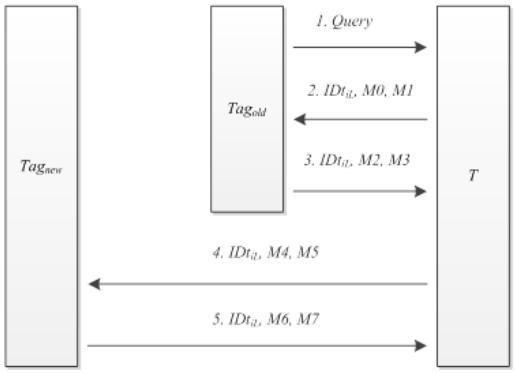

Figure 5: The flow chart of the ownership transfer

scribed below.

Step 1: The tag's original owner Tag_{old} sends a *Query* command to the tag Tag and initiates a transfer of ownership request.

Step 2: After the tag Tag receives the ownership transfer request, the value of the flag $FLAG$ is checked, and the current $FLAG$=0 indicates that the ownership belongs to the tag original owner Tag_{old}, and the ownership transfer can be started. Then the tag Tag generates a random number $R_{Tag} \in \{0,1\}^l$, and then calculates the values of $M0$, $M1$, and finally sends $M0$, $M1$, and $ID_{t_{i_L}}$ to the tag's original owner Tag_{old}.

Step 3: After the tag's original owner Tag_{old} receives the message, it first looks for $ID_{t_{i_L}}$ in the database. If $ID_{t_{i_L}}$ exists, Step 4 is performed; otherwise, the tag is forged by the attacker and *PSU-TOTP* terminates immediately.

Step 4: The tag's original owner Tag_{old} finds the $ID_{t_{i_R}}$ corresponding to $ID_{t_{i_L}}$, calculates the value of $ID_{t_{i_R}} \oplus M0$, then calculates the value of $M1$', and finally compares whether the values of $M1$' and $M1$ are equal.

If they are equal, the tag's original owner Tag_{old} correctly verifies the tag Tag, and Step 5 is performed; otherwise, the tag is false and the *PSU-TOTP* terminates immediately.

Besides, $M1' = CroLink(ID_{t_{i_R}} \oplus M0, K_{i_{old}})$.

Step 5: The tag's original owner Tag_{old} generates a random number $R_{Tag} \in \{0,1\}^l$, then calculates the values of $M2$, $M3$, and finally sends the $M2$, $M3$, and $ID_{t_{i_L}}$ to the tag Tag.

Step 6: After the tag Tag receives the information, it first calculates the value of $ID_{t_{i_R}} \oplus M2$, then calculates the value of $M3$', and finally compares whether the values of $M3$' and $M3$ are equal.

If they are equal, the tag Tag verifies that the tag's original owner Tag_{old} passes and proceeds to Step 7; otherwise, it indicates that the tag's original owner Tag_{old} is false and the *PSU-TOTP* terminates immediately.

Besides,$M3' = CroSyn(ID_{t_{i_R}} \oplus M2, K_{i_{old}})$.

Step 7: The tag Tag calculates the values of $M4$ and $M5$. Finally, $M4, M5$, and $ID_{t_{i_L}}$ are sent to the tag's new owner Tag_{new}.

Step 8: After receiving the information, the tag's new owner Tag_{new} first looks for $ID_{t_{i_L}}$ in the database. If $ID_{t_{i_L}}$ exists, Step 9 is performed; otherwise, the tag is forged by the attacker and the *PSU-TOTP* terminates immediately.

Step 9: The tag's new owner Tag_{new} finds the $ID_{t_{i_R}}$ corresponding to $ID_{t_{i_L}}$, calculates the value of $K_{i_{new}} \oplus M4$, then calculates the value of $M5$', and finally compares whether the values of $M5$' and $M5$ are equal.

If they are equal, the tag's new owner Tag_{new} correctly verifies the tag Tag and Step 10 is performed; otherwise, the tag is false and the *PSU-TOTP* terminates immediately.

Besides, $M5' = CroLink(K_{i_{new}} \oplus M4, ID_{t_{i_R}})$.

Step 10: The tag's new owner Tag_{new} generates a random number $R_{Tag_{new}} \in \{0,1\}^l$, then calculates the values of $M6$ and $M7$, and finally sends the $M6, M7$, and $ID_{t_{i_L}}$ to the tag Tag.

Step 11: After the tag Tag receives the information, it first calculates the value of $ID_{t_{i_R}} \oplus M6$, then calculates the value of $M7$', and finally compares whether the values of $M7$' and $M7$ are equal.

