Wireless Netw (2018) 24:69–78
https://doi.org/10.1007/s11276-016-1323-y

SLAP: Succinct and Lightweight Authentication Protocol for low-cost RFID system

Hanguang Luo[1] · Guangjun Wen[1] · Jian Su[1] · Zhong Huang[1]

Published online: 14 June 2016

Abstract Data security is crucial for a RFID system. Since the existing RFID mutual authentication protocols encounter the challenges such as security risks, poor performance, an ultra-lightweight authentication protocol named Succinct and Lightweight Authentication Protocol (SLAP) is proposed. SLAP is only composed of bitwise operations like XOR, left rotation and conversion which is easy to implement on a passive tag. The proposed conversion operation as the main security component guarantees the security of RFID system with the properties such as irreversibility, sensibility, full confusion and low complexity, which better performed or even absent in other previous protocols. Security analysis shows that SLAP guarantees the functionalities of mutual authentication as well as resistance to various attacks such as de-synchronization attack, replay attack and traceability attack, etc. Furthermore, performance evaluation also indicates that the proposed scheme outperforms the existing protocols in terms of less computation requirement and fewer communication messages during authentication process.

Keywords RFID · Security · Conversion · Ultra-lightweight protocols

1 Introduction

As a promising wireless communication technique for objects automatic identification, RFID has various merits including waterproof, antimagnetic, heat-resistance, long service life, long-distance contact, more flexible memory and will take the place of barcode in becoming a new identification scheme [1, 2]. Since the communication channel between a reader and tags is vulnerable, privacy and security is crucial for RFID system. Up to now, a number of typical attack methods have been applied on RFID system. Eavesdropping Attack is one of the most commonly used easy way to attack a RFID system, from which an adversary can easily steal the transmitted messages or even control the communication between reader and tags by modifying the obtained messages. Also, a Replay Attack can be implemented by an attacker to replay obtained messages as a valid tag (or reader) to deceive readers (or tags). Disclosure Attack can also occur. Due to the insecure wireless channel between reader and tags, segmental or whole secrets in the tags may be recovered by an attacker through the intercepted messages. Another attack carried out to track the target tag, from which the detail information was leaked, is called Malicious Traceability Attack. In order to defense the traceability attack, some secret information shared between reader and tag should be updated after each authentication. This updated process may draw out the Desynchronizing Attack which makes a de-synchronization state between the tags and the reader, so that the tags cannot be authenticated in the next round. Even worse, the data on a valid tag may be copied to embed into a fake tag by Cloning Attack.

To enhance the security of a RFID system and reduce the complexity, it is necessary to propose an efficient authentication protocol [3]. In 2007, Chien [4] roughly

✉ Hanguang Luo
luohanguang_uestc@outlook.com

[1] University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, People's Republic of China

classified the RFID authentication protocols into four types: (1) Full-fledged [5]. This class of protocol should support conventional cryptographic functions such as one-way function, symmetric encryption, even so as far as the public key algorithms. (2) Simple [6]. For this part of protocol demand to support one-way hashing function and can generate random number on tags. (3) Lightweight [7]. The third class refers to the protocols which have the ability to generate a random number on tags and has simple functions like Cyclic Redundancy Code (CRC) checksum which is more simple than one-way hash function. 4) Ultra-lightweight [8–10]. The last class of protocols can only use simple bitwise logical operations like XOR, AND, OR, etc. on tags. Moreover, even the random number generator cannot be incorporated at the tag's side. Based on the above four kinds of classification, the strong incentive to build a passive RFID tags below 5 cents and makes traditional security protocol cannot be used on it, due to the limited of resources and energy (the power supply comes from the radio waves emitted by reader). Broad consideration have been made that about 5–10 K logic gates can be used for a low-cost RFID tag and only 250–3 K can be used for security functions.

In this paper, we proposed a new ultra-lightweight mutual authentication protocol based on a security ultra-lightweight bitwise conversion. We aim to improve the poor security resistance in recent ultra-lightweight authentication protocol, which used the permutation methods to enhance the diffusion result for the exchanged messages, with some good safety features in the new conversion and to reduce the total communication messages needed for authentication especially the computational complexity on tag side. The protocol doesn't use any unbalanced operations like OR ($\vee$) and AND ($\wedge$) which has a bias output and might lead additional security vulnerabilities. Instead, we only used bitwise XOR, left rotation operations and bitwise conversion. The proposed conversion scheme has four very important properties together: irreversibility, sensibility, full confusion and low complexity, which haven't appeared in other protocols. Finally, we not only adopt the scheme that the last messages in the protocol run are sent by the tag, but also reduce the transmitted message length in each session without reducing its security. Performance evaluation shows that SLAP is more efficient,more secure and has higher speed than other ultra-lightweight protocols.

The organization of this paper is as follows: Sect. 2 discusses the related works. In Sect. 3 we analyzes in detail the vulnerability of two recently proposed protocols and present the proposed novel protocol SLAP. The security analysis is discussed in Sect. 4. Section 5 illustrates the performance of the proposed scheme and gives the properties of comparison with other protocols. Finally, conclusions are made in Sect. 6.

