Wireless Pers Commun (2014) 79:1941–1958
DOI 10.1007/s11277-014-1966-x

A Low-Cost RFID Authentication Protocol Against Desynchronization with a Random Tuple

Lijun Gao · Maode Ma · Yantai Shu · Feng Lin · Lei Zhang · Yuhua Wei

Published online: 15 October 2014

Abstract Radio frequency identification (RFID) technology will become one of the most popular technologies to identify objects in the near future. However, the major barrier that the RFID system is facing presently is the security and privacy issue. Recently, a lightweight anti-desynchronization RFID authentication protocol has been proposed to provide security and prevent all possible malicious attacks. However, it is discovered that a type of desynchronization attacks can successfully break the proposed scheme. To overcome the vulnerability under the desynchronization attacks, we propose a low-cost RFID authentication protocol which integrates the operation of the XOR, build-in CRC-16 function, permutation, a random tuple and secret key backup technology to improve the security functionality without increasing any cost than the utralightweight protocols. The analysis shows that our proposal has a strong ability to prevent existing malicious attacks, especially the desynchronization attacks.

Keywords Low-cost · Permutation · Desynchronization · A random tuple

1 Introduction

Radio frequency identification (RFID) is a technology for automatic identification of objects and people [1]. There are many application scenarios to employ the RFID technology with aims to promote the production efficiency in the areas such as agriculture, industry, trans-

L. Gao · Y. Shu · L. Zhang
School of Computer Science and Technology, Tianjin University, Tianjin, China

L. Gao · F. Lin · Y. Wei
Department of Computer Science and Technology, Shenyang Aerospace University, Shenyang, China

M. Ma (✉)
School of Electrical and Electronic Engineering, Nanyang Technological University,
Nanyang Avenue, Singapore
e-mail: maodema@163.com

portation, education, military and defence, and government, etc. With much more and more applications, the RFID technology will become one of the most popular technologies to improve economy and social lives in the near future. A RFID system contains three types of key elements: RFID tags, RFID readers, and a back-end database server which has the ability to identify objects with increased speed and accuracy. The reader is used to query the tag identify (TID) and forwards it to the back-end server. Once the tag is found valid, the back-end server will check the information kept by the tag for further processing. The RFID tags have experienced two generation of the development. And it is widely believed that Generation 2 (Gen2) tags are the major tags used currently for the development of RFID applications because the effective reading range is relatively larger [2]. In a typical RFID system, the information transmitted in the air between the tag and the reader could easily be intercepted and eavesdropped due to its radio transmission nature, which indicates that the security issues will be the major concerns to block further development of the RFID applications, especially, in the military area or some other secrecy sensitive areas.

Currently, the RFID security and privacy protection mechanisms mainly can be classified into two major categories: physical approaches and encryption mechanisms and protocols. The proposals on the physical security mechanisms for the RFID tags mainly include the Faraday Cage [3], kill command mechanism [4], the locker tag [5]. Further research results indicate that although the physical security approaches can achieve some degree of security, it will cause the increase of the cost of an entire RFID system. On the other hand, the encryption technology based security protocols have shown to be more attractive to the development of the RFID systems, which will be soon widely adopted. The encryption technology based security protocols can be classified into four classes in Chien [6]. They are full-fledged, simple, lightweight and ultralightweight RFID authentication protocols.

In terms of simple protocols, the hash-Lock scheme has been introduced in [3,7] used $metaID = \mathrm{H}(K)$ to hide the real ID of a tag, where K is the shared secret between the tag and the back-end server, H is a one-way hash function. Although this scheme offers certain level of reliability at a low cost, an adversary can easily track the tag via its *metaID* and thus the transaction secret or privacy would be at risks. Furthermore, since the key shared between the tag and the back-end server is sent in plaintext, even an inactive adversary can easily sniff the transmission channel to spoof the tag information. The hash based ID variation protocol in Henrici and Muller [8] is similar to the hash chain protocol, which uses a random number to refresh the tag identifier dynamically. The random number increases after each successful authentication session so that this protocol is able to defend against the replay attacks. The protocol can also resolve the location attacks by making the ID of a tag randomized in every interrogation. It is also reliable to prevent data loss because it can restore the data from the previous record. Unfortunately, this protocol cannot resist the man-in-the-middle attacks, the intermittent position tracing attacks defined in Gao et al. [9], and the desynchronization attacks reported in Zhou et al. [10], where a novel RFID security protocol (RIPTA-DA) has been designed, which employs a stochastic dynamic multi-key mechanism to encrypt the information and employs the noise disturbance technology to overcome the vulnerabilities under the both attacks.

On the other hand, in terms of lightweight protocols, Hopper and Blum (HB), HB+, HB++ protocols have been proposed in [11–14] as a family, which has used the learning parity in the presence of noise (LPN) to provide stronger security functionality. However, it is found that if an aggressor replays challenges on a tag with $\mathrm{O}[(1-\eta)/(1-2\eta)^2]$, where η is a noise parameter. Each tag has a noise generator, by which the probability of generating a noise is $v = \{0, 1 | \operatorname{prob}[v = 1] = \eta\}, \eta \in (0, 1/2)$, where v is a vector, which is a binary string, while η is the probability of the number of "1" in the binary string v times. It is possible to obtain the value of $a \cdot x$, where $\cdot$ is a point multiplication operation, with very high probability. A synchronization-based communication protocol for RFID devices has been presented in Duc et al. [15]. The protocol targets to protect the Gen-2 RFID tags which support only simple cryptographic primitives like pseudo random noise generation (PRNG) and cyclic redundancy check (CRC). It can prevent the cloned tags and the malicious readers from the impersonating attacks and abusing legitimate tags, respectively. In addition, the protocol is able to provide that each RFID tag emits a different bit string (pseudonym) when receiving each query from different readers. Therefore, it makes possible for the tracking activities and personal preferences of a tag's owner impractical to provide the user's privacy. It's possible for a malicious reader can get $M_1 = \mathrm{CRC}(\mathrm{TID}||r_1) \oplus K_i$, and $M_2 = \mathrm{CRC}(\mathrm{TID}||r_2) \oplus K_i$, where k represents string concatenation and r_1, r_2 are nonce values. In this way, the attacker can identify a tag by the following way $M_1 \oplus M_2 = \mathrm{CRC}(\mathrm{TID}||r_1) \oplus \mathrm{CRC}(\mathrm{TID}||r_2)$. Once the tag is queried by a valid reader which causes the key update, the attacker can restart the attack. Although the protocol is defective, the application of CRC function in the design has opened a new way to design a low cost RFID system. In Doss et al. [16,17], three solutions have been proposed for the authentication and privacy in the RFID systems employing the quadratic residues technology. But due to the usage of high cost hash functions and complex encryption algorithms, they are not suitable to the low cost RFID systems.

In terms of ultralightweight protocols, a minimalist mutual-authentication protocol (M^{2}AP) for low cost RFID tags has been proposed in Lopez and Castro [18] using some simple logical operations such as XOR, OR, AND, and sum of modulo. A tag and a reader can share a pseudonym session identifier (SID) and four keys. During each session, the reader generates two random numbers. By this protocol, the tag verifies the reader by checking the value extracted from the first two messages. The tag then responds to the reader if it is correct. Both SID and four keys must be updated after each session to provide forward secrecy. Recently, a desynchronization attack to break the M^{2}AP protocol has been reported in Bárász et al. [19]. By this attack, an adversary could discover the tag's identity and some shared secrets in two rounds of eavesdropping. Furthermore, the attacker can undertake desynchronization attacks by using the known keys.

An interesting lightweight authentication protocol has been proposed providing strong authentication and strong integrity (SASI) for the low cost RFID systems in Chien [6]. An index-pseudonym, the tag's private identification (ID), and two keys (k_1, k_2) are stored both on the tag and in the back-end database. Simple logical functions such as bitwise XOR, bitwise AND, bitwise OR, addition and left rotate function are required on the tag. Additionally, a PRNG is required at the reader. The proposed scheme is ultralightweight, while the active tracking attacks are possible among two valid readers because the IDS in SASI is a static value. It is also shown that a desynchronization attack on the SASI scheme can succeed with at most 96 trials [20]. Gossamer protocol has been introduced in Peris-Lopez et al. [21], which has a very good security performance to keep the confidentiality and integrity of data in the authentication procedure with a forward security by a rotation operation, which is a combined function with circular shift function and the Mixbits function. Gossamer protocol has shown to have an extremely lightweight nature, as only bitwise right shift and additions functions have been employed. The abovementioned protocols have certain security functionality equipped with simple operations at a low cost, while they are not able to resist some type of the desynchronization attacks [22].