If they are equal, the tag Tag correctly verifies the tag's new owner Tag_{new} and Step 12 is performed; otherwise, it indicates that the tag's new owner Tag_{new} is false and the *PSU-TOTP* terminates immediately.

Besides, $M7' = CroSyn(ID_{t_{i_R}} \oplus M6, K_{i_{new}})$.

Step 12: The tag Tag sets the value of the flag $FLAG$ to 1, indicating that the ownership transfer is completed. At this time, the ownership of the tag is attributed to the tag's new owner Tag_{new}.

7 Security Analysis

7.1 Replay Attack

After the attacker listens on a complete communication session, all the communication messages can be obtained. The attacker tries to obtain the private information of the tag by replaying the message, but the attacker cannot succeed. In the communication message, each message in $M0$ to $M7$ is transmitted after encryption, not in plain text; and random numbers are used in the message encryption process. Random numbers are different each time, and at the same time they have unpredictability. Therefore, the attacker cannot replay messages for any private information.

7.2 Backward Privacy Protection

The new protocol can protect the privacy of the tag's new owner. After the ownership transfer is completed, the value of flag $FLAG$ in the tag is 1, is: $FLAG$=1, indicating that the ownership of the current tag belongs to the tag's new owner. In the process of ownership transfer, the value of $FLAG$ cannot be changed arbitrarily. The value of $FLAG$ will only change after the strict authentication and the correctness is determined. When the value of $FLAG$ changes, it indicates that the ownership transfer is completed; When $FLAG$=1, the tag's original owner sends a message to the tag for information access. At this time, the tag will recognize that the original owner of the current tag does not have the ownership of the tag according to the value of $FLAG$. Therefore, the access request of the tag's original owner will be rejected. So *PSU-TOTP* has backward privacy protection.

7.3 Forward Privacy Protection

The new protocol can protect the privacy of the tag's original owner from infringement. Before the transfer of ownership is completed, the value of the $FLAG$ is always 0, and it cannot be physically destroyed. If the value of $FLAG$ is not 1, the tag's new owner does not possess the ownership of the tag. If the tag's new owner sends a message to the tag, the tag will give an access deny response according to the value of $FLAG$. The tag's new owner has no right to access the tag, so the tag's new owner cannot obtain the communication message between the tag's original owner and the tag. Therefore, *PSU-TOTP* has forward privacy protection.

7.4 Bidirectional Authentication

Bidirectional authentication in *PSU-TOTP* refers to mutual authentication between the tag's original owner and the tag. It determines that the tag is the target tag of the transfer, and determines that the tag's original owner actually has the ownership of the target tag. Bidirectional authentication also refers to mutual authentication between the tag's new owner and tag. It determines that the tag is the target tag of the transfer, and determines that the tag's new owner is indeed the owner of the upcoming tag ownership.

Mutual authentication between the tag and the tag's original owner. The tag's original owner in the *PSU-TOTP* will confirm the authenticity of the tag for the first time in Step 3. Even if the attacker obtains the $ID_{t_{i_L}}$ through the interception method, the attacker still cannot pass the subsequent authentication. In Step 4, the tag's original owner will perform a second authentication on the tag. Since the attacker cannot obtain the $ID_{t_{i_R}}$ and $K_{i_{old}}$, the attacker cannot calculate the correct $M0$ and $M1$. The calculation can identify the authenticity of the tag. The authenticity of the tag to the tag's original owner is accomplished through Step 6: The attacker does not obtain the random number $R_{Tag_{old}}$ through the previous steps, and the attacker does not know the values of $ID_{t_{i_R}}$ and $K_{i_{old}}$; even the attacker can obtain the value of $ID_{t_{i_L}}$. However, the communication message does not use $ID_{t_{i_L}}$ in the calculation process, but uses $ID_{t_{i_R}}$. $ID_{t_{i_R}}$ does not have any relationship with $ID_{t_{i_L}}$.Therefore, the attacker cannot calculate the correct $M2$ and $M3$, so the tag can implement the tag's original owner's certification.

The certification between the tag and the tag's new owner. The tag's new owner's certification is completed in the Step 9. The random number R_{Tag} generated by the tag can be obtained by calculating the $M4$, which is then substituted into the $M5$ for comparison, and the authenticity of the tag can be identified according to the comparison result. Because the attacker does not know the values of $ID_{t_{i_R}}$, $K_{i_{new}}$, and R_{Tag}, the attacker cannot calculate the correct $M4$ and $M5$, thereby making it impossible for the attacker to pass the authentication in Step 9. The verification of the tag's new owner and the tag is performed in the Step 11. After receiving the $M6$ and $M7$, the tag first calculates the random number $R_{Tag_{new}}$ generated by the tag's new owner according to the $M6$, and then substitutes it into the $M7$ for verification; The user does not have the $ID_{t_{i_R}}, K_{i_{new}}$, or $R_{Tag_{new}}$ values, so the correct $M6$ and $M7$ values cannot be calculated.