2 Related works

In 2006, Peris-Lopez et al. [8–10] firstly proposed a family of ultra-lightweight protocols called ultra-lightweight mutual authentication protocol (UMAP): lightweight mutual authentication protocol (LMAP), minimalist mutual-authentication protocol (M2AP) and efficient mutual authentication protocol (EMAP). All of these protocols perform simple bitwise operations, i.e. XOR, OR and addition modulo etc., and all achieve low computation cost and are efficient for low-cost passive RFID tags. However, these protocols were vulnerable to the de-synchronization attacks and full disclosure attacks [11, 12]. After that, some other attacks were published to destroy these protocols. In 2007, Chien [4] developed a new ultra-lightweight protocol named SASI (Strong Authentication and Strong Integrity) which claimed to provide a strong authentication and strong integrity. However, in the next year, Phan [13] proposed a tracking attack on SASI, and then various kinds of attacks [14–17] involving replay attack, full-disclosure attack, de-synchronization attack were pointed out, showing that SASI is not as safe as it says. In 2008, Gossamer protocol [17] was proposed by Peris-Lopez et al. which introduced a new MixBits function to enhanced diffusion properties of the transmitted messages. Soon after that, several methods [18, 19] were reported to highlight that Gossamer protocol's vulnerability of denial of service (DoS) and de-synchronization attacks. In the same year, Qingling et al. [20] protocol and LMAP++ [21] protocol were published but proved to be insecurity a few years later in [22, 23], respectively. Later then many ultra-lightweight protocols based on bitwise operations were proposed to protect the security of low-cost RFID like NRS [24], LPP [25] and DIDRFID/SIDRFID [26], etc., until in 2012, Tian et al. [27] reported a new protocol with permutation transformation. The article introduced a new primitive to make permutation methods in an ultra-lightweight protocol used to enhance the diffusion result of simple bitwise operations for the exchanged messages between reader and tags. However, the shortcoming of the permutation and the weakness of the exchanged messages lead the protocol attack to several articles [28–31]. After that, more proposed ultra-lightweight authentication protocols employed permutation method which is widely used in block symmetric cryptography to achieve fast diffusibility. Nevertheless, the realization of the diffusivity in symmetric cryptography always depends on numerous repeated round function due to the oversimplified of permutations. Hence it is inadvisable to used existing methods of symmetric cryptography's permutation straightforward in ultra-lightweight protocols.

Recently RRAP [32] presented a kind of reconstruction which claimed to be Hamming weight unpredictability, irreversibility and effectiveness. However according to our analysis, it is not the case. Given the output and one of two parameters' partial or whole bits, an attacker can probably or even surely recover the other parameter. Another recent reconstruction scheme which claimed as a recursive hash transformation was reported in [33] named RCIA. The vulnerabilities of the above two protocols are analyzed in detail in Sect. 3.

3 Vulnerability analysis of two protocol and proposed scheme

In this section, we analyzed in detail the vulnerability of two recently published protocols RRAP [32] and RCIA [33] which all used permutation methods to enhance the diffusion result of the transmitted messages, and then proposed a new ultra-lightweight authentication protocol named Succinct and Lightweight Authentication Protocol. The SLAP only adopts bitwise XOR and left rotation operations because they won't introduce extra security vulnerabilities. In order to achieve full confusion, a new conversion *Con*(A, B) was used to hide the information in the A and B.

3.1 Vulnerability analysis of two protocol

Permutation method is a lightweight linear transformation widely used in SPN block cipher designed to achieve fast diffusion characteristic. Regularly, permutation operation can be realized in various ways and as its linearity and reversibility is unsuitable to be used for ultra-lightweight protocol straightforwardly. In order to take advantage of the fast diffusion of permutation, the permutation operation used here must be changed and optimized. Zhuang et al. [32] is an ultra-lightweight protocol used a reconstruction structure which is the transformation of permutation. The main idea of the reconstruction is defined as follows:

A and B are n bits strings, where $A = a_{n-1}a_{n-2}\ldots a_1a_0$, $a_i \in \{0, 1\}$, $i = 0, 1, 2\ldots n-1$, $B = b_{n-1}b_{n-2}\ldots b_1b_0$, $b_j \in \{0, 1\}, j = 0, 1, 2\ldots n-1$. The *reconstruction* of A with B is:

$$\mathrm{Re}c(A, B) = c_{n-1}c_{n-2}\ldots c_1c_0, \quad c_i = F(a_i, b_i) \tag{1}$$

where

$$F(a_i, b_i) = \begin{cases} a_{(i-1)\bmod n}, & a_i > b_i \\ b_{(i-1)\bmod n}, & a_i < b_i \\ a_i, & a_i = b_i \end{cases} . \tag{2}$$

Without loss of generality we take it as an example that the attacker knows the parameter B defined in [32]. In order to facilitate understanding, the symbol here is as the same meaning in [32]. Through our studies, three conclusions are obtained as follows: (i) when $b_ib_{i-1} = 00$ (the adjacent two bits of B) and corresponding output bit $c_i = 1$, there must be achieved that $a_i = 1$ and $a_{i-1} = 1$; (ii) when $b_ib_{i-1} = 01$ and corresponding output bit $c_i = 1$, there must be achieved that $a_i = 1$ and $a_{i-1} = 1$; (iii) if $b_ib_{i-1} = 10$, the corresponding conclusion is $a_i = 0$ according to $c_i = 0$, otherwise $a_i = 1$. Use above three conclusions together with the rules for the reconstruction, the parameter A can be recovered or in several candidates.

When comes to the [33], which claims to introduce a recursive hash as its fast diffusion operation. In order to better analyze, we use the same symbol as in [33], i.e. A is a n bit string, and the computation result of recursive hash of A is $R_h(A)$. The main idea of the recursive hash is as follow. First, average divide a target strings A into several certain numbers of parts and the bit numbers in each substring must aliquot A. Then randomly choose one of the divided parts and XOR with the rest parts respectively, each result replace the old one. Finally, replaces the original chosen substring with its left rotating output, the rotation times are according to its Hamming weight (the Hamming-weight of rotation function is explained as follows. For example, *Rot*(A, B) represents cyclic left rotation of A according to B's Hamming weight wt(B), where wt(B) is expressed as the total numbers of "1" bit in string B). Based on the methods described above, the vulnerability analysis will be presented in the following. Taking the example in [33], the 24 bits $A = (100100)(101011)(110101)(111110)$ was divided into 4 parts and the output was $R_h(A) = (010001)_1(011110)_2(011101)_3(001011)_4$. In order to recover A, we only need to guess which one is the left rotation part (the chosen substring). The Hamming weight of each part is known, so we can guess the original bits by right rotation each of the 4 parts in certain times, respectively. Finally, the four candidates of chosen substring are $(010001)_1 \rightarrow (010100)_1$, $(011110)_2 \rightarrow (111001)_2$, $(011101)_3 \rightarrow (110101)_3$, $(001011)_4 \rightarrow (011001)_4$, then obtain the four corresponding guessed A are $A_1 = (010100)_1(001010)_2(001001)_3(011111)_4$, $A_2 = (101000)_1(111001)_2(100100)_3(110010)_4$, $A_3 = (100100)_1(101011)_2(110101)_3(111110)_4$, $A_4 = (001000)_1(000111)_2(000100)_3(011001)_4$, respectively. It can be obviously observed that A_3 is the primitive strings used in [33], fortunately, the primitive strings must be one of the guessed candidates and the guess complexity is O(n) where n is the numbers of segmentation.