A new ultralightweight RFID authentication protocol with permutation (UAPP) has been proposed in Tian et al. [23]. It has avoided using unbalanced OR and AND operations and has introduced a new operation named permutation. A tag involves only with three operations: bitwise XOR, left rotation and permutation. The performance evaluation illustrates that since the UAPP scheme only uses fewer resources on the tags in terms of computation operation, storage requirement and necessary communication, the total cost of the UAPP scheme is much lower. The security analysis in Tian et al. [23] has claimed that the UAPP scheme can resist to all possible existing attacks. However, one type of the desynchronization attacks has been found to be able to break the protocol. Based on the solution in Tian et al. [23], we have proposed a security authentication protocol to prevent the desynchronization attacks with CRC function and permutation function to improve the security functionality of the authentication protocols without increase any hardware cost in Paolo and Santis [24].

It is obvious that the simple authentication protocols can effectively resist various malicious attacks by using complicated hash functions resulting in a higher cost. Although the lightweight authentication protocols have not been equipped with complex hash functions, the security cost is relative higher due to the random number generator introduced. On the other hand, the security functionality of the ultralightweight RFID authentication protocols are questionable. In this paper, the anti-desynchronization RFID authentication protocol reported in Zhou et al. [10] will be reviewed to explore its vulnerability under one type of the desynchronization attacks. Further, to overcome the vulnerability under the desynchronization attacks, we propose a low cost RFID authentication protocol authentication protocol with a random tuple (APRT) which integrates the operation of the XOR, the CRC-16 function, the permutation function, random tuples and the secret key backup technology to improve the security functionality without increasing much cost than the existing utralightweight protocols. The analysis shows that our proposal has a strong ability to prevent the existing malicious attacks, especially the desynchronization attacks.

The remainder of the paper is organized as follows. The scheme in Zhou et al. [10] is reviewed to explore its vulnerability under one type of the desynchronization attacks in Sect. 2. In Sect. 3, the UP^{2}RT scheme is presented to overcome the flaws in the scheme in Zhou et al. [10]. The security analysis on the UP^{2}RT scheme is presented in Sect. 4. Then, in Sect. 5, the performance evaluation on the proposed UP^{2}RT is demonstrated in terms of the computation operation, the storage requirement, the communication cost and the capability to resist malicious attacks. Finally, the paper is concluded in Sect. 6.

2 Vulnerability in the Protocol in Zhou et al. [10]

A desynchronization attack is an active malicious attack with aim to make the attacked RFID system lose desynchronization without an ability to be authenticated as normal. The RFID ultralightweight protocols are mainly used for special circumstances, such as library, warehouse and hospital. By the desynchronization attacks, the attacker can make the library, warehouse and hospital out of working as normal, where the system could be paralyzed under the desynchronization attacks. We have also found some references such as [9,10,20,22,23,25,26] to address against the desynchronization attacks. So we believe that the research results against the desynchronization attacks are significant.

There are two types of the desynchronization attacks, which are retransmission desynchronization attacks and bit tamper desynchronization attacks [6]. An retransmission desynchronization attack refers to the interception action in the secret key updating phase of authentication process. Suppose the database send a message with variables to a tag, then the database update the secret key. An attacker could interrupt the message so that the tag will not be able to update its variables, which will cause that the secret keys at the database and the tag are not able to be synchronized. A bit tamper desynchronization attack is that, for example, the database will reply with A, B, C to the tag. An attacker's goal is to forge a tuple

(A′, B′, C′) that is accepted by the tag. The attacker makes $A' = A^*$ where A^* is the flip of the kth bit in A, $B' = B$, and $C' = C^*$, where C^* is the flip of the kth bit in C. Then, the attacker replies the tag with (A′, B′, C′). In this way, C′ always flips and C^* from the attacker will pass the verification process of the tag. In the next authentication, when the reader tries to read the tag, the tag can be found in the database. But the reader will be rejected by the tag because the secret key in the tag is no longer synchronized with the database.

In order to show the vulnerability of the protocol in Zhou et al. [10] under the retransmission desynchronization attacks, we need first to review the operations of the protocol as shown in Fig. 1.

2.1 Review the Protocol in Zhou et al. [10]

Step 1: Reader → Tag(Challenge Message): First, the reader generates a random number r and challenges the tag with it.
Step 2: Tag → Reader(Responding Message): While receiving the challenge, the tag responds the reader with $IDS = H(Key_i)$, $H(T_i \oplus r)$ *and* $m\text{-}left = H\text{-}left(key_i \oplus r \oplus H(T_i \oplus r) \oplus C)$ where *m-left* is the left part of the output of the hash function H, C is a constant.
Step 3: Reader → Back-end Database(Forwarding Message): While receiving the response from the tag, the reader forwards the received authentication message r, IDS_i, *m-left, and* $H(T_i \oplus r)$ to the back-end database.
Step 4: Back-end Database → Reader(Authenticating Tag Message): After receiving the authentication message from the reader, the back-end database needs to complete the authentication and respond R, $n\text{-}right = H\text{-}right(key_i \oplus R \oplus H(T_i \oplus r))$ to the reader. If the authentication succeeds, the back-end database updates secret key.

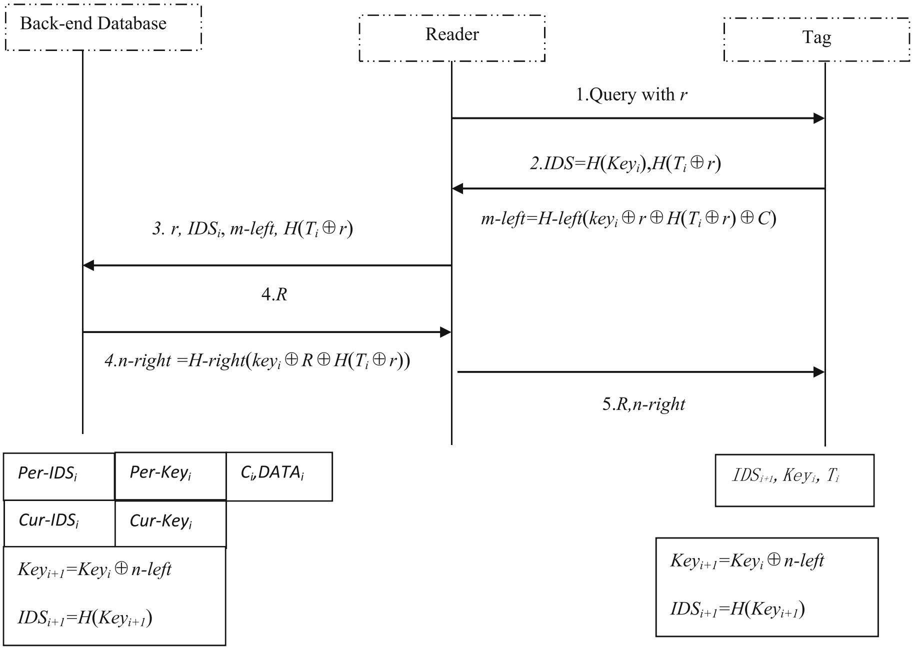

Back-end Database
Reader
Tag
1.Query with r
2.IDS=$H(Key_i)$,$H(T_i \oplus r)$
m-$left$=H-$left(key_i \oplus r \oplus H(T_i \oplus r) \oplus C)$
3. r, IDS_i, m-$left$, $H(T_i \oplus r)$
4.R
4.n-$right$ =H-$right(key_i \oplus R \oplus H(T_i \oplus r))$
5.R,n-$right$
Per-IDS_i
Per-Key_i
C_i,$DATA_i$
Cur-IDS_i
Cur-Key_i
Key_{i+1}=$Key_i \oplus n$-$left$
IDS_{i+1}=$H(Key_{i+1})$
IDS_{i+1}, Key_i, T_i
Key_{i+1}=$Key_i \oplus n$-$left$
IDS_{i+1}=$H(Key_{i+1})$

Fig. 1 The operation of the authentication protocol in Zhou et al. [10]

Step 5: Reader → Tag(Authenticating Reader Message): The reader will send R and *n-right* to the tag. While receiving the reader's authenticating messages, the tag retrieves the shared key from its local storage and calculates the local $n\text{-}right = H\text{-}right\,(key_i \oplus R \oplus H(T_i \oplus r))$. If the value of local *n-right* equals to the received one, the tag authenticates the reader successfully and updates the shared key_{i+1} to $Key_i \oplus n\text{-}left$. Otherwise, the tag will consider that the reader is invalid and will not update the shared key.