In summary, *PSU-TOTP* enables bidirectional authentication between communicating entities.

7.5 Asynchronous Attack

Asynchronous attack, also known as desynchronization attack, refers to the attacker adopting some measure to make the shared private key between communication parties no longer maintain consistency. The attacker breaks

the consistency of shared private keys shared by both parties by using the following methods:

The parameters used by the two parties sharing the secret value update process are different;

The shared key is updated by one of the communication parties and the other party is not updated. In *PSU-TOTP*, no shared private key update mechanism is used, which makes it impossible for an attacker to use the above-mentioned method to destroy the shared private key between the two communication parties; the communication message is encrypted and then transmitted, and the communication message is calculated. Random numbers are useful in this process, and the random numbers are different each time, which assures that it is safe and reliable even if the shared private key is not updated. Therefore, the *PSU-TOTP* can resist the asynchronous attack.

7.6 Impersonation Attack

During the communication process, the attacker may fake the information exchange between any of the communication entities and other communication entities. Therefore, the protocol must be able to resist impersonation attacks by any of the attackers.

The attacker counterfeit the tag to communicate. When an attacker disguised as a tag to communicate, but the attacker does not know the following information: $ID_{t_{i_R}}, K_{i_{new}}, K_{i_{old}}$, so that the attacker can't calculate any of the correct value of $M0$ to $M7$. Even if the attacker has previously acquired all the messages of the previous round of communication by listening, and then replays the messages, the attacker still cannot obtain any private information because the attacker replays the message and the tag's original owner or the tag's new owner. A new random number will be generated and the new M_i value will be calculated at the same time, making the attacker's authentication fail. In the same way, the attacker would fake the tag's original owner to communicate or the counterfeit tag's new owner would fail to communicate, making it impossible to obtain any private information. Therefore, *PSU-TOTP* can resist the impersonation attack.

7.7 Brute Force Attack

The protocol must be able to resist the deliberately mandatory attack of the attacker, that is, the attacker uses a computing-intensive computer and cannot crack any useful private information.

By listening to a complete communication process, the attacker can obtain the following messages: $ID_{t_{i_L}}, Query, M0, M1, M2, M3, M4, M5, M6 and M7$. The attacker wants to use some of the useful information from the above information in the intercepted message, but the attacker cannot succeed. Here, messages $M0$ and $M1$ are selected as examples for analysis. In the formula $M0 = ID_{t_{i_R}} \oplus R_{Tag}$, the attacker only knows $M0$, and the two quantities of $ID_{t_{i_R}}$ and R_{Tag} are not known by the attacker, so the attacker cannot enumerate useful messages; meanwhile, the attacker is in the exhaustive process. As long as any one of $ID_{t_{i_R}}$ and R_{Tag} has an error, it is impossible for an attacker to obtain valid private information. In the formula $M1 = CroLink(ID_{t_{i_R}} \oplus M0, K_{i_{old}})$, even if the attacker substitutes M0 from the interception, the attacker can't exhaust any useful private information. First, the attacker does not know $ID_{t_{i_R}}$, $K_{i_{old}}$; second, the attacker does not know the details of each bit encryption in the cross-link encryption method, making it impossible for an attacker to violently crack private information. In the same way, the attacker analyzes and cracks $M2$, $M3$, $M4$, $M5$, $M6$ and $M7$, and cannot obtain private information. Therefore, the *PSU-TOTP* can resist the brute force attack.

Table 3 is a comparison of the security between PSU-TOTP and other RFID tag ownership transfer protocols.

8 GNY Logic Formal Proof

That the security of a complete protocol can be analyzed in words is far from enough. It can also be proved by the rigorous mathematical formulas. Based on this thought, in 1989 *Burrows* et al proposed a BAN formal logic analysis method, which was regarded as a milestone in the analysis of security protocols [2]. BAN logic is only concerned with the part of the protocol that is directly related to the authentication logic, and the rest is not a concern. It uses the rigorous mathematical rules to formalize the analysis and proof of the certification of the protocol. It also derives the target authentication step from the initialized hypothesis step of the protocol.

