Wireless Pers Commun (2014) 79:1787–1802
DOI 10.1007/s11277-014-1958-x

A New Ultralightweight RFID Protocol for Low-Cost Tags: R^{2}AP

Xu Zhuang · Yan Zhu · Chin-Chen Chang

Published online: 26 July 2014

Abstract Several ultralightweight radio frequency identification (RFID) authentication protocols have been proposed in recent years. However, all of these protocols are reported later that they are vulnerable to various kinds of attacks (such as replay attack, de-synchronization attack, full disclosure attack, *etc*.) and/or have user privacy concerns. In this paper, we propose a new ultralightweight RFID protocol named reconstruction based RFID authentication protocol (R^{2}AP), which is based on the use of a new bitwise operation *reconstruction*. Operation reconstruction has three important properties: Hamming weight unpredictability, irreversibility and effectiveness. Some or all of these properties are absent in previous protocols and therefore has caused a lot of insecurity issues. The proposed R^{2}AP takes advantage of reconstruction to guarantee security of RFID system. Furthermore, we improve the Juels–Weis untraceability model so that the extended mathematic model can be used to analyze security functionality for ultralightweight RFID protocols. Our security analysis and performance evaluations demonstrate that (1) R^{2}AP can withstand all attacks mentioned in the paper and protect users' privacy; (2) R^{2}AP is indeed an effective RFID protocol that can be implemented on low-cost tags.

Keywords Ultralightweight · RFID protocol · Hamming weight · Low-cost tag · Wireless authentication

X. Zhuang · Y. Zhu (✉)
School of Information Science and Technology, Southwest Jiaotong University,
Chengdu 610031, Sichuan, China
e-mail: yzhu@swjtu.edu.cn

X. Zhuang
e-mail: zhuangxusc@gmail.com

C.-C. Chang
Department of Information Engineering and Computer Science, Feng Chia University,
Taichung City 40724, Taiwan

C.-C. Chang
Department of Computer Science and Information Engineering, Asia University, Taichung 41354, Taiwan
e-mail: alan3c@gmail.com

1 Introduction

Nowadays radio frequency identification (RFID) systems have been developed in many fields, including accesses control system, supply chain management system, payment [4] and e-passport management. As the widespread use of RFID technique, more and more security and privacy concerns of RFID systems are exposed, such as the de-synchronization attack, full disclosure attack, tags' traceability, *etc*.

In a typical RFID system, three parties are needed: tags, readers and a back-end database. A tag communicates with a reader via a wireless channel which can be eavesdropped by a malicious attacker, while a reader generally utilizes the wire channel to interact with the back-end database. Due to the inherent insecure issues occurred in wireless channel, an attacker can easily steal the transmitted messages and control the communication between a tag and a reader by modifying obtained messages. On the basis of information gained from intercepted messages, the attacker can try to fully compromise the secrets in the tag (full disclosure attack), to break the data synchronization between the tag and reader (de-synchronization attack), to track the victim tag (traceability attack), to replay obtained messages to deceive readers or tags (replay attack), *etc*. Therefore, more and more researchers began to focus on improving the security and user privacy for RFID system. Among all methods, RFID protocol is the primary choice for system designers because of its easy for implementation and relative low cost.

Tags used in ultralightweight RFID protocols generally are passive, which means that these low-cost tags cannot send radio signals actively and have extremely limited recourses (250–3 K logic gates) can be used in logic computations [22]. Obviously, it is not applicable to perform traditional encryption algorithm on these tags because of their extremely limited computation power. So the concept of *ultralightweight* RFID protocol is proposed recently. Compared with previous RFID protocols which are called *simple* and *lightweight*, the ultralightweight protocols aim to remove costly operations (*ex*. random number generation and hash function) from the tag side and only adopt some very simple operations (*ex*. bitwise exclusive or (XOR) and cycle rotation) to guarantee the security and user privacy for RFID systems.

In most previous ultralightweight RFID protocols, T-functions[1] Klimov and Shamir [14] play a very important role in the encryption processing. In 2013, Zahra et al. [2] presented the recursive linear and differential cryptanalysis for ultralightweight RFID protocols where T-functions are exclusively used. Their research demonstrates the poor encryption property of T-functions, and hence most T-functions based ultralightweight RFID protocols are not secure. On the other hand, operations that are not T-functions like Hamming weight based rotation [16] and permutation [29] are adopted in some protocols [16,29]. For these protocols, their security functionality mainly depend on the security property of operations rotation and permutation. However, both the rotation and permutation output binary strings with the same Hamming weight as its first parameter. This weakness (Hamming weight invariant) of rotation and permutation leads series of attacks on these protocols [5,31,32].