2.2 Vulnerability of the Protocol

In the operation of the protocol, it is assumed that there is a synchronized tag. We call the legal reader which controlled by the adversary as the malicious legal reader. An adversary is able to trigger a malicious legal reader which can generate a specified random number to attack the tag. The notations used in this section are listed in Table 1.

The fist step of the attack to the protocol is shown in Fig. 2. The adversary can interrupt r at step *1*, IDS_i, *m-left*, $H(T_i \oplus r)$ at step 2 and R, *n-right* at the step 5 in Fig. 1. Let *r as* r', IDS_i *as* IDS_i', *m-left as m-left′ and* $H(T_i \oplus r)$ as $H(T_i \oplus r)'$. Then the adversary holds up the messages which is sent to the tag at the step 5 in Fig. 1. Since the tag is not able to receive the messages form the reader, it will not update its variables at the last step. But the database has updated its variables as follows. (a) $IDS_1 = IDSD_{Old} = 30$. (b) $IDS_2 = IDSD_{New} = H(key_{i+1}) = H(key_i \oplus n\text{-}left) = H(key_i \oplus H\text{-}left(key_i \oplus R \oplus H(T_i \oplus r))) = 47$, while the

Table 1 Notations

Key_T	Keep the secret key of the tag	IDS_{New}	Keep the new IDS of the back-end database
IDS_T	Keep the IDS of the tag	r	A random number generated by the reader
Key_{Old}	Keep the old secret key of the back-end database	R	A random number generated by the back-end database
Key_{New}	Keep the new secret key of the back-end database	T_i	A random number generated by the tag
IDS_{Old}	Keep the old IDS of the back-end database	*m-left*	The left part of m

	INPUT	OUTPUT
TAG	$KEY_T = 20$	$KEY_T = 20$
TAG	$IDS_T = 30$	$\boldsymbol{IDS_T = 30}$
SERVER	$KEYD_{NEW} = 20$	$KEYD_{NEW} = 21$
SERVER	$IDSD_{NEW} = 30$	$\boldsymbol{IDSD_{NEW} = 47}$
SERVER	$KEYD_{OLD} = 10$	$KEYD_{OLD} = 20$
SERVER	$IDSD_{OLD} = 15$	$\boldsymbol{IDSD_{OLD} = 30}$
RANDOM	$R=5, R=6, T_I=7$	$M\text{-}LEFT=30$

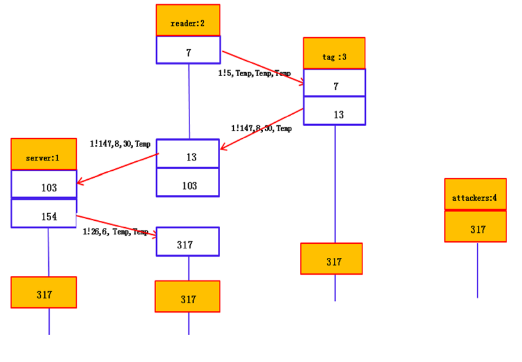


Fig. 2 The Step 1 of the attack to the protocol in Zhou et al. [10]

tag variable of IDS_T is still 30. The first step is preparing the retransmission information for the following desynchronization attack.

The second step of the attack to the protocol in Zhou et al. [10] is shown in Fig. 3. At this moment, the reader and the tag execute the authentication without any attack. Since $IDS_2 = IDSD_{New} = 47$ is not able to be found in the tag, both the database and the tag use the old secret $IDS_1 = IDS_T = 30$ as the communication secret key. Thus, the tag will update its variable list to $IDS_3 = IDSD_{New} = (key_i \oplus H\text{-}left(key_i \oplus R \oplus H(T_i \oplus r))) = 50$. In the database, the value is updated as $IDS_1 = IDSD_{Old} = 30$ and $IDS_3 = IDSD_{New} = 50$. The second step is the prerequisite of the subsequent desynchronization attack. At the third step, the desynchronization attack has been launched by using the interrupted information at the first step to break the consistency of secret key between the tag and the database.

The third step of the attack to the protocol in Zhou et al. [10] is shown in Fig. 4. The adversary is able to use a malicious legal reader to produce a random number $R_0 = r'$, where r' is the value intervened at the first step before. Then, the malicious legal reader sends the R_0 to the adversary. The adversary compounds R_0 and IDS_i', $m\text{-}left'$, $H(T_i \oplus r)'$ obtained at the first step and sends them to the malicious legal reader. After that, the adversary sends IDS_i', $H(T_i \oplus r)'$, $m\text{-}left'$ and R_0 to the back-end database by a replay attack and a spoofing attack. Then, the back-end database will authenticate the retransmission of IDS_i', $H(T_i \oplus r)'$, $m\text{-}left'$ *and* R_0 as a valid message at step 3 in Fig. 1. Then it will update its variables and

	Input	Output
Tag	$keyT = 20$	$keyT = 37$
Tag	$IDST = 30$	**$IDST = 50$**
Server	$keyD_{New} = 21$	$keyD_{New} = 37$
Server	$IDSD_{New} = 47$	**$IDSD_{New} = 50$**
Server	$keyD_{Old} = 20$	$keyD_{Old} = 20$
Server	$IDSD_{Old} = 30$	**$IDSD_{Old} = 30$**
Random	$r=8, R=9, T_i=10$	$m\text{-}left=19$

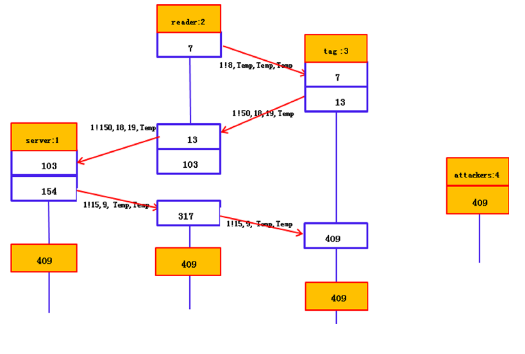


Fig. 3 The Step 2 of the attack to the protocol in Zhou et al. [10]

	Input	Output
Tag	$key_T = 20$	$key_T = 37$
Tag	$IDS_T = 30$	**$IDS_T = 50$**
Server	$keyD_{New} = 37$	$keyD_{New} = 27$
Server	$IDSD_{New} = 50$	**$IDSD_{New} = 53$**
Server	$keyD_{Old} = 20$	$keyD_{Old} = 20$
Server	$IDSD_{Old} = 30$	**$IDSD_{Old} = 30$**
Random	$r=5, R=12, T_i=13$	$m\text{-}left=24$

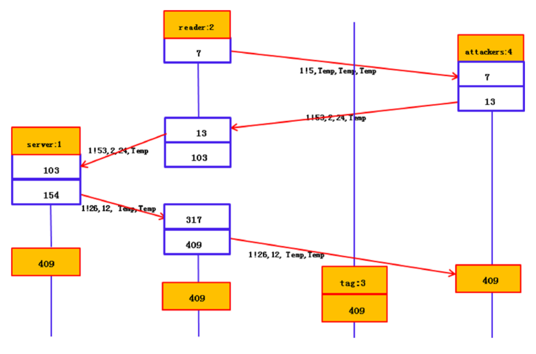


Fig. 4 The Step 3 of the attack to the protocol in Zhou et al. [10]

secret key as: (a) $IDS_1 = IDSD_{Old} = 30$, (b) $IDS_2 = IDSD_{New} = H(key_{i+1}) = H(key_i \oplus n\text{-}left) = H(key_i \oplus H\text{-}left(key_i \oplus R \oplus H(T_i \oplus R_0))) = 48$. Since $IDS_T = 50 \cap (IDSD_{New} = 48 \cup IDSD_{Old} = 30) = \varnothing$. It is clear to draw the conclusion that the desynchronization attack to the protocol in [5] is successful.