3.2 Definition of conversion

Suppose A and B are both n bit strings, where

$$A = a_na_{n-1}\ldots a_2a_1, \quad a_i \in \{0, 1\}, \quad i = 1, 2\ldots n, \tag{3}$$

$$B = b_nb_{n-1}\ldots b_2b_1, \quad b_i \in \{0, 1\}, \quad i = 1, 2\ldots n. \tag{4}$$

The conversion of *Con*(A, B) consisted of three steps: grouping, rearrange and composition.

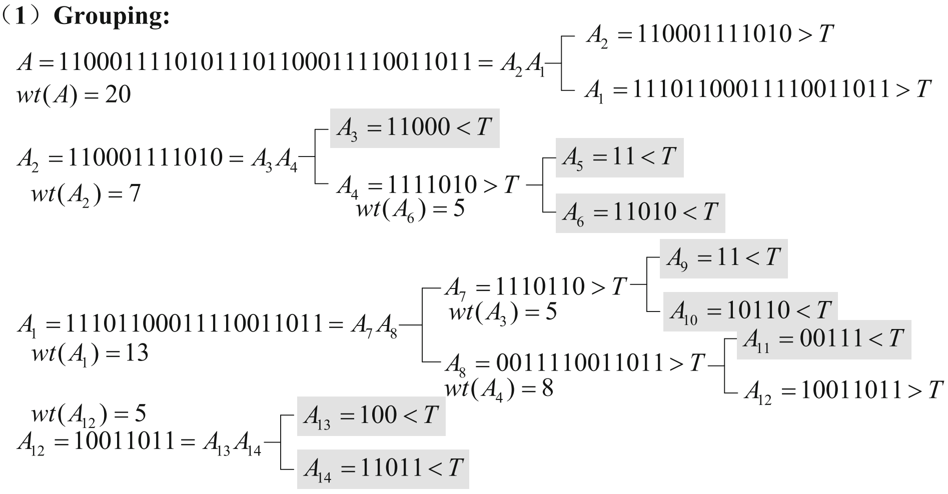
1. *Grouping* At the beginning, A and B were divided into several small blocks and the rules of the segmentation were depended on the Hamming weight of A and B. In addition, a threshold "T" must be set in advance to limit the size of each substring. For example, suppose the Hamming weight of A is m ($m \leq n$), so the division results of A are $A_2 = a_n a_{n-1} \ldots a_{m+2} a_{m+1}$ and $A_1 = a_m a_{m-1} \ldots a_2 a_1$. Then, the two substring A_1 and A_2 continue to be segmented by their Hamming weight with the same rules until all the substrings $A_j (j \leq n)$ are shorter than the threshold T. Similarly, the string B is divided into $B_k (k \leq n)$.
2. *Rearrange* After the grouping phase, the strings A and B are divided into unique substrings depending on their Hamming weight. In this phase, the rearrangement is executed with A and B. Since the length of A and B are the same, exchanging the grouping form between A and B, i.e. Regrouping string A into k groups (A_k, $k \leq n$) depends on the segmentation way with B and regrouping string B into j groups (B_j, $j \leq n$) depends on the segmentation way with A. And then, do left rotation operations with each substring in A and B with their Hamming weight, respectively. For example, the Hamming weight of A_1 is 4, then circular left rotate string A_1 by 4 bits. Finally, the results of the rearranged strings are A' and B', respectively.
3. *Composition* The last phase, take XOR operation with A' and B' which are the results of second phase. The output of XOR operation is the final result of the conversion denoted as

$$\begin{aligned} Con(A,B) = C &= c_n c_{n-1} \ldots c_2 c_1 \\ &= A' \oplus B', \quad c_i \in \{0,1\}, \quad i = 1,2\ldots n. \end{aligned} \tag{5}$$

In order to better understand, an example for $Con(A, B)$ is showed in the Example 1.

We assume A and B are 32 bits strings: $A = a_{32} a_{31} \ldots a_2 a_1 = 11000111101011101100011110011011$ (Hamming weight $wt(A) = 20$) and $B = b_{32} b_{31} \ldots b_2 b_1 = 10111101110101100011110111000010$ ($wt(B) = 19$). Threshold is $T = 6$.