Because BAN form logic analysis has certain limitations, Gongli, *etc.* in 1990 put forward the GNY formal logic analysis method [7]. GNY formal logic analysis method is an expansion for BAN formal logic analysis method. GNY formal logic analysis method is more comprehensive than BAN logic analysis method, mainly in expanding the type and scope of analyzing the protocol. In this paper, the formal analysis and proof of *PSU-TOTP* protocol are carried out by using GNY formal logic analysis method.

1) Formal description of the protocol.

   To make the *PSU-TOTP* protocol easy to describe in the GNY formal logic language, the following convention is used: Tag_{old} indicates the tag's original owner, Tag indicates the tag, and Tag_{new} indicates the tag's new owner. The *PSU-TOTP* protocol flow is as follows:

   $Msg1$: $Tag_{old} \rightarrow Tag$: $Query$; indicates Tag receives message $\{Query\}$.

   $Msg2$: $Tag \rightarrow Tag_{old}$: $ID_{t_{i_L}}, M0 = ID_{t_{i_R}} \oplus R_{Tag}$, $M1 = CroLink(RTag, K_{i_{old}})$; indicates Tag_{old}

Table 3: Security comparison of authentication protocols

Attack Type	Reference[9]	Reference[11]	Reference[12]	Reference[13]	Reference[15]	Reference[16]	This Paper
Replay Attack	√	√	√	×	√	√	√
Backward Privacy Protection	√	√	×	√	√	√	√
Forward Privacy Protection	√	√	√	×	√	√	√
Bidirectional Authentication	√	√	√	√	√	√	√
Asynchronous Attack	√	×	×	×	√	√	√
Impersonation Attack	×	√	√	√	√	×	√
Brute Force Attack	√	√	√	√	×	√	√

Note: × means not provided; √ means provided

receives messages $\{M0, M1, ID_{t_{i_L}}\}$.

$Msg3$: $Tag_{old} \rightarrow Tag$: $ID_{t_{i_L}}$, $M2 = R_{Tag_{old}} \oplus ID_{t_{i_R}}$, $M3 = CroSyn(R_{Tag_{old}}, K_{i_{old}})$; indicates Tag receives messages $\{M2, M3, ID_{t_{i_L}}\}$.

$Msg4$: $Tag \rightarrow Tag_{new}$: $ID_{t_{i_L}}$, $M4 = R_{Tag} \oplus K_{i_{new}}$, $M5 = CroLink(R_{Tag}, ID_{t_{i_R}})$; indicates Tag_{new} receives messages $\{M4, M5, ID_{t_{i_L}}\}$.

$Msg5$: $Tag_{new} \rightarrow Tag$: $ID_{t_{i_L}}$, $M6 = R_{Tag_{new}} \oplus ID_{t_{i_R}}$, $M7 = CroSyn(R_{Tag_{new}}, K_{i_{new}})$; indicates Tag receives messages $\{M6, M7, ID_{t_{i_L}}\}$.

The above protocol is specified in the GNY formal logic language and can be described as follows:

$Msg1$: $Tag < * \, Query$.

$Msg2$: $Tag_{old} < * \, \{ID_{t_{i_L}}, M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(RTag, K_{i_{old}})\}$;

$Msg3$: $Tag < * \, \{ID_{t_{i_L}}, M2 = R_{Tag_{old}} \oplus ID_{t_{i_R}}, M3 = CroSyn(R_{Tag_{old}}, K_{i_{old}})\}$;

$Msg4$: $Tag_{new} < *\{ID_{t_{i_L}}, M4 = R_{Tag} \oplus K_{i_{new}}, M5 = CroLink(R_{Tag}, ID_{t_{i_R}})\}$;

$Msg5$: $Tag < *\{ID_{t_{i_L}}, M6 = R_{Tag_{new}} \oplus ID_{t_{i_R}}, M7 = CroSyn(R_{Tag_{new}}, K_{i_{new}})\}$.

2) Protocol initialization supposition.

The *PSU-TOTP* protocol is assumed to be as follows: Tag_{old}, Tag_{new}, and Tag indicate the main entities, that is, Tag_{old} indicates the tag's original owner, Tag indicates the tag, and Tag_{new} indicates the tag's new owner.

$Sub1$: $Tag \ni (ID_{t_{i_L}}, ID_{t_{i_R}}, K_{i_{old}}, K_{i_{new}}, R_{Tag})$; indicates Tag has the shared private key $K_{i_{old}}$, $K_{i_{new}}$, and has self-identifiers $ID_{t_{i_L}}$, $ID_{t_{i_R}}$ and self-generated random number R_{Tag}.