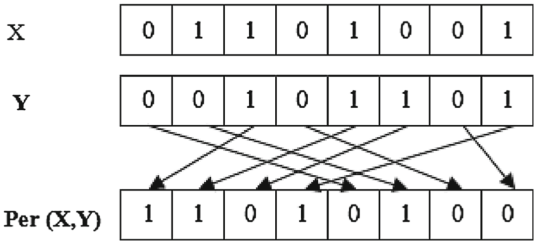
3 Proposed UP^{2}RT Scheme

Defines 1: Definition of the variable

Suppose X and Y are two l-bit strings, where $X = a_1 a_2 \ldots a_l$, $a_i \in \{0, 1\}$, $i = 1, 2, \ldots, l$, $Y = b_1 b_2 \ldots b_l$, $b_i \in \{0, 1\}$, $j = 1, 2, \ldots, l$. Moreover, the Hamming weight of B, wt(Y), is m($0 \leq m \leq l$) and $b_{k1} = b_{k2} = \cdots = b_{km} = 1$, $b_{km+1} = b_{km+2} = \cdots = b_{kl} = 0$, where $1 \leq k_1 < k_2 < \cdots < k_m \leq l$ and $1 \leq k_{m+1} < k_{m+2} < \cdots < k_l \leq l$. Then, the permutation of X according to Y, denoted as Per(X, Y), is Per(X, Y) = $a_{k1}\ a_{k2} \ldots a_{km} a_{kl} a_{kl-1} a_{km+2} a_{km+1}$. Figure 5 shows the computation of Per(X, Y).

In view of the defects of the existing protocols, we propose a low-cost RFID authentication protocol which integrates the operation of the XOR, build-in CRC-16 function, the permutation, a random tuple and secret key backup technologies to overcome the vulnerability under the desynchronization attacks without increasing the cost. The analysis shows that our proposal has a strong ability to prevent existing possible malicious attacks. The notations used are listed in Table 2. The proposed protocol is shown in Fig. 6. The detailed operation for each step is described as follows.

Step 1 The reader challenges the tag.
Step 2 While receiving the challenge, the tag responds with *TID* to the reader.
Step 3 After receiving *TID*, the reader uses it as an index to search a matched entry in the database. If it is an old *TID*, the reader will use $\{KeyH^{old}, KeyM^{old}, KeyL^{old}\}$ to compute the messages. If *TID* is new, $\{KeyH^{new}, KeyM^{new}, KeyL^{new}\}$ will be used. If *TID* is not in the database, the reader will terminate the session as this may be an invalid tag. Suppose the reader has found { *KeyH, KeyL, KeyM* } as the tag's entry. It will compute $\gamma_1, \gamma_2 \ldots \gamma_i$ and α with $R_S, R_{T1}, R_{T2}, \ldots R_{Ti}$. Then the reader send the random number R_{T1}, R_{T2}, R_{Ti} and R_S with a mask to the tag.
Step 4 The reader sends $\gamma_1, \gamma_2 \ldots \gamma_i$ and α to the tag.
Step 5 The tag extracts R_S by XOR α with CRC(Per(*keyH,keyM*)), and R_{Ti} by XOR γ_i with CRC(Per(R_{Ti-2}, R_{Ti-1})) . The tag computes the value of B with R'_S and R_T. Finally the tag computes the value of β by CRC (Per(R'_S, CRC($R_{Ti} \oplus B$))).
Step 6 The tag sends the β to the reader.



X	0	1	1	0	1	0	0	1
Y	0	0	1	0	1	1	0	1
Per (X,Y)	1	1	0	1	0	1	0	0

Fig. 5 The computation of the example

Table 2 Notation

Symbol	Meaning
$\oplus$	XOR operator
CRC-16(X)	CRC-16 is an effective checksum algorithm, in our protocol, the input X is divided into group of 16 bits, then to encrypt each group one by one and consolidate all the outputs together. For example, CRC $(\eta_1\eta_2\ldots\eta_{16}\ldots\eta_{32}) = CRC(\eta_1\eta_2\ldots\eta_{16}) \cup CRC(\eta_{17}\eta_{18}\ldots\ldots\eta_{32})$
TID	Unique serial number of tags, which tags and the back-end database share
n	n is the length of secret key
Ror_ left(x,y)	Round take y bits from x from the left to right
R_s	R_s is a random number of n bits
T	$T = \{(p_0, m_0), \ldots(p_i, m_i), \ldots(p_i, m_j), (p_{i+1}, m_{j+1})\}$, in this expression, the p_i in T(the length is $\log_2^n$ bits) means that the intercepting position is the ith bit of R_s, the m_i in T is the intercepting length, the length of m_i is 3 bits (the possible value of b_i is 000, 001, 010, 011, 100, 101, 110 or 111), $m_0 + \cdots + m_i + \cdots + m_j \leq n \leq m_0 + \cdots + m_i + \cdots + m_j + m_{j+1}$
R_T	The generating amount of random number will be control by the reader according to the need. $R_T = R_{T1} \cup R_{T2} \cup \ldots \cup R_{Ti}$, R_{Ti} is a random number of n bits, then the tag intercept m_j bits from the R_T from the starting position p_i in turn until $m_0 + \cdots + m_i + \cdots + m_j \leq n \leq m_0 + \cdots + m_i + \cdots + m_j + m_{j+1}$. Let $L = \log_2^n + 3$ and the value of $Ti = (L * j)/n$ or $Ti = (L * (j + 1))/n$. The tag intercepts the $\log_2^n$ bits from R_T as p_i and 3 bits as m_i from the R_T in turn
R_X	The generation process of R_X is the same as R_T
B	$B = \{b_1, b_2\ldots b_i\ldots b_j\ldots b_{j+1}\}$, let $B = Ror\_left(B, n)$
key^{Old}	Contains three secret key storage rooms $keyH^{Old}$, $keyM^{Old}$, $keyL^{Old}$ which are used to keep the provious secret key in the back-end database
Key^{New}	Contains three secret key storage rooms $KeyH^{New}$, $KeyM^{New}$, $KeyL^{New}$ which are used to keep the current secret key in the back-end database
key_T	Contains three secret key storage rooms $Key_T H$, $Key_T L$, Key_T M which are used to keep the tag secret key
Update	Is a secret key update function

Step 7 When receiving β, the reader will compare it with the local β to authenticate the tag. If the tag is authenticated, the reader will compute the value of $\zeta_1\ldots\zeta_i$, T', B' and δ with R''_S, $R_{x1}\ldots$ and R_{xi}.

Step 8 The reader sends $\zeta_1\ldots\zeta_i$, and δ to the tag.

Step 9 The tag extracts R_{xi} from ζ_i and computes a local value of δ. If the local value of δ equates to the received δ, the tag will authenticate the reader and update the corresponding entry. And it updates the pseudonym and the secret key.

Table 3 shows the experimental data of the authentication process of the UP^{2}RT Protocol. And the variable marked in bold is the transmitted message. The variables in the Table 3 are binary numbers with 16 bits.