(1) Grouping:



$A = A_3A_5A_6A_9A_{10}A_{11}A_{13}A_{14} = \boxed{11000}\boxed{11}\boxed{11010}\boxed{11}\boxed{10110}\boxed{00111}\boxed{100}\boxed{11011}$

$B = \boxed{1011}\boxed{110}\boxed{111010}\boxed{110}\boxed{001111}\boxed{011100}\boxed{0010}$

(2) Rearrange:

Exchanging the block form between A and B, the results are:

$A' = \boxed{1100}\boxed{011}\boxed{110101}\boxed{110}\boxed{110001}\boxed{111001}\boxed{1011}$

$B' = \boxed{10111}\boxed{10}\boxed{11101}\boxed{01}\boxed{10001}\boxed{11101}\boxed{110}\boxed{00010}$

Each substring in A' and B' does left rotation with their Hamming weight and achieve:

$A'' = \boxed{0011}\boxed{101}\boxed{011101}\boxed{011}\boxed{001110}\boxed{011110}\boxed{1101}$

$B'' = \boxed{11011}\boxed{01}\boxed{11110}\boxed{10}\boxed{00110}\boxed{11110}\boxed{011}\boxed{00100}$

(3)Composition:

$$Con(A,B) = A'' \oplus B'' = 11100001000000111110011110001001 \quad wt\left(Con\left(A,B\right)\right) = 15$$

Example 1. The conversion of $Con(A,B)$.

The proposed conversion is not only easy to realize in a low cost tag, but also has four very important properties:

- *Irreversibility* The two inputs are confused by each other not only according to their Hamming weight, but also according to every bit's position and value. Knowing one of the two inputs and the corresponding output, a malicious attacker cannot recover or predict the other input by the proposed conversion algorithm.
- *Sensibility* Even if one bit is changed in one of the two inputs, the corresponding output will be completely different. This means that the output is quite sensitive and every bit in the input will enormously influence the output.
- *Full confusion* Each of the input is confused by the other one, and has no fixed or predicted bit produced through the conversion process. So the malicious attacker cannot even fetch one bit useful information from the output to predict the input.
- *Low complexity* The conversion only used bitwise XOR and left rotation operations which are easy and costless for a passive tag to implement.

3.3 The SLAP protocol

Our protocol mainly involves three entities: tag, reader and backend server. It assumed that channel between the reader and the backend server is secure and can be seen as a whole, because usually the reader and backend server are wired connection or incorporate traditional cryptographic to protect the communications. However, due to the fact that limited resources can be used to protect the transmitted data at tag side, the wireless channel between the reader and the tag may be attacked by all kinds of existing and potential methods. Each tag has a unique static identification (*ID*), a pseudonym (*IDS*) and two secret keys (K_1, K_2), shared with backend server. Both of the reader and the tag will update the pseudonym (*IDS*) and keys (K_1, K_2) after one successful authentication process. All the strings used here are L-bit length. The specifications of the SLAP protocol are illustrated in Fig. 1, and the details of the authentication procedure are presented as follows.

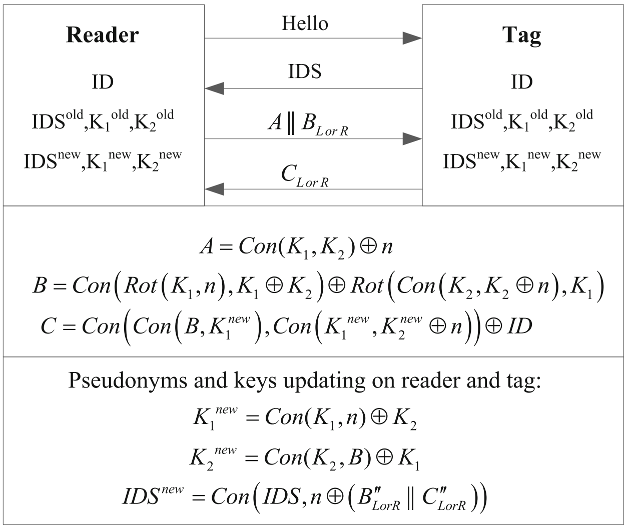

Reader
ID
$\mathrm{IDS}^{\mathrm{old}},\mathrm{K}_1{}^{\mathrm{old}},\mathrm{K}_2{}^{\mathrm{old}}$
$\mathrm{IDS}^{\mathrm{new}},\mathrm{K}_1{}^{\mathrm{new}},\mathrm{K}_2{}^{\mathrm{new}}$
Hello
IDS
$A \parallel B_{L\,or\,R}$
$C_{L\,or\,R}$
Tag
ID
$\mathrm{IDS}^{\mathrm{old}},\mathrm{K}_1{}^{\mathrm{old}},\mathrm{K}_2{}^{\mathrm{old}}$
$\mathrm{IDS}^{\mathrm{new}},\mathrm{K}_1{}^{\mathrm{new}},\mathrm{K}_2{}^{\mathrm{new}}$
$A = Con(K_1, K_2) \oplus n$
$B = Con\left(Rot\left(K_1, n\right), K_1 \oplus K_2\right) \oplus Rot\left(Con\left(K_2, K_2 \oplus n\right), K_1\right)$
$C = Con\left(Con\left(B, K_1^{new}\right), Con\left(K_1^{new}, K_2^{new} \oplus n\right)\right) \oplus ID$
Pseudonyms and keys updating on reader and tag:
$K_1{}^{new} = Con(K_1, n) \oplus K_2$
$K_2{}^{new} = Con(K_2, B) \oplus K_1$
$IDS^{new} = Con\left(IDS, n \oplus \left(B''_{LorR} \parallel C''_{LorR}\right)\right)$