To overcome drawbacks of operations used in previous ultralightweight RFID protocols, we develop a new bitwise operation *reconstruction* in this paper. Reconstruction has two parameters and output a binary string with the same length as its parameters. There are three very important merits of the new operation reconstruction: (1) Hamming weight unpredictability; (2)irreversibility and (3) effectiveness. Exploiting these properties of reconstruction, we

[1] T-function (Triangular-function) refers to function that the ith bit of its output only depends on bits $0, \ldots, i$t of its input(s), where the Least Significant Bit is indexed by 0. Obviously, operations exclusive or (XOR), and (AND), addition are T-functions. For formal definition of T-function, please refer to [14].

propose a new ultralightweight RFID authentication protocol named R^{2}AP (Reconstruction based RFID authentication protocol). Furthermore, we extend the Juels–Weis untraceability model [13] so that it can be used to analyze security functionality of ultralightweight RFID protocol. Our security analysis and performance evaluations demonstrate that R^{2}AP is indeed an effective and secure ultralightweight RFID protocol.

Generally, as the readers connect with back-end database via wired channel, it is possible for them to use classic encryption methods to ensure the channel's security. However, due to the wireless communications, the forward channel (reader to tag) and backward channel (tag to reader) are quite insecure because any attacker can eavesdrop them. In the paper, we focus on the security of the forward channel and the backward channel and we always assume that the channel between any reader and the back-end database is secure. Thus the concept "reader" and "back-end database" may not be rigidly distinguished because they can be regarded as a single entity in all protocols mentioned in the rest of the paper.

The remainder of the paper is organized as follows. Section 2 describes our related work. The proposed protocol R^{2}AP is presented in Sect. 3. Section 4 firstly presents the extended security model for ultralightweight RFID protocol and then gives our security analysis and performance evaluations for R^{2}AP. Finally, conclusions are drawn in Sect. 5.

2 Related Work

In 2006, Lopez et al. [21–23] proposed a family of ultralightweight RFID protocols named UMAP (Ultralightweight Mutual Authentication Protocol): LMAP [22] (Lightweight Mutual Authentication Protocol), M^{2}AP [23] (Minimalist Mutual Authentication Protocol) and EMAP [21] (Efficient Mutual Authentication Protocol). These three protocols only implement some very simple bitwise operations on tags, including exclusive or (XOR), or (OR) and addition modulo n (+), where n is the length of binary string used in protocols. These schemes are quite efficient for low-cost RFID tags due to their low computation cost and saving storage space in the tag side. However, literatures [3,6,17,18] revealed many possible malicious attacks against these three protocols. Li and Wang [17] presented two de-synchronization attacks and a full disclosure attack on LMAP (these attacks also can be applied to M^{2}AP). Their active attacks are based on utilizing the weakness of LMAP for cutting off potential relation between some messages (such as the relation between message C and D). In addition, the authors [17] proposed two mechanisms (sending $\overline{D}$ and storing status) to protect the protocol from their attacks. However, sending $\overline{D}$ is not helpful since an attacker still can know whether his attack is successful or not by checking the tag's next *IDS* (new or old). Furthermore due to the inherent issue in communication between two entries, there must be one entry cannot ensure the other's state, which means that one side (reader/tag) cannot know the state of the other side (tag/reader) in RFID system. Thus, it is still a problem for storing status mechanism to guarantee the integrity of status bit in both sides. Later, [3,6,18] presented several approaches to fully compromise RFID tags in LMAP, M^{2}AP and EMAP.

In 2007, Chien [9] developed the SASI protocol having the merit of providing strong authentication and strong integrity as claimed. Unfortunately, [8,26,27] pointed out that SASI is vulnerable to various kinds of attacks, namely DoS[2] (denial of service) attack, de-synchronization attack, anonymity tracingattack, replay attack and full-disclosure attack.

[2] Actually, we can classify DoS attack into two groups: one is caused by the instinctive issues in wireless communications, named *hard-DoS*; the other one is caused by the de-synchronization between a tag and the back-end database, named *soft-DOS*. In this paper, we only consider the later form of DoS: soft-DoS.