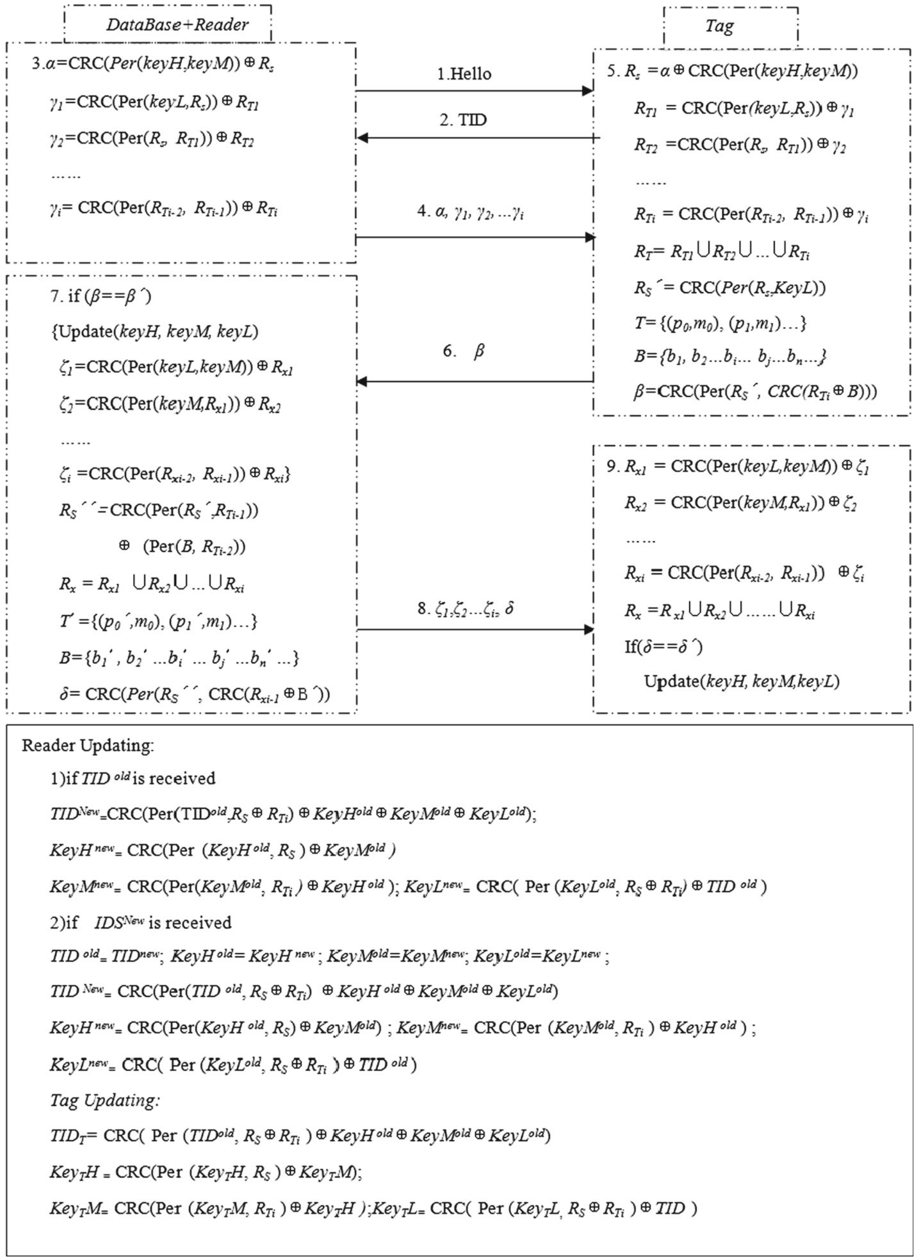

DataBase+Reader
Tag
1.Hello
2. TID
3. α=CRC($Per(keyH,keyM)$) $\oplus R_s$
γ_1=CRC(Per($keyL,R_s$)) $\oplus R_{T1}$
γ_2=CRC(Per(R_s, R_{T1})) $\oplus R_{T2}$
……
γ_i= CRC(Per(R_{Ti-2}, R_{Ti-1})) $\oplus R_{Ti}$
4. α, γ_1, γ_2, …γ_i
5. R_s = $\alpha \oplus$ CRC(Per($keyH,keyM$))
R_{T1} = CRC(Per($keyL,R_s$)) $\oplus \gamma_1$
R_{T2} =CRC(Per(R_s, R_{T1})) $\oplus \gamma_2$
……
R_{Ti} = CRC(Per(R_{Ti-2}, R_{Ti-1})) $\oplus \gamma_i$
R_T= $R_{T1} \cup R_{T2} \cup … \cup R_{Ti}$
R_S´= CRC($Per(R_s,KeyL)$)
T={(p_0,m_0), (p_1,m_1)…}
B={b_1, b_2…b_i… b_j…b_n…}
β=CRC(Per(R_S´, CRC($R_{Ti} \oplus B$)))
6. β
7. if (β==β´)
{Update($keyH$, $keyM$, $keyL$)
ζ_1=CRC(Per($keyL,keyM$)) $\oplus R_{x1}$
ζ_2=CRC(Per($keyM,R_{x1}$)) $\oplus R_{x2}$
……
ζ_i =CRC(Per(R_{xi-2}, R_{xi-1})) $\oplus R_{xi}$}
R_S´´=CRC(Per(R_S´,R_{Ti-1}))
$\oplus$ (Per(B, R_{Ti-2}))
R_x = $R_{x1} \cup R_{x2} \cup … \cup R_{xi}$
T´ ={(p_0´,m_0), (p_1´,m_1)…}
B={b_1´, b_2´ …b_i´ … b_j´ …b_n´ …}
δ= CRC(Per(R_S´´, CRC($R_{xi-1} \oplus$ B´))
8. ζ_1,ζ_2…ζ_i, δ
9. R_{x1} = CRC(Per($keyL,keyM$)) $\oplus \zeta_1$
R_{x2} = CRC(Per($keyM,R_{x1}$)) $\oplus \zeta_2$
……
R_{xi} = CRC(Per(R_{xi-2}, R_{xi-1})) $\oplus \zeta_i$
R_x =$R_{x1} \cup R_{x2} \cup …… \cup R_{xi}$
If(δ==δ´)
Update($keyH$, $keyM$,$keyL$)
Reader Updating:
1)if TID^{old} is received
TID^{New}=CRC(Per(TID^{old},$R_S \oplus R_{Ti}$) $\oplus KeyH^{old} \oplus KeyM^{old} \oplus KeyL^{old}$);
$KeyH^{new}$= CRC(Per ($KeyH^{old}$, R_S) $\oplus KeyM^{old}$)
$KeyM^{new}$= CRC(Per($KeyM^{old}$, R_{Ti}) $\oplus KeyH^{old}$); $KeyL^{new}$= CRC(Per ($KeyL^{old}$, $R_S \oplus R_{Ti}$) $\oplus TID^{old}$)
2)if IDS^{New} is received
TID^{old}= TID^{new}; $KeyH^{old}$= $KeyH^{new}$; $KeyM^{old}$=$KeyM^{new}$; $KeyL^{old}$=$KeyL^{new}$;
TID^{New}= CRC(Per(TID^{old}, $R_S \oplus R_{Ti}$) $\oplus KeyH^{old} \oplus KeyM^{old} \oplus KeyL^{old}$)
$KeyH^{new}$= CRC(Per($KeyH^{old}$, R_S) $\oplus KeyM^{old}$) ; $KeyM^{new}$= CRC(Per ($KeyM^{old}$, R_{Ti}) $\oplus KeyH^{old}$) ;
$KeyL^{new}$= CRC(Per ($KeyL^{old}$, $R_S \oplus R_{Ti}$) $\oplus TID^{old}$)
Tag Updating:
TID_T= CRC(Per (TID^{old}, $R_S \oplus R_{Ti}$) $\oplus KeyH^{old} \oplus KeyM^{old} \oplus KeyL^{old}$)
Key_TH = CRC(Per (Key_TH, R_S) $\oplus Key_TM$);
Key_TM= CRC(Per (Key_TM, R_{Ti}) $\oplus Key_TH$);Key_TL= CRC(Per (Key_TL, $R_S \oplus R_{Ti}$) $\oplus TID$)

Fig. 6 The operation of the UP^{2}RT protocol

4 Security Analysis

We analyse the security of the UP^{2}RT scheme in the ability to resist various malicious attacks. We show that the proposed UP^{2}RT scheme has the ability to prevent various existing attacks including the desynchronization attacks, tracing attacks, replay attacks, and man-in-the-middle attacks.