$Sub2$: $Tag_{old} \ni (ID_{t_{i_L}}, ID_{t_{i_R}}, K_{i_{old}}, R_{Tag_{old}})$; indicates Tag_{old} has the shared private key $K_{i_{old}}$, and has Tag's identifiers $ID_{t_{i_L}}$, $ID_{t_{i_R}}$ and self-generated random number $R_{Tag_{old}}$.

$Sup3$: $Tag_{new} \ni (ID_{t_{i_L}}, ID_{t_{i_R}}, K_{i_{new}}, R_{Tag_{new}})$; indicates Tag_{new} has the shared private key $K_{i_{new}}$, and has Tag's identifiers $ID_{t_{i_L}}$, $ID_{t_{i_R}}$ and self-generated random number $R_{Tag_{new}}$.

$Sup4$: $Tag_{old}| \equiv \#(R_{Tag_{new}}, R_{Tag_{old}}, R_{Tag})$; indicates Tag_{old} believes the random numbers $R_{Tag_{new}}$, $R_{Tag_{old}}$, R_{Tag} are fresh.

$Sup5$: $Tag| \equiv \#(R_{Tag_{new}}, R_{Tag_{old}}, R_{Tag})$; indicates Tag believes the random numbers $R_{Tag_{new}}$, $R_{Tag_{old}}$,R_{Tag} are fresh.

$Sup6$: $Tag_{new}| \equiv \#(R_{Tag_{new}}, R_{Tag_{old}}, R_{Tag})$; indicates Tag_{new} believes the random numbers $R_{Tag_{new}}$,$R_{Tag_{old}}$, R_{Tag} are fresh.

$Sup7$: $Tag| \equiv Tag_{new} \overset{\{ID_{t_{i_R}}, ID_{t_{i_L}}, K_{i_{new}}\}}{\leftrightarrow} Tag$; indicates that Tag believes the information $ID_{t_{i_L}}$, $ID_{t_{i_R}}$, $K_{i_{new}}$ shared between the Tag and Tag_{new}.

$Sup8$: $Tag| \equiv Tag_{old} \overset{\{ID_{t_{i_R}}, ID_{t_{i_L}}, K_{i_{old}}\}}{\leftrightarrow} Tag$; indicates that Tag believes the information $ID_{t_{i_L}}$, $ID_{t_{i_R}}$, $K_{i_{old}}$ shared between the Tag and Tag_{old}.

$Sup9$: $Tag_{old}| \equiv Tag \overset{\{ID_{t_{i_R}}, ID_{t_{i_L}}, K_{i_{old}}\}}{\leftrightarrow} Tag_{old}$; indicates that Tag_{old} believes the information $ID_{t_{i_L}}$, $ID_{t_{i_R}}$, $K_{i_{old}}$ shared between the Tag_{old} and Tag.

$Sup10$: $Tag_{new}| \equiv Tag \overset{\{ID_{t_{i_R}}, ID_{t_{i_L}}, K_{i_{new}}\}}{\leftrightarrow} Tag_{new}$; indicates that Tag_{new} believes the information $ID_{t_{i_L}}$, $ID_{t_{i_R}}$, $K_{i_{new}}$ shared between the Tag_{new} and Tag.

3) Protocol proof target.

The *PSU-TOTP* protocol has four targets for proof, which are mainly the trust in the freshness of mutual information exchange with the tag and the tag's new owner, and with the tag and the tag's original owner. The proof formulas of the goal are as follows:

$Goal1$: $Tag_{old}| \equiv Tag| \sim \#\{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$;

$Goal2$: $Tag| \equiv Tag_{old}| \sim \#\{M2 = R_{Tag_{old}} \oplus ID_{t_{i_R}}, M3 = CroSyn(R_{Tag_{old}}, K_{i_{old}})\}$;

$Goal3$: $Tag_{new}| \equiv Tag| \sim \#\{M4 = R_{Tag} \oplus K_{i_{new}}, M5 = CroLink(R_{Tag}, ID_{t_{i_R}})\}$;

$Goal4{:}Tag| \equiv Tag_{new}| \sim \#\{M6 = R_{Tag_{new}} \oplus ID_{t_{i_R}}, M7 = CroSyn(R_{Tag_{new}}, K_{i_{new}})\}$;

4) Protocol proof process.

The proof of the *PSU-TOTP* protocol is based on the initial supposition. The proof process follows the logical reasoning rules, being-told rules, freshness rules and possession rules in Reference [24]. The message interpretation rules follow the written form of the GNY logical reasoning rule in Reference [24], which are represented by T, P, F, I respectively.