Fig. 1 SLAP protocol

1. The reader (R) sends a "Hello" message to the tag (T) to initiate a protocol session.
2. After receiving the R's query, the T first responds R with its $\mathrm{IDS}^{\mathrm{new}}$, otherwise the T will responds with $\mathrm{IDS}^{\mathrm{old}}$ if no responds from the R after a transmission round or receives "Hello" again.
3. Upon receiving *IDS*, the R uses it as an index to search the corresponding entry in the database. If the *IDS* is the old one, the R will use $\mathrm{K}_1^{\mathrm{old}}/\mathrm{K}_2^{\mathrm{old}}$ to compute the transmitted data, otherwise use the new data $\mathrm{K}_1^{\mathrm{new}}/\mathrm{K}_2^{\mathrm{new}}$. If there is no matched *IDS* in the database, the R will regard the T as an invalid tag and terminate the session immediately. However, when a matched entry is found, the pseudo random number n and transmitted data A and B are computed by R, where $Rot(X, Y)$ denotes cyclic left rotation of X according to Y's Hamming weight. A is used to conceal the random nonce n with a mask and B is the authentication data used to confirm the legal status of R. But the message B will not be sent in integrated, instead, left half (B_L) or right half (B_R) of string B will be transmitted to the T depending on the Hamming weight of B (if $wt(B)$ is odd sent B_L, otherwise sent B_R).
4. After receiving the messages from R, the T extracts random number n from A by XORing with $Con(K_1, K_2)$. Next, T computes a local value of B' with its corresponding local secrets $\mathrm{K}_1/\mathrm{K}_2$, and then does the same operation in step 3 with B' to check whether $B'_{L\,or\,R}$ is equal to $B_{L\,or\,R}$ or not. If $B'_{L\,or\,R} = B_{L\,or\,R}$, the T authenticates R as a valid reader and will execute the following tasks: (1) Update the secret keys ($K_1^{new} = Con(K_1, n) \oplus K_2, K_2^{new} = Con(K_2, B) \oplus K_1$) and the pseudonym $IDS^{new} = Con\left(IDS, n \oplus \left(B''_{LorR} \parallel C''_{LorR}\right)\right)$, where B''_{LorR} and C''_{LorR} are the other half of message B and C which haven't been transmitted by R and T during step 3 and 4, respectively; (2) Calculate the message C with updated keys ($\mathrm{K}_1^{\mathrm{new}}/\mathrm{K}_2^{\mathrm{new}}$) and random number n, then sent the corresponding message $C_{L\,or\,R}$ (if $wt(C)$is odd sent C_L, otherwise sent C_R) to the R. If $B'_{L\,or\,R} \neq B_{L\,or\,R}$, the tag will terminates the authentication protocol immediately.
5. Upon receiving message from T, the reader will computes a local C' and check whether $C'_{L\,or\,R}$ is equal to $C_{L\,or\,R}$. If $C'_{L\,or\,R} = C_{L\,or\,R}$, the authentication is successful and the reader will update its pseudonym

(IDS) and keys (K_1/K_2) the same way as the tag, otherwise authentication fails and corresponding entry will not be updated.

4 Security analysis

In the following parts, we present the security analysis for SLAP, and the analyses mainly contain the functionality of the protocol and the resistance to the attacks.

4.1 Functionality of SLAP

4.1.1 Mutual authentication

As mentioned in Sect. 3, in SLAP protocol, the reader is authenticated by the tag via checking the transmitted message $B_{L or R}$ with the corresponding left or right parts of the local B', which computed with the secret keys and the pseudorandom number n extracted by message A. After the reader is authenticated successfully, the tag will calculate the message C and transmit corresponding $C_{L or R}$ to the reader for authentication. All the computed messages here are based on the shared keys K_1 and K_2, and one cannot be authenticated successfully without the pre-shared secrets. That is to say, only the valid tag and the valid reader can authenticate each other.

4.1.2 Confidentiality

The transmitted messages A, $B_{L or R}$ and $C_{L or R}$ between the reader and the tag are all related to the pre-shared secret keys. The random numbers n and the tag ID are also protected by the secret keys, with which a malicious adversary is difficult to recover them. Furthermore, in SLAP, the secret keys are concealed by the proposed conversion $Con(A, B)$ which has some fine cryptographic properties presented before. So the transmitted messages used to authenticate can only be recognized by the valid tag and the valid reader.

4.1.3 Integrity

In order to keep randomness of the communication messages, a random number should be used both in the reader side and the tag side. Since the low-cost tag is difficult in owning a random number generator, the random number can only be implemented in readers. So it is essential to ensure the received random number in the tag side is the same as the random number generated by the reader. For the sake of this goal, the messages $B_{L or R}$ and $C_{L or R}$ not only provide the evidence for mutual authentication, but also assure the integrity of the transmitted messages. For example, when a malicious adversary tries to modify n by flipping certain bits in A, it must transfer the result of random number n which extracts from A by the tag. Finally, the local computed message B' at tag side will be incorrect and lead a failed authentication. It is worth mentioning that the adversary is impossible to adjust $B_{L or R}$ to a correct value since $B = Con(Rot(K_1, n), K_1 \oplus K_2)$ - $Rot(Con(K_2, K_2 \oplus n), K_1)$ ensures that little change in n will lead a fully different in output.

4.1.4 Forward security

Due to the updating for the pseudonym IDS and the secret keys with the random number n, even if a tag is compromised some day after, the previous secrets cannot be found by the adversary. So the previous communicated information between the tag and the reader are still keeping security.

4.2 Security attacks of SLAP

4.2.1 Resistance to replay attack

Since an adversary can obtain all transmitted messages by eavesdropping the wireless channel, he can disguise himself as the tag or reader to deceive the other one. However, in SLAP, it is difficult for an adversary to forge messages as a valid tag to pass the authentication because at the initial of the protocol the reader will generated a new random number which cannot be got by the adversary (the random number is protected by $Con(K_1, K_2) \oplus n$). This makes all the replayed messages by the adversary who impersonate the tag to deceive the reader to be the illegal messages. When an adversary acts as the reader, the replayed valid reader's messages will not impact the tag since the updated secrets are the same as before and no secrets will be leaked. However, another possible replay attack scenario [13] implemented on SISA protocol, through which the adversary makes the tag and the reader desynchronized. We apply the attack on our proposed protocol as follows: (1) On the first authentication protocol, the adversary records the messages $A^1 \parallel B^1_{L or R}$ and interrupts the message $C^1_{L or R}$ at the end of the protocol (After that, secrets in the reader are $\{IDS^1, K1^1, K2^1; IDS^2, K1^2, K2^2\}$, secrets in the tag are $\{IDS^2, K1^2, K2^2; IDS^3, K1^3, K2^3\}$); (2) Running a normal protocol without intervening (Secrets in the reader are $\{IDS^2, K1^2, K2^2; IDS^4, K1^4, K2^4\}$, secrets in the tag are $\{IDS^2, K1^2, K2^2; IDS^4, K1^4, K2^4\}$); (3) At last, when the tag leaves the reading range of the reader, the adversary starts an authentication protocol with the secrets $\{IDS^2, K1^2, K2^2\}$ and replay the message