Table 3 Experimental data of the UP^{2}RT protocol

Steps 1–4	Steps 5–6	Steps 7–9
$TID = 46468$	$R'_S = 12404$	$R''_S = 19187$
$KeyH^{new} = 60121$	$R_T =$ 1101**0010** 000**001**11	$R_{x1} = 59684$
$KeyM^{new} = 43434$	10**101**011 0**010**0110	$R_{x2} = 62799$
$KeyL^{new}$ =43808	**110**0000**1** **01**(011011)	R_{x3} =64684
$R_S = 62927$	$T = (11, 1)(0, 1)(12, 5)(6, 2)(6, 6)(0, 5)$	$\boldsymbol{\zeta_1} = \mathbf{21877}$
$R_{T1} = 53767$	$B = 1001000000001110$	$\boldsymbol{\zeta_2} = \mathbf{39808}$
$R_{T2} = 43814$	$\boldsymbol{\beta} = \mathbf{36878}$	$\boldsymbol{\zeta_3} = \mathbf{50106}$
$R_{T3} = 49499$		$R_x = 1110100100100100$
$\boldsymbol{\gamma}$**1=5224**		11110101 0100 1111
$\boldsymbol{\gamma}$**2=32786**		11111100 10101100
$\boldsymbol{\gamma}$**3=23170**		$T' = (14, 4)(9, 1)(3, 6)(10, 4)(15, 7)$
$\boldsymbol{\alpha} = \mathbf{20706}$		B' =1110 **1101**0111 **1001**100101
		$\boldsymbol{\delta} = \mathbf{55002}$
Reader updating	Tag updating	
$TID^{New} = 6241$	TID$_T$ =6241	
$KeyH^{new} = 22310$	$Key_T H = 22310$	
$KeyM^{new} = 43184$	$Key_T M = 43184$	
$KeyL^{new} = 34288$	$Key_T L = 34288$	
$TID^{old} = 46468$		
$KeyH^{old} = 60121$		
$KeyM^{old} = 43434$		
$^{KeyL}{}_{old} = 43808$		

We introduce the function of the random triple before performing the security analysis. In the UP^{2}RT protocol, the random triple (R'_s, T, B) will be employed to resist the existing attacks. The R'_s in (R'_s, T, B) is a random number of n bits. The value of T equal to $\{(p_0, m_0), \ldots(p_i, m_i), \ldots, (p_i, m_j), (p_{i+1}, m_{j+1})\}$, where p_i with length of $log2^n$ bits is the intercepting bit of i in R'_s. The m_i in T is the intercepting length from the R'_s starting at i. If the length of m_i is 3 bits, and the possible intercepted value of R'_s is 000, 001, 010, 011, 100, 101, 110 or 111. Let $m_0 + \cdots + m_i + \cdots + m_j \leq n \leq m_0 + \cdots + m_i + \cdots + m_j + m_{j+1}, b_i = \text{Ror\_left}(R_s, p_i, m_i), B = \{b_1, b_2 \ldots b_i \ldots b_j \ldots b_{j+1}\}$ and $B = \text{Ror\_left}(B, 0, n)$. For example, if $R'_s = 10111101/10111110/11110111/11101101$, $R'_T = R_{T1} \cup R_{T2} \cup \cdots R_{Ti} =$ **00000**001/**00011**011/**00101**100/1**0010**010/**00100**011/**11001**010/ **11010**100/**10001**001 /**01000**001/**01101**011/**01010**101/**11111**011, then we can deduce that the value of T equal to { (0, 1), (3, 3), (5, 4), (20, 2), (4, 3), (25, 2), (26, 4), (17, 1), (8, 1), (13, 3), (10, 5), (31, 3)} . Further, we can obtain that the value of B equal to 1 111 1011 01 110 11 1011 1 1 110 11111 110.

4.1 Resistance to Desynchronization Attacks

If an adversary tries to attack the tag by using the retransmission desynchronization attacks, he will intercept and retransmit some messages at the Steps 4, 6, 8 in Fig. 7. Because the UP^{2}RT scheme is a mutual authentication protocol and the transmitted messages are all correlative with the random number, the method used in Sect. 2 is not able to create the desynchronization

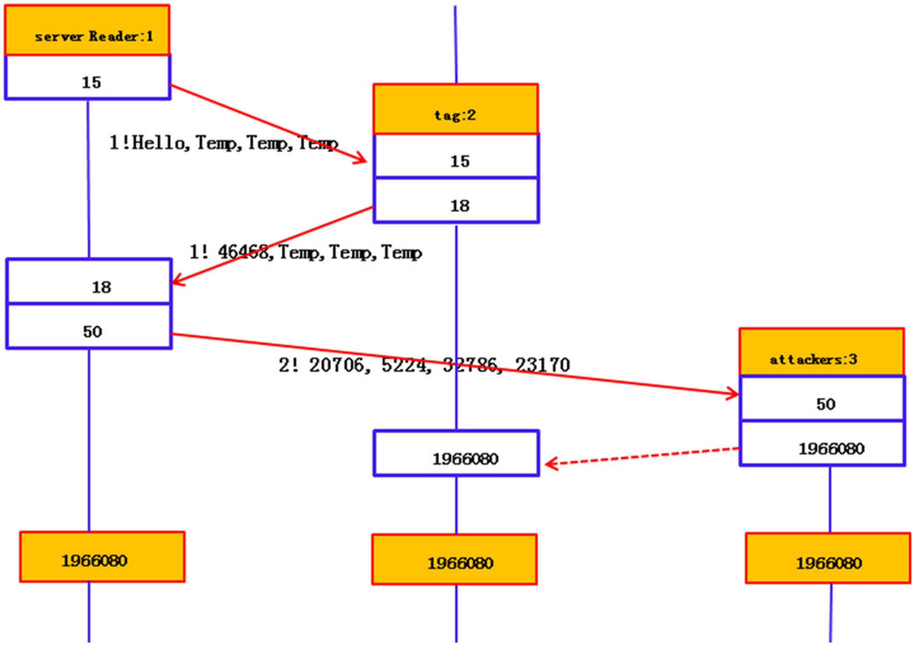


Fig. 7 Bit tampering desynchronization attacks to UP^{2}RT

between tags and readers. For example, an adversary intercepts and retransmits α, γ_1, $\gamma_2 \ldots \gamma_i$ to the tag. Then he attack UP^{2}RT scheme by the method used in Sect. 2.2. The first step and the sceond step can perform as Sect. 2.2. But the third step can not succeed, according to the retransmit α, γ_1, $\gamma_2 \ldots \gamma_i$ the tag will compute β with the old R_S, R_{Ti}, B. Then, β will be not able to pass the authentication of the reader because the new random number R_S is used to compute the local β at the reader. So we may use the secret key backup technology to complete the normal authentication in the next time. We can concluded that UP^{2}RT scheme resist the retransmission desynchronization attacks. Next we will analyze the bit tampering desynchronization attacks.

It is possible for an adversary to attack the tag by the bit tampering desynchronization attacks as Chien [6]. For example, if an adversary tries to modify R_S by flipping certain bits in α, and wants to use the forged messages to passing the subsequent authentication, the tag cannot authenticate the messages because it is very difficult for the adversary to guess a correct β by using the permutation, CRC-16, XOR and the random tuple. The analysis shows that it is not feasible to attack UP^{2}RT by using the bit tampering desynchronization attacks. Firstly, for the UP^{2}RT scheme, the permutation, XOR and the random tuple have been used to reduce the correlation of the transmitting messages. Secondly, the one-way CRC-16 function is also able to reduce the guessing possibility of the secret key greatly. The probability which the attackers can guess a forged β to pass the authentication is far smaller than $\mathrm{ADV_A} = 1/(2^{\mathrm{n1}} \cdot 2^{\mathrm{n2}} \cdot (\mathrm{n2/m_i}) \cdot 2^p \cdot \sum_{0 \le p \le i} C_n^{mi}))^{Ti}$. The length of the tuple one is n1, which is no less than the length of the secret key. The length of the tuple two is n2 and is greater than the length of the secret key, determined by the interception on the tuple three. For an easy transfer, the length of the tuple three is n, which equals to the length of the secret key. The $\mathrm{p_i}$ in the tuple two indicates that the interception position is starting from ith bit of the tuple one. The value of $\mathrm{p_i}$ could be between 1 to $\log_2^{\mathrm{n}}$. The $\mathrm{m_i}$ in the tuple two is the length of the interception, or the number of the bits to be intercepted. When the value of n is larger, the attacker is almost impossible to derive the tuple one and tuple two from the tuple three, so the protocol can resist various existing attacks well. A more detailed description of the random triples can be found in Tables 2, 3 and Sect. 4.