Since the proof process of $Goal2$: $Tag| \equiv Tag_{old}| \sim \#\{M2 = R_{Tag_{old}} \oplus ID_{t_{i_R}}, M3 = CroSyn(R_{Tag_{old}}, K_{i_{old}})\}$; $Goal3$: $Tag_{new}| \equiv Tag| \sim \#\{M4 = R_{Tag} \oplus K_{i_{new}}, M5 = CroLink(R_{Tag}, ID_{t_{i_R}})\}$, $Goal4$: $Tag| \equiv Tag_{new}| \sim \#\{M6 = R_{Tag_{new}} \oplus ID_{t_{i_R}}, M7 = CroSyn(R_{Tag_{new}}, K_{i_{new}})\}$ is similar to the proof process of $Goal1$: $Tag_{old}| \equiv Tag| \sim \#\{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$.Therefore, Goal1 is used as an example in this section. The proof process is described as follows.

$\because$ $RuleP1$: $\frac{P<X}{P \ni X}$ $and Msg2 : Tag_{old} < * \{ID_{t_{i_L}}, M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$;

$\therefore$ $Tag_{old} \ni \{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}}\}$.

$\because$ $Rule1F1$: $\frac{P|\equiv(X)}{P|\equiv(x,y), P|\equiv \#F(X)}$ and $Sup5$: $Tag| \equiv \#(R_{Tag_{new}}, R_{Tag_{old}}, R_{Tag})$;

$\therefore$ $Tag_{old} = \#\{M0 = ID_{t_{i_R}} \oplus RTag, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$.

$\because$ $RuleP2$: $\frac{P \ni X, P \ni Y}{P \ni (X,Y), P \ni F(X,Y)}$, $Sup1$: $Tag \ni (ID_{t_{i_L}}, ID_{t_{i_R}}, K_{i_{old}}, K_{i_{new}}, R_{Tag})$ and $Sup2$: $Tag_{old} \ni (ID_{t_{i_l}}, ID_{t_{i_R}}, K_{i_{old}}, R_{Tag_{old}})$;

$\therefore$ $Tag_{old} \ni \{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$.

$\because$ $RuleF10$: $\frac{P|\equiv(X), P \ni X}{P|\equiv \#(H(X))}$ and the derived formula $Tag_{old} = \#\{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$, $Tag_{old} \ni \{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$;

$\therefore$ $Tag_{old}| \equiv \#\{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$.

$\because$ Rule $I3$: $\frac{P<H(X,<S>)>, P \ni (X,S), P|\equiv P \leftrightarrow Q, P|\equiv \#(X,S)}{P|\equiv Q|\sim(X,S), P|\equiv Q \sim H(X,<S>)}$;

$\because$ $Sup8$: $Tag| \equiv Tag_{old} \overset{ID_{t_{i_R}}, ID_{t_{i_L}}, K_{i_{old}}}{\leftrightarrow} Tag$, $Sup9$: $Tag_{old}| \equiv Tag \overset{ID_{t_{i_R}}, ID_{t_{i_L}}, K_{i_{old}}}{\leftrightarrow} Tag_{old}$ and $Msg2$: $Tag_{old} < *\{ID_{t_{i_L}}, M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$;

$\therefore$ $Tag_{old}| \equiv Tag \sim \{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$.

$\because$ The definition of freshness and the derived formula $Tag_{old} = \#\{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}, Tag_{old} \mid = Tag \sim \{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$;

$\therefore$ $Goal1$: $Tag_{old}| \equiv Tag| \sim \#\{M0 = ID_{t_{i_R}} \oplus R_{Tag}, M1 = CroLink(R_{Tag}, K_{i_{old}})\}$ is proved.

9 Performance Analysis

In the process of the tag ownership transfer, there are three communication entities involved: The tag, the original owner of the tag, and the new owner of the tag. The original owner of the tag and the new owner of the tag all include the database. Therefore, the two parts of the communication entities have powerful query capabilities, data calculation capabilities, and storage space. The tag does not have the above capabilities, there-by, the performance analysis will focus on three aspects of the tag calculation, storage space, and session times. Table 4 shows the performance comparison between *PSU-TOTP* and other RFID tag ownership transfer protocols.