$A^1 \parallel B^1_{L\,or\,R}$ to the tag to make its secrets updated (By doing these, secrets in the reader are {IDS^2, K1^2, K2^2; IDS^4, K1^4, K2^4}, secrets in the tag are {IDS^2, K1^2, K2^2; IDS^3, K1^3, K2^3}). Obviously, through the attack experiment the tag and the reader are still synchronized. The reason why the attack in [13] is useless in SLAP is that the reader not only preserves the new updated secrets, but also keeps the old secrets.

4.2.2 Resistance to de-synchronization attack

There are two kinds of de-synchronization attacks that can be implemented on ultra-lightweight authentication protocol. First one, the adversary can try to break the data integrity in the protocol to make the tag and reader updated into different secrets. However, the data integrity of SLAP has been studied in the Sect. 4.1.3. The second one is to interrupt the corresponding transmitted message to leads failed updating in one of the two sides. Unfortunately, if the message $A \parallel B_{L\,or\,R}$ is interrupted in SLAP, the tag will abort the protocol without updating the secrets. Secondly, the situation of interrupting the message $C_{L\,or\,R}$ has been discussed in the Sect. 4.2.1. By the way, the illegal computed $C_{L\,or\,R}$ will not pass the authentication by the reader. So the SLAP ensures the synchronism for the reader and the tag.

4.2.3 Resistance to traceability attack

During each authentication instance, the tag doesn't reveal its *ID* or secrets since all the transmitted messages are competed with random number. Besides, the pseudonym *IDS* and secret keys will be updated after each successful authentication. Furthermore, none of the unbalanced operations is used in the authentication protocol which in many cases may lead to additional security vulnerabilities [34]. However, it might be the only weakness that if an adversary makes the tag unable to update its *IDS* with an unsuccessful authentication, the tag will be traceable between two successful authentications. Due to the occasion applicable of ultra-lightweight authentication protocol is to defense passive attack and costless, and the security limitation should be further improved if it is important.

4.2.4 Resistance to full disclosure attack

By definition, the simple T-function (XOR, AND and OR) is impossible to make all output bits depend on all input bits, the reasonable way to overcome the shortcoming is by influencing one output bit with as many inputs bits as possible. In SLAP, the proposed string conversion scheme $Con(A, B)$ largely improved the cryptography features of the output with its several important properties. On the contrary, it is difficultly to disclose one of the two secrets (K_1 and K_2) even if the adversary obtains $Con(K_1, K_2)$. Moreover, the messages transformed in SLAP are hidden with at least two pre-shared secrets and combining with other operations. Thus, it is impossible for an adversary to achieve any secrets by any existing method, such as, slightly modifying the transmitted messages to find suitable approximation of internal secrets.

5 Performance evaluation

In order to obtain sufficient security, our scheme not only enhances the confidentiality of each input message, but also controls the connection between each transmitted message. I.e. On the one hand, each transmitted message is fully confused by our *conversion* (the output of each bit depends on as much input bits as possible, the more information input bits are based on, the more security an output bit is obtained). On the other hand, each transmitted message should be irrelevant as far as possible on external performance, so that the adversary cannot execute an attack through relevancy between each message on some attack methods.

In this section, the performance analysis of the proposed protocol is presented. In SLAP, tags involve three operations: XOR, Hamming weight based left rotation (Rot) and *conversion* ($Con(A, B)$). The first two operations are all low-cost and easy to be implemented in a passive tag. The conversion $Con(A, B)$ is primarily composed of three bitwise operations: grouping, Hamming-weight based left rotation and XOR. The implementation of $Con(A, B)$ is briefly described as follows:

1. Grouping string A and B depend on the rules described in Sect. 3.1, respectively. Then copy the grouping pattern of A into the third memory C.
2. Regrouping string A depends on the grouping pattern of B, and rearranging it according to the rearrange rules. Then does the same operation with the string B according to the string C.
3. XORing string A with B and preserving the output.

Obviously, these operations are all extremely lightweight and can easily be implemented on a low-cost tag. And the computational complexity depends on the threshold T. The smaller threshold T is set, the greater degree of confusion is made. However, it is worth noting that grouping is not as small as possible, because the grouping with short substring may be easier to produce fixed bits after Hamming weight based left rotation and will lead secure vulnerability. Here we suggest setting the threshold T larger than 5.

Table 1 Performance comparison of some ultra-lightweight authentication protocols

	LMAP [8]	EMAP [10]	SASI [4]	GOASSMER [17]	RAPP [27]	RRAP [32]	RCIA [33]	Ours
Memory requirement on tag	6L	6L	7L	7L	5L	5L	7L	7L
Total communication messages for authentication	6L	7L	6L	6L	7L	7L	6L	4L
Communication messages generated on tag	2L	3L	2L	3L	2L	2L	2L	1.5L
Resistance to de-synchronization attacks	No	No	No	No	No	Yes	Yes	Yes
Resistance to disclosure attacks	No	No	No	Yes	No	Yes	Yes	Yes
Resistance to tag tracking	No	No	No	No	No	Yes	Yes	Yes
Security of the diffusion function	–	–	–	Yes	No	No	No	Yes

We compare the performance and robust security of SLAP with some other ultra-lightweight authentication protocols shown in Table 1, and assume all strings used here have the length of L. Here we only concern about the tag side of the protocol because the hardware resources of the reader and the back-end database are sufficient in the general case. In SLAP, storage requirement with each tag are 7L bits, which is used to preserve its L bits of *ID* and 6L bits of pseudonyms and secret keys. The total communication messages that transmitted in one protocol run are 4L bits and only 1.5L bits of which are sent by the tag. These effective properties not only reduce the power consumption in authentication communication, but also improve the authentication speed compared with other protocols. Improving the authentication efficiency does not mean that sacrificed its security, instead, the robust security provided by SLAP is better than any other protocols demonstrated in Table 1. Especially, the security of the diffusion function in SLAP which guaranteed the major security characteristic in an ultra-lightweight authentication protocol proved to be more secure than the two other recently proposed protocols RRAP and RCIA.