Therefore, we can conclude that the UP^{2}RT scheme can resist the desynchronization attacks well by using the permutation, CRC-16, XOR functions with a random tuple. The Fig. 7 shows the SPIN model of the operation of the proposed UP^{2}RT scheme under a bit tampering desynchronization attack. In the process of the attack, the attacker changes two bits of α. The value of the β will be changed to β' due to the changes of α. It is shown by the experiment result, the attack is not able to cheat the tag to get it authenticated by the proposed UP^{2}RT scheme due to its use of XOR operation, build-in CRC-16 function, permutation and the secret key backup technology. It is clear that the proposed solution has a strong ability to prevent the particular desynchronization attacks.

4.2 Man-in-the-Middle Attacks

A man-in-the-middle (MITM) attack is a form of eavesdropping where communication between two users is monitored and modified by an unauthorized party. Data confidentiality refers that the information cannot be unauthorized to use in the transmission and stored in the process. So if a protocol can assure the data confidentiality, it will be able to resist the MITM attack as well. In terms of data confidentiality, the messages $\alpha, \gamma_1, \gamma_2 \ldots \gamma_i, \beta, \zeta_1 \ldots \zeta_i$, and δ are all related to the secret key and a random number. And the messages $\alpha, \gamma_1, \gamma_2 \ldots \gamma_i, \beta, \zeta_1 \ldots \zeta_i$, and δ are encrypted by the application of the permutation, CRC-16, XOR and the random tuple. It is difficult to recover the random number without knowing the secret key. And it is impossible to guess the *KeyH*, *KeyM*, *KeyL*, R_S, R_T or R_X due to the application of the permutation, CRC, XOR and the random tuple. So the data confidentiality can be assured. In addition, the transmitted messages β and δ do not only provide the evidence for authentication of the reader, but also assure the integrity of the tag. Since our protocol can assure data confidentiality, it can resist the MITM attacks too.

4.3 Resistance Tracing Attacks Scheme Choose

A tracing attack is one of the most powerful attacks which could be issued by a "malicious active reader". The goal of the attack is to discover the presence of a specific tag. The attacker actively scans the tag from a far distance by the small device near the tag. According to the A_{query} phase (the certification stage of tag to reader), the RFID security protocols can be divided into two types. One type is the static RFID security protocol, which is the security protocol with a fixed *TID* or pseudonym. But, this type protocol cannot resist tracing attacks. Another one is the dynamic RFID security protocol, which has the authentication information changed in the A_{query} phase. The proposed protocol belongs to the second type. The value of *TID*, $\alpha, \gamma_1, \gamma_2 \ldots \gamma_i, \beta, \zeta_1 \ldots \zeta_i$, and δ will be changed according to the value of *Rs*, R_T *and* R_X after each successful authentication. In this way, the UP^{2}RT can resist the tracing attacks well.

4.4 Resistance to Replay Attacks

If there is a malicious reader attempts to retransmit *TID*, $\alpha, \gamma_1, \gamma_2 \ldots \gamma_i, \beta, \zeta_1 \ldots \zeta_i$, and δ to a tag, the tag will compute R_s, R_T by the replayed $\alpha, \gamma_1, \gamma_2 \ldots \gamma_i$ passively. Then, the tag will compute the value of β with a new random tuple and random numbers. Since the retransmitted variables use the old random numbers, it is possible to cause the authentication failure. Similarly, at each follow-up step, only one authentication on either the tag or on the reader can be successful by using the old random numbers. The authentication of both the tag and the reader cannot be successful. Therefore, the UP^{2}RT can resist the replay attacks.

5 Cost Analysis and Performance Evaluation

Figure 8 shows the logic diagram of the proposed UP^{2}RT scheme. Due to the fact that a message consists of two or more pieces, it requires one register of n bits to temporarily store intermediate results. The core component of the UP^{2}RT logic is the arithmetic logic unit (ALU). The ALU has two inputs and one control signal. One of the inputs is the data path for data to be fetched from the register, while another is the bit stream from outside. The control to the ALU is the control signal (C_1) to select the input to the ALU from either the bit stream or the data stored in the register. The control signal C_2 will determine the operation that will be performed in the ALU.

Table 4 shows the comparison of logical gates required for different length of the secret key in UP^{2}RT scheme. A hash function like MD5 generally needs 16,000 logical gates. SHA-1 needs 20,000 logical gates. The number of the logical gates required by the proposed protocol is much less than that of the protocols equipped with the complicated hash functions obviously. Therefore, the proposed UP^{2}RT scheme is suitable for the low cost RFID systems.

We analyse the performance of the proposed UP^{2}RT scheme in terms of the number of computation operations, the storage requirements and communication cost for a tag. The number of the computation operations is indicated by the number of different types of opera-

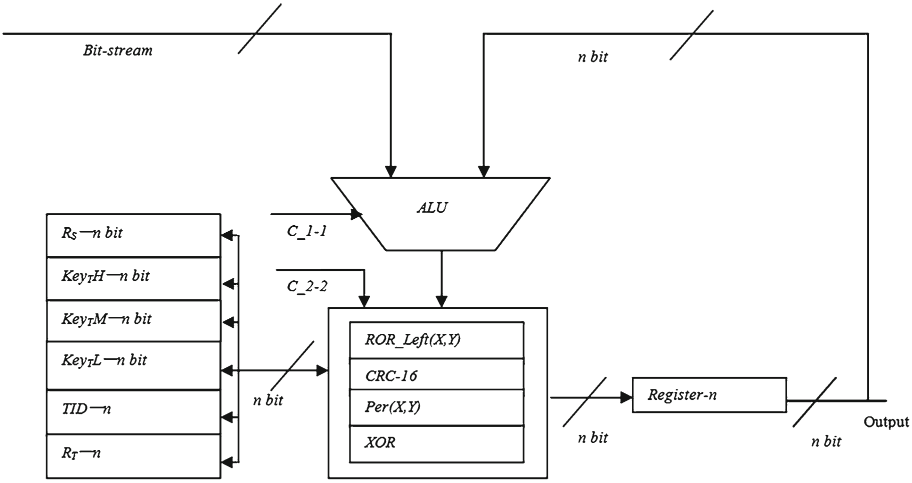


Fig. 8 Logic scheme of UP^{2}RT

Table 4 Comparison of logical gates and length of the secret key

Key length (n)	8-bit	16-bit	32-bit	64-bit	128-bit	256-bit
Gates number	51	99	195	387	771	1,539

Table 5 Comparison of logical gates and length of the secret key

	M^2AP [18]	SASI [6]	Gossamer [21]	UAPP [23]	UP^2RT
Types of computation operations	$+$, $\oplus$, AND, OR	$+$, $\oplus$, OR, Ror	$+$, $\oplus$, Ror, MixBits	$\oplus$, Ror, Per	$\oplus$, Per, CRC-16, random tuple, Ror_left
Storage requirement	6L	7L	7L	5L	6L
Communication messages	3L	2L	2L	2L	6L
Resistance to desynchronization attacks	No	No	No	No	Yes
Resistance to disclosure attacks	No	No	Yes	Yes	Yes
Resistance to tag tracking	No	No	Yes	Yes	Yes

tions required for each tag. The storage requirements are easured by the memory size required to store a dynamic tag *TID*, three shared elements and some random numbers in a tag. The communication cost is calculated by the amount of the messages sent by the tag in one execution of the protocol. The comparison results among the solution in [17], some other protocols and the proposed protocol are listed in Table 5. In Table 5, "+" denotes the addition mod 2^L. We can conclude that the cost of UP^2RT is very close to the existing ultralightweight protocols, but it has a strong ability to prevent existing possible attacks.

6 Conclusion

In this paper, we have reviewed the scheme in Zhou et al. [10] with the vulnerability exploration. It is discovered that the scheme in Zhou et al. [10] cannot resist one particular type of desynchronization attacks. In order to overcome the vulnerability, we have proposed a low-cost RFID authentication protocol which integrates the operation of the XOR, build-in CRC-16 function, permutation function, secret key backup with a random tuple to improve the security functionality without increasing much cost than the utralightweight protocols. The analysis shows that our proposal has a strong ability to prevent existing malicious attacks, especially the particular type of desynchronization attacks.