Later, Gossamer protocol [24], David-Prasad protocol [10] and Lee-Hsieh-You-Chen [15] protocol were proved to be failing due to their security and privacy vulnerabilities as reported in [20,25,28].

In 2012, Tian et al. [29] proposed a quite interesting ultralightweight RFID authentication protocol (named RAPP) with a new bitwise operation *permutation*. RAPP utilizes operation permutation to conceal the potential relation between different messages. However since the Hamming weight of the output of permutation is the same as its first parameter's, Avoine et al. [5] described a traceability attack on it. Based on this traceability and two properties of permutation reported in [32], Zhuang et al. [32] presented a replay attack to break the synchronization of RAPP and a de-synchronization attack using the main idea shown in [1]. In [30], Wang et al. proposed an approach to fully compromise secrets in a RAPP tag, but it is not applicable since mass of protocol sessions are needed.

In 2013, Jeon and Yoon [12] developed a new ultralightweight RFID protocol named RAPLT (RFID Authentication Protocol for Low-cost Tags) based on two new operations, *Merge* (*Mer*) and *Separation* (*Sep*). However, Zhuang *et al.* [31] found that RAPLT is not secure in the sense of resisting de-synchronization attack and replay attack as well as protecting data integrity and user privacy.

3 Proposed Scheme

Notations used in the paper are shown in Table 1.

Definition of reconstruction:

A and B are two l-bits strings, where $A = a_{l-1}, a_{l-2}, \ldots, a_0,\ a_i \in \{0,1\},\ i = 0, 1, \ldots l-1,\ B = b_{l-1}b_{l-2}\ldots b_0,\ b_j \in \{0,1\},\ j = 0, 1, \ldots l-1$. The *reconstruction* of A with B is:

$$Rec(A, B) = c_{l-1}c_{l-2}\ldots c_0,\ c_i = F(a_i, b_i), \tag{1}$$

where

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

Figure 1 gives an example of this new bitwise operation.

Operation reconstruction uses two strings as its input and output a new string. Reconstruction has three very important merits:

A. Hamming weight unpredictability. Each bit in the output of reconstruction is from one of the two parameters. Therefore, the Hamming weight of output neither equals to the first parameter's nor the second parameter's. So the Hamming weight of output of recon-

Table 1 Notations used in the paper

Notations	
$\oplus$	Bitwise XOR operation
$wt(x)$	The Hamming weight of string x
$Rot(x, y)$	Circular left rotate string x by $wt(y)$ bit(s)
$Rec(x, y)$	The reconstruction operation of x with y
$[S]_i$	The ith bit of string S. LSB (Least Significant Bit) is indexed by 0

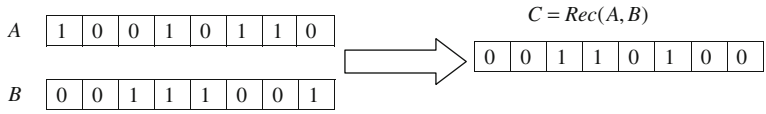

Fig. 1 Computation example of the new operation reconstruction

struction is unpredictable. Furthermore, this property complex the relation between the output and parameters.

B. Irreversibility. Given the output and any one of two parameters, the attacker cannot compute the other parameter using the rules for reconstruction. Thus, reconstruction is irreversible.

C. Effectiveness. The implementation of reconstruction only needs $2l$ comparison operations and one traversal for each parameter, where l is length of strings. The time complexity of reconstruction has the same complexity level with exclusive or XOR. We will explain the reason in detail in Sect. 4.2.

Based on the new bitwise operation reconstruction, we propose a new ultralightweight RFID authentication protocol named R^{2}AP (Reconstruction based RFID Authentication Protocol) without considering the security issues of communications between readers and the backend database.

To protect the anonymity of tags, we adopt the index-pseudonym (*IDS*) mechanism. *IDS* is an index of a table where a tag's secrets are stored. Based on the*IDS*, the backend server can locate the memory storing all secrets of a given tag at the time complexity level O (1). At the end of each successful protocol run, *IDS* and all secret keys will be updated by using two random numbers.

3.1 The R^{2}AP protocol

Supposition of our model: a tag T and a reader R (backend database) share *IDS*, *ID* and three secret keys, denoted as K_1, $K_2$and K_3, and all strings used in our protocol have length l.