6 Conclusion

To enhance the security of the RFID system, many researches have been made in low-cost protocols and structures as the limited resources with electronic tags. Unfortunately, it is quite usual that the time for publication of a serious attack on a new scheme is extremely short. This is a good phenomenon and at the same time also gives us some admonishment. We notice more researchers have taken part in RFID system security research, at the same time, some common faults must be avoided when we are in designing. For example, don't use a biased output operation in the exchanged messages used for authentication from which an attacker can easily impersonate it with a high probability or rely too much on some security proofs like BAN logic which not only has its limitation in most formal models but also not very relevant for our purposes.

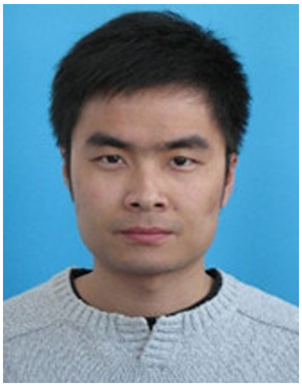
In this paper, we analyzed the vulnerability of two diffusion function employed in recently published protocols and have proposed a novel ultra-lightweight mutual authentication protocol with a new conversion. In SLAP, only three bitwise operations were employed for the tag: XOR, left rotation and conversion. The proposed conversion has four very important properties: irreversibility, sensibility, full confusion and low complexity, which as the main security component in the protocol is more secure than any other previous proposed scheme by comparison. At last, security analysis shows that SLAP can resist various existing attack, moreover, communication cost is reduced. The analyses and prominent features show that the SLAP is more preferable for a low-cost RFID system.

References

1. Collotta, M., Pau, G., & Tirrito, S. (2015). A preliminary study to increase baggage tracking by using a RFID solution. In *Proceedings of the international conference on numerical analysis and applied mathematics 2014 (ICNAAM-2014)* (Vol. 1648). AIP Publishing.
2. Chung, C., Hsieh, Y., Wang, Y., & Chang, C. (2016). Aware and smart member card: RFID and license plate recognition systems integrated applications at parking guidance in shopping mall. In *2016 Eighth international conference on advanced computational intelligence (ICACI), Chiang Mai, Thailand* (pp. 253–256).
3. Shen, J., et al. (2016). A practical RFID grouping authentication protocol in multiple-tag arrangement with adequate security assurance. In *2016 18th international conference on advanced communication technology (ICACT)*. IEEE.
4. Chien, H.-Y. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transactions on Dependable and Secure Computing, 4*(4), 337–340.
5. Juels, A., Molnar, D., & Wagner, D. (2005). Security and privacy issues in E-passports. In *SecureComm 2005. First international*

conference on security and privacy for emerging areas in communications networks, 2005. IEEE.