Table 4: Performance comparison of ownership transfer protocol

Ref.	Calculation	Storage	Session Times
Ref[9]	5P+H	$2l$	5
Ref[11]	13P+6H	$1l$	7
Ref[12]	2P+7H	$3l$	6
Ref[13]	6P+M	$3l$	5
Ref[15]	3P+H+M	$3l$	5
Ref[16]	3H+3M	$4l$	6
This Paper	4P+2N+2Q	$3l$	5

In Table 4, H represents a hash function operation. P represents a bitwise operation. M represents a modular square operation. Q represents a cross-synthesis operation. N represents a cross-link operation.

1) The tag calculation. Compared to other references, the *PSU-TOTP* in this paper does not encrypt the information by using a large computational hash function or a modular squaring operation. Instead, it selects ultra-lightweight bitwise operation to encrypt its transmission information, which can greatly reduce the amount of computation on the tag. The computational complexity of the tag in this paper differs from other references by more than one order of magnitude, which can greatly reduce the computational cost of the tag. At the same time, the bitwise AND operation and the bitwise XOR operation are also used in the cross-synthesis operation and the cross-link operation, so that some circuits can be shared among the four operations. The cost of the tag will also be reduced.

2) The storage space on the tag. Set the ID_t, $K_{i_{new}}$ and $K_{i_{old}}$ with the length of l bits, so the storage space

on the tag only needs $3l$ bits. Compared with other references, $3l$ storage space has been improved. In the protocol of this paper, only one random number is generated at the tag. In other references, multiple random numbers are generated at the tag. Therefore, the overall cost of the storage space in this paper is not too large and it is acceptable.

3) Session times. Relative to references [16, 17, 27], the protocol in this paper reduces the times of session, which can reduce the cost of communication time of the entire protocol. Although the times of session in this paper are equivalent to the references [1, 6, 11], this protocol can make up for the security flaws existing in other protocols.

To sum up, the protocol in this paper can effectively reduce the tag calculation and it's much improved compared to other protocols. In terms of the storage space and the session times, the protocol in this paper is not improved much, but it can make up for the security flaws existing in other protocols, so this protocol still has some advantages, which is suitable for low-cost RFID systems.

10 Conclusions

In the life cycle of RFID tag, ownership often changes. In order to ensure the security of the privacy of the tag, an ultra-lightweight RFID tag ownership transfer protocol *PSU-TOTP* based on bitwise operation is proposed. Based on the analysis of the deficiencies in existing protocols, the paper proposes an improved protocol *PSU-TOTP*. It introduces cross-synthesis operation and cross-link operation to encrypt the transmission information so that the protocol can achieve ultra-lightweight levels. At the same time, the use of bitwise operations described above can effectively reduce the tag calculation and reduce the cost of tag. According to the different values of flag $FLAG$, the corresponding operation is performed to ensure the uniqueness and definiteness of the ownership. The security analysis shows that the $PSU-TOTP$ can meet the requirements for the ownership transfer. GNY formal logic proves the accuracy of $PSU-TOTP$. Comprehensive performance analysis shows the advantages of *PSU-TOTP* and achieve the goal of reducing tag calculation, so it is suitable for low-cost RFID systems. The next research directions of the paper are: To optimize the *PSU-TOTP* protocol in order to reasonably reduce the communication traffic; To implement the prototype of the *PSU-TOTP* RFID system and figure out the total number of required gate circuits and the time of a complete communication, so as to combine the theory with practice.

References

[1] Y. Bian-qing and L. Ji-qiang, "Provable secure ownership transship protocol for RFID tag," *Journal of Communications*, vol. 36, no. 8, pp. 83–90, 2015.

[2] M. Borrows, M. Abadi and R. M. Needham, "A logic of authentication," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, pp. 233–271, 1989.

[3] J. S. Chen, C. Y. Yang, and M. S. Hwang, "The capacity analysis in the secure cooperative communication system," *International Journal of Network Security*, vol. 19, no. 6, pp. 863–869, 2017.

[4] Y. C. Chen, W. L. Wang, M. S. Hwang, "RFID authentication protocol for anti-counterfeiting and privacy protection", in *The 9th International Conference on Advanced Communication Technology*, pp. 255–259, 2007.

[5] K. Elkhiyaout, E. O. Blass and R. Molva, "ROTIV: RFID ownership transfer with issuer verification[A]," in *LNCS7055: 7th International Workshop on RFID Security and Privacy*, pp. 163–182, 2012.

[6] S. Fouladgar and H. Afifi, "An efficient delegation and transfer of ownership protocol for RFID tags[C]," in *Proc of the 1st Int EURASIP Workshop on RFID Technology*, pp. 10–14, 2007.

[7] L. Gong, R. Needham, and R. Yanhalom, "Reasoning about belief in cryptographic protocols," *IEEE Computer Society Symposium in Security and Privacy*, pp. 234–248, 1990.