There are seven steps of R^{2}AP:

Step 1: The reader R sends a hello message to tag T to initialize a new protocol session.

Step 2: The tag T responds R with its *IDS*.

Step 3: Upon receiving T's *IDS*, R uses it as an index to search T's secrets in the backend database. If R finds the record, Step 4 will be carried out; otherwise, R terminates the ongoing protocol session.

Step 4: R generates a random number n_1, and then transmits messages A and B to the tag T, where

$$A = Rec(K_1, K_2) \oplus n_1, \tag{3}$$

and

$$B = Rot(Rec(K_2, n_1), Rec(K_3, n_1)) \oplus Rot(n_1, n_1). \tag{4}$$

Step 5: After receiving messages $A$and B, the tag T extracts random number n_1 from message A and then computes message B' by using Formula (4) with its local secrets K_1, K_2, K_3 and the extracted random number n_1. If $B = B'$, the tag T authenticates R as a valid reader and then transmits message C as a response, where:

$$C = Rec(Rec(K_2, K_3), Rec(n_1, K_1)) \oplus ID. \tag{5}$$

Otherwise, T terminates the protocol round.

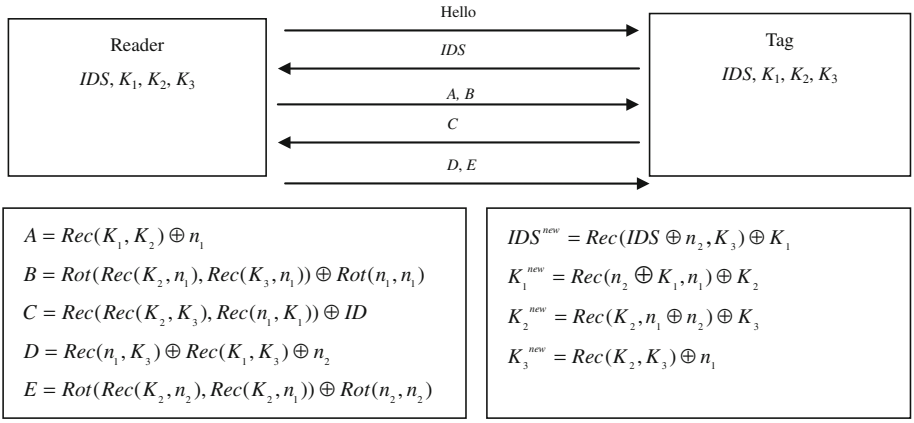

Fig. 2 R^{2}AP protocol

Step 6: Obtaining message C, the reader R can authenticate T using the ID extracted from message C. If the ID matches the one in the backend database, R generates a random number n_2 and computes messages $D$and E as follows and transmits them to tag T:

$$D = Rec(n_1, K_3) \oplus Rec(K_1, K_3) \oplus n_2, \tag{6}$$

$$E = Rot(Rec(K_2, n_2), Rec(K_2, n_1)) \oplus Rot(n_2, n_2). \tag{7}$$

And then, R will update its secrets as follows:

$$IDS^{new} = Rec(IDS \oplus n_2, K_3) \oplus K_1, \tag{8}$$

$$K_1^{new} = Rec(n_2, n_1) \oplus K_2, \tag{9}$$

$$K_2^{new} = Rec(K_2, n_1 \oplus n_2) \oplus K_3, \tag{10}$$

$$K_3^{new} = Rec(K_2, K_3) \oplus n_1. \tag{11}$$

Step7: Upon receiving messages D and E, T extracts random number n_2 from message D and tests the validity of message E using its local secrets. If T accepts E, T updates its secrets and IDS as the same way as R does. Otherwise, T does nothing.

All procedures of our protocol are shown in Fig. 2.

Note that, since the updating of the reader side is activated by the transition of messages D and E, the reader cannot know whether the tag gets these two messages or not. This problem would be utilized by a malicious attacker to lead the de-synchronization attack on our protocol. To overcome such a problem, protocols [9–11,25,29] adopt the secret redundancy mechanism on one side (tag or reader) or both sides. However, as analyzed in [19], redundancy mechanism is still useless for withstanding the de-synchronization attack. Therefore we adopt a new method (denial of the old IDS) described in [32] to keep the synchronization state between the reader and the tag, and we introduce it briefly here.