6. Chien, H.-Y. (2006). Secure access control schemes for RFID systems with anonymity. In *Proceedings of 2006 international workshop future mobile and ubiquitous information technologies (FMUIT'06)*.
7. Bringer, J., Chabanne, H., & Dottax, E. (2006). HB++: A lightweight authentication protocol secure against some attacks. In *Proceedings of IEEE international conference pervasive service, workshop security, privacy and trust in pervasive and ubiquitous computing*.
8. Peris-Lopez, P., Hernandez-Castro, J. C., Estévez-Tapiador, J. M., et al. (2006). LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In *Workshop on RFID security* (pp. 12–14).
9. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2006). M^{2}AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In J. Ma, H. Jin, L. T. Yang & J. J.-P. Tsai (Eds.), *Ubiquitous intelligence and computing* (pp. 912–923). Berlin: Springer.
10. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., et al. (2006). EMAP: An efficient mutual-authentication protocol for low-cost RFID tags. In *On the move to meaningful internet systems 2006: Otm 2006 workshops* (pp. 352–361). Berlin: Springer.
11. Li, T., & Wang, G. (2007). Security analysis of two ultra-lightweight RFID authentication protocols. In H. Venter, M. Eloff, L. Labuschagne, J. Eloff & R. von Solms (Eds.), *New approaches for security, privacy and trust in complex environments* (pp. 109–120). Berlin: Springer.
12. Li, T., & Deng, R. (2007). Vulnerability analysis of EMAP—An efficient RFID mutual authentication protocol. In *ARES 2007. The second international conference on availability, reliability and security, 2007* (pp. 238–245). IEEE.
13. Phan, R. C. W. (2009). Cryptanalysis of a new ultralightweight RFID authentication protocol—SASI. *IEEE Transactions on Dependable and Secure Computing, 6*(4), 316–320.
14. Avoine, G., Carpent, X., & Martin, B. (2010). Strong authentication and strong integrity (SASI) is not that strong. In S. B. O. Yalcin (Ed.), *Radio frequency identification: Security and privacy issues* (pp. 50–64). Berlin: Springer.
15. 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.
16. Sun, H.-M., Ting, W.-C., & Wang, K.-H. (2009). On the security of Chien's ultralightweight RFID authentication protocol. *IEEE Transactions on Dependable and Secure Computing, 2*, 315–317.
17. Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. (2009). Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In K.-I. Chung, K. Sohn & M. Yung (Eds.), *Information security applications* (pp. 56–68). Berlin: Springer.
18. Bilal, Z., Masood, A., & Kausar, F. (2009). Security analysis of ultra-lightweight cryptographic protocol for low-cost RFID tags: Gossamer protocol. In *NBIS'09. International conference on network-based information systems, 2009*. IEEE.
19. Tagra, D., Rahman, M., & Sampalli, S. (2010). Technique for preventing DoS attacks on RFID systems. In *2010 International conference on software, telecommunications and computer networks (SoftCOM)*. IEEE.
20. Qingling, C., Yiju, Z., & Yonghua, W. (2008). A minimalist mutual authentication protocol for RFID system & BAN logic analysis. In *CCCM'08. ISECS international colloquium on computing, communication, control, and management, 2008* (Vol. 2). IEEE.
21. Li, T. (2008). Employing lightweight primitives on low-cost rfid tags for authentication. In *Vehicular technology conference, 2008. VTC 2008-Fall. IEEE 68th*. IEEE.
22. Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., Li, T., & van der Lubbe, J. C. A. (2010). Weaknesses in two recent lightweight RFID authentication protocols. In F. Bao, M. Yung, D. Lin & J. Jing (Eds.), *Information security and cryptology* (pp. 383–392). Berlin: Springer.
23. Safkhani, M., et al. (2011). Security analysis of LMAP++, an RFID authentication protocol. In *2011 International conference for internet technology and secured transactions (ICITST)*. IEEE.
24. Fernando, H., & Abawajy, J. (2011). Mutual authentication protocol for networked RFID systems. In *2011 IEEE 10th international conference on trust, security and privacy in computing and communications (TrustCom)*. IEEE.
25. Fan, X., et al. (2011). A lightweight privacy-preserving mutual authentication protocol for RFID systems. In *GLOBECOM workshops (GC Wkshps), 2011 IEEE*. IEEE.
26. Lee, Y.-C. (2012). Two ultralightweight authentication protocols for low-cost RFID tags. *Applied Mathematics and Information Sciences, 6*(2S), 425–431.
27. Tian, Y., Chen, G., & Li, J. (2012). A new ultralightweight RFID authentication protocol with permutation. *Communications Letters, IEEE, 16*(5), 702–705.
28. Avoine, G., & Carpent, X. (2013). Yet another ultralightweight authentication protocol that is broken. In J.-H. Hoepman & I. Verbauwhede (Eds.), *Radio frequency identification. Security and privacy issues* (pp. 20–30). Berlin: Springer.
29. Zhuang, X., et al. (2013). Security analysis of a new ultra-lightweight RFID protocol and its improvement. *Journal of Information Hiding and Multimedia Signal Processing, 4*(3), 166–177.
30. Ahmadian, Z., Salmasizadeh, M., & Aref, M. R. (2013). Desynchronization attack on RAPP ultralightweight authentication protocol. *Information Processing Letters, 113*(7), 205–209.
31. Shao-hui, W., Zhijie, H., Sujuan, L., & Dan-wei, C. (2012). *Security analysis of RAPP an RFID authentication protocol based on permutation*. Cryptology ePrint Archive, Report 2012/327.
32. Zhuang, X., Zhu, Y., & Chang, C.-C. (2014). A new ultralightweight RFID protocol for low-cost tags: RRAP. *Wireless Personal Communications, 79*(3), 1787–1802.
33. Mujahid, U., Najam-ul-Islam, M., & Ali Shami, M. (2015). RCIA: A new ultralightweight RFID authentication protocol using recursive hash. *International Journal of Distributed Sensor Networks*. doi:10.1155/2015/642180.
34. Avoine, G., Carpent, X., & Hernandez-Castro, J. (2015). Pitfalls in ultralightweight authentication protocol designs. *IEEE Transactions on Mobile Computing*. doi:10.1109/TMC.2015.2492553.



Hanguang Luo received his B.S. degree in Communications Engineering, and M.S. degree in Signal Processing from Guilin University of Electronic Technology, Guilin, China, in 2010 and 2013, respectively. He is currently a Ph.D. candidate in School of Communication and Information Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include Optical Communication, Radio Frequency Identification (RFID) technology, Wireless Sensors Networks (WSN), and Cryptography.

Guangjun Wen (M'04–SM'10 of IEEE society) received his B.S. degree in Applied Physics and M.E. degree in Optic Engineering from Chongqing University in China in 1986 and 1992, respectively, and received his Ph.D. degree in electronic engineering from University of Electronic Science and Technology of China (UESTC) in 1998. He is currently a Professor at school of Communication and Information Engineering in UESTC. His research and industrial experience covers a broad spectrum of electromagnetics including RFIC/MMIC/MMMIC/ASIC/SoC design for Wireless Communication, Navigation, Identification and Mobile TV applications, ''The Internet of things'' devices and system, RFID system and networks, antennas, as well as model of electromagnetic metamaterial and its application in microwave engineering area. He has authored and coauthored approximately 300 papers and 3 books in Radio frequency Integrated Circuits and RFID Systems.

Jian Su received his B.S. and M.S. degrees in Electrical Engineering from Central China Normal University in 2008 and 2012, respectively. He is now with a Ph.D. candidate in electrical and computer engineering from University of Electronic Science and Technology of China (UESTC). His research interests include Radio Frequency Identification (RFID) technology, Wireless Sensors Networks (WSN), Cognitive Radio Networks, and Image Processing.

Zhong Huang received B.S. and M.S. degrees from Henan University, Kaifeng, China, in 2011 and 2014, respectively. He is currently a research assistant working toward a Ph.D. degree at the School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests lie in dynamic spectrum access and cognitive radio-based networks, vehicular ad hoc networks, and wireless networks security.