[8] M. S. Hwang, T. H. Sun, and C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service," *Journal of Circuits Systems and Computers*, vol. 26, no. 5, 2017.

[9] M. S. Hwang, W. Y. Chao, C. Y. Tsai, "An improved key management scheme for hierarchical access control," *International Journal of Network Security*, vol. 19, no. 4, pp. 639–643, 2017.

[10] P. Jappinen and H. Hamalainen, "Enhanced RFID security method with ownership transfer[C]," in *Proc of Int Conf on Computational Intelligence and Security*, pp. 382–385, 2008.

[11] Y. M. Jin, H. P. Sum and Z. Guan, "Ownership transfer protocol for RFID tag," *Journal of Computer Research and Development*, vol. 48, no. 8, pp. 1400-1405, 2011.

[12] D. W. Liu, J. Ling, X. Yang, "An improved RFID authentication protocol with backward privacy," *Computer Science*, vol. 43, no. 8, pp. 128–130, 2016.

[13] D. Molnar, A. Soppera and D. Wagner, "A scalable, delegatable pseudonym protocol enabling ownership transfer of rfid tags[A]," in *12th International Workshop on Selected Areas in Cryptography[C]*, pp. 276–290, 2006.

[14] K. Osaka, T. Takagi and K. Yanazaki, "An efficient and secure RFID security method with ownership transfer[A]," in *Proceedings of IEEE International Conference on Computational Intelligence and Security[C]*, pp. 1090–1095, 2006.

[15] H. T. Pan, C. S. Pan, S. C. Tsaur, and M. S. Hwang, "Cryptanalysis of efficient dynamic id based remote

user authentication scheme in multi-server environment using smart card," in *12th International Conference on Computational Intelligence and Security (CIS'16)*, pp. 590–593, Dec. 2017.

[16] B. Song, *RFID Tag Ownership Transfer [EB/OL]*, 2008. (http://rfidsec2013.iaik.tugraz.at/RFIDSec08/Papers/Publication/15%20-%20Song%20-%20Ownership%20Transfer%20-%20Paper.pdf)

[17] B. Song, and C. J. Mitchell, "Scalable RFID security protocols supporting tag ownership transfer," *Computer Communications*, vol. 34, no. 4, pp. 556–566, 2011.

[18] C. Y. Tsai, C. Y. Liu, S. C. Tsaur, and M. S. Hwang, "A publicly verifiable authenticated encryption scheme based on factoring and discrete logarithms," *International Journal of Network Security*, vol. 19, no. 3, pp. 443–448, 2017.

[19] Y. L. Wang, J. J. Shen, and M. S. Hwang, "An improved dual image-based reversible hiding technique using LSB matching," *International Journal of Network Security*, vol. 19, no. 5, pp. 858–862, 2017.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] N. I. Wu and M. S. Hwang, "Development of a data hiding scheme based on combination theory for lowering the visual noise in binary images," *Displays*, vol. 49, pp. 116–123, 2017.

[25] R. Xie, J. Ling, and D. W. Liu, "Wireless key generation algorithm for rfid system based on bit operation," *International Journal of Network Security*, vol. 20, no. 5, pp. 938–950, 2018.

[26] R. Xie, B. y. Jian, and D. w. Liu, "An improved ownership transfer for RFID protocol," *International Journal of Network Security*, vol. 20, no. 1, pp. 149–156, 2018.

[27] C. Xiuqing, C. Tianjie, and Z. Jingxuan, "Provable secure ownership transship protocol for RFID tag," *Journal of Electronics & Information Technology*, vol. 36, no. 8, pp. 83–90, 2015.

[28] C. Y. Yang, T. Y. Chung, M. S. Hwang, C. Y. Li, and J. F. J. Yao, "Learning performance evaluation in elearning with the web-based assessment," in *8th iCatse Conference on Information Science and Applications (ICISA'17), Lecture Notes in Electrical Engineering*, pp. 645–651, Mar. 2017.

Biography

Jia-qi Wang ,received her Master's degree in computer science from Sun Yat-sen University (China) in December 2012. She is a lecturer in Information Engineering Institute in Zhujiang College of South China Agricultural University, director of the Teaching and Research Office of electronic business and information management. Her current research interest fields include Electronic Business and computer applications.

Yun-feng Zhang, from 2009 to 2011 he was an expert in RFID tag in The Automation Society of Guangdong Province. Now he is a teacher in Continuing Education College of Guangzhou City Construction College, and his main research interest is information security.

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.