A reader not only shares the ID, IDS, K_1, K_2 and K_3 with a tag, but also needs to store old secret keys and random numbers used in last protocol session. All information stored in the reader for a tag can be denoted as ID, IDS, K_1, K_2, K_3, IDS^{old}, K_1^{old}, K_2^{old}, K_3^{old}, n_1^{old}, n_2^{old}. Once receives an old IDS, the reader should use its corresponding old secret keys and random numbers to generate messages A and B which are as the same as the old ones used

in last session, because they use the same random numbers. Then, the tag computes message C, which is also the same as the old one, and responds it to the reader. Note that, in a normal protocol session, message C is used to authenticate the tag by a reader. Thus the reader should give the "service" to the tag if message C is accepted. However, due to the potential security threats caused by the loss of messages D and E in last protocol session, the reader should take action to synchronize the tag as soon as possible rather than offer it service, although the reader has received a correct message C. For this reason, the reader should reject the service request but still uses the old random numbers and old secrets to generate messages D, E and then sends them to the tag. If the tag accepts D and E, it would update its secrets while reader needn't to do so because of the same random numbers used. After that, the tag can be authenticated again by any reader. In such case, the reader is to keep synchronization between the tag and itself rather than offer the tag a normal protocol session. So for any old *IDS*, the reader will reject the service request immediately and carry out protocol session which is used to synchronize the tag as described above. For convenience, in the rest of the paper, we use the term "synchronization session" to represent the protocol session which is initialized by an old *IDS* in Step 2.

4 Security Analysis and Performance Evaluation

In this section, we present the security analysis and performance evaluations for R^{2}AP. We firstly improve the untraceability model [26] so that we can use the extended model to analyze security functionality of the proposed R^{2}AP.

4.1 Security and Privacy Model for Ultralightweight RFID Protocol

A protocol party $\mathcal{P}$ is a tag $\mathcal{T} \in Tags$ or reader $\mathcal{R} \in readers$ in protocol sessions. An adversary $\mathcal{A}$ can control the communications between all protocol parties by active or passive manners. Following five queries are used to model $\mathcal{A}$'s abilities:

A. **Execute**($\mathcal{R}$, $\mathcal{T}$, i) query. This query models $\mathcal{A}$'s passive behavior eavesdropping the communication channel between $\mathcal{R}$ and $\mathcal{T}$ in the ith genuine protocol session. Using this query, $\mathcal{A}$ can obtain all transmitted messages in the ith protocol session between $\mathcal{R}$ and $\mathcal{T}$.
B. **Send**($\mathcal{P}_1$, $\mathcal{P}_2$, i, m) query. This query models the $\mathcal{A}$'s active behavior that $\mathcal{A}$ can impersonate a certain protocol party $\mathcal{P}_1$ to send message m to party $\mathcal{P}_2$ in the ith genuine protocol session. Note that, $\mathcal{P}_1$ and $\mathcal{P}_2$ must be opposite parties in this query.
C. **Corrupt**($\mathcal{T}$, K') query. This query models the action that $\mathcal{A}$ first obtain the secret key K in a tag $\mathcal{T}$ and then set K to K'. This model is based on the assumption that RFID tags generally are not tamper-resistant.
D. **Test**(i, $\mathcal{T}_0$, $\mathcal{T}_1$) query. This query is used to define the untraceability (UNT) instead of any specific ability of $\mathcal{A}$. In the ith session of a protocol, a Test(i, $\mathcal{T}_0$, $\mathcal{T}_1$) query gives $\mathcal{A}$ ID_b from the set $\{ID_0, ID_1\}$ referring to tags $\{\mathcal{T}_0, \mathcal{T}_1\}$, where $b \in \{0, 1\}$. $\mathcal{A}$ wins the test if he can guess the bit b.
E. **Intercept**($\mathcal{P}_1$, $\mathcal{P}_2$, i, m) query. This query models $\mathcal{A}$'s active behavior that $\mathcal{A}$ can intercept the message m transmitted from $\mathcal{P}_1$ to $\mathcal{P}_2$ in the ith genuine protocol session. Using this query, $\mathcal{P}_2$ cannot receive m sent from $\mathcal{P}_1$.

4.2 Security Analysis

In this subsection, we discuss the security functionality of R^2AP on the basis of the above mentioned model. In the following discussions, each security functionality will be defined as a game $\mathcal{G}$ having only one player, an adversary. If the adversary wins a game under a set of rules, R