Acknowledgments This work was partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61202379 and Natural Science Foundation of Liaoning Province under Grant No. 2013024020.

References

1. Juels, A. (2006). RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Communications*, *24*(2), 381–394.
2. Sun, H. M., & Ting, W. C. (2009). A Gen2-based RFID authentication protocol for security and privacy. *IEEE Transactions on Mobile Computing*, *8*(8), 1052–1062.

3. Sarma, S. E., Weis, S. A., & Engels, D. W. (2003). Radio-frequency identification: Secure risks and challenges. *RSA Laboratories Cryptobytes*, *6*(1), 2–9.
4. Weis, S. A. (2003). *Security and privacy in radio-frequency identification devices*. Massachusetts Institute of Technology.
5. Juels, A., Rivest, R. L., & Szydlo, M. (2003). The Blocker Tag: Selective blocking of RFID tags for consumer privacy. In *Proceedings of the l0th ACM conference of computer and communications security* (pp. l03–111).
6. Chien, H. Y. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transaction of Dependable and Secure Computing*, *3*(4), 337–340.
7. Sarma, S. E., Weis, S. A., & Engels, D. W. (2003). RFID systems and security and privacy implications. In *Proceedings of the 4th international workshop on cryptographic hardware and embedded systems* (pp. 454–469).
8. Henrici, D., & Muller, P. (2004). Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In *Proceedings of 2nd IEEE annual conference on pervasive computing and communications workshops* (pp. 149–153).
9. Gao, L., Ma, M., Shu, Y., & Wei, Y. (2013). A security protocol resistant to intermittent position trace attacks and synchronization attacks in RFID systems. *Wireless Personal Communications*, *68*(4), 1943–1959.
10. Zhou, S., Zhang, Z., & Luo, Z. (2010). A lightweight anti-desynchronization RFID authentication protocol. *Information Systems Frontiers*, *12*(5), 521–528.
11. Blurn, A., Furst, M., & Keams, M. (1993). Cryptographie primitives based on hard leaming problems. *Advances in Cryptology-CRYPTO, 773*(1993), 1–10.
12. Juels, A., & Weis, S. A. (2005). Authenticating pervasive devices with human protocols. *Advances in Cryptology-CRYPTO, 3621*(2005), 293–308.
13. Bringer, J., Chabanne, H., & Dottax, E. (2006). HB++: A lightweight authentication protocol secure against some attacks. In *Proceedings of IEEE international conference on pervasive services workshop on security* (pp. 28–33).
14. Piramuthu, S. (2007). HB and related lightweight authentication protocols for secure RFID tag/reader authentication. In *Proceedings of the CollECTeR (Europe) conference* (pp. 1–8).
15. Duc, D. N., Park, J., Lee, H., & Kim, K. (2006). Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning. White Paper, pp. 1–11.
16. Doss, R., Saravanan, S., & Zhou, W. L. (2012). A practical quadratic residues based scheme for authentication and privacy in mobile RFID systems. *Ad Hoc Networks*, *11*(1), 383–396.
17. Doss, R., Zhou, W. L., Saravanan, S., Yu, S., & Gao, L. X. (2012). A minimum disclosure approach to authentication and privacy in RFID systems. *Computer Networks*, *56*(15), 3401–3416.
18. Lopez, P. P., & Castro, J. H. (2006). M^2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In *Proceedings of the international conference on ubiquitous intelligence and computing* (pp. 912–923).
19. Bárász, M., Boros, B., & Lója, P. L. K. (2007). Passive attack against the M^2AP mutual authentication protocol for RFID tags. In *Proceedings of the first international workshop on RFID technology* (pp. 1–4).
20. Sun, H. M., Ting, W. C., & Wang, K. H. (2011). On the security of Chien's ultralightweight RFID authentication protocol. *IEEE Transaction Dependable and Secure Computing*, *8*(2), 315–317.
21. Peris-Lopez, P., Hernandez-Castro, J. C., & Tapiador, J. M. E. (2009). Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In *Proceedings of the 9th international workshop on information security applications* (pp. 56–68).
22. Ahmed, E. G., Shaaban, E., & Hashem, M. (2010). Lightweight mutual authentication protocol for low cost RFID tags. *Journal of Network and Computer Applications*, *2*(2), 27–37.
23. Tian, Y., Chen, G. L., & Li, J. H. (2012). A new ultralightweight RFID authentication protocol with permutation. *IEEE Communications Letters*, *16*(5), 702–705.
24. Gao, L., Ma, M., Shu, Y., & Wei, Y. (2014). An ultralightweight RFID authentication protocol with CRC and permutation. *Journal of Network and Computer Applications*, *41*(5), 37–46.
25. Paolo, D., & Santis, A. D. (2011). On ultralightweight RFID authentication protocols. *IEEE Transactions on Dependable and Secure Computing*, *8*(4), 548–563.
26. Avoine, G., Carpent, X., & Martin, B. (2012). Privacy-friendly synchronized ultralightweight authentication protocols in the storm. *Journal of Network and Computer Applications*, *35*(2), 826–843.

Lijun Gao received his B.Sc. and M.Sc. degrees in Department of Computer Science and Technology, Shenyang Aerospace University in 2000 and 2007 respectively. He has been a lectuer at Shenyang Aerospace University since 2005. He has extensive research interests including wireless networking and wireless network security, etc. He is currently pursuing his Ph.D. at the Department of Computer Engineering, Tianjin University, doing research on the RFID security.

Maode Ma received his B.E. degree from Tsinghua University in 1982, his M.E. degree from Tianjin University in 1991 and his Ph.D. degree in computer science from Hong Kong University of Science and Technology in 1999. Now, Dr. Ma is an Associate Professor in the School of Electrical and Electronic Engineering at Nanyang Technological University in Singapore. He has extensive research interests including wireless networking and network security. He has led and/or participated in around 20 research projects funded by government, industry, military and universities in various countries. He has been a member of the technical program committees for more than 120 international conferences. He has been a general chair, technical symposium chair, tutorial chair, publication chair, publicity chair and session chair for more than 50 international conferences. Dr. Ma has more than 200 international academic publications. He currently serves as the Editor-in-Chief of *International Journal of Electronic Transport.* He also serves as a Senior Editor for *IEEE Communications Surveys and Tutorials*, and an Associate Editor for International *Journal of Network and Computer Applications,* International Journal of *Security and Communication Networks,* International Journal of *Wireless Communications and Mobile Computing* and *International Journal of Communication Systems.* He had been an Associate Editor for *IEEE Communications Letters* from 2003 to 2011. Dr. Ma is a senior member of *IEEE Communication Society and IEEE Education Society.*

Yantai Shu 1981–1984 Yantai Shu was a visiting scholar at UCLA. Yantai Shu is a professor of computer science at Tianjin University, China. He is a member of the IEEE and the ACM. He has published more than 120 papers and contributed to one book. His current interests are focused on computer communication networks, wireless networks, real-time systems, modeling and simulation.

Feng Lin Vice President of Shenyang Aerospace University, received his B.E. degree from Northeastern University in 1985, his M.E. degree from Shenyang University of Technology in 1987 and his Ph.D. degree from Shenyang University of Technology in 2003. Now, he is a professor at Shenyang Aerospace University, China. He research interests including wireless networking and network security.

Lei Zhang received her Ph.D. degree in Computer Science from Auburn University (Auburn, AL, USA) in 2008. She worked as an assistant professor from 2008–2011 in the Computer Science Dept. at Frostburg State University (Frostburg, MD, USA). She is now an assistant professor in the School of Computer Science and Technology at Tianjin University (Tianjin, P. R. China). Her research interests include computer networks, wireless communications, distributed algorithms and network security.

Yuhua Wei received her B.Sc. degrees in Department of Computer Science and Technology, Shenyang Aerospace University in 2005. She has extensive research interests including wireless networking and wireless network security, etc. She is currently pursuing her M.Sc. degrees at the Department of Computer Science and Technology, Shenyang Aerospace University, doing research on the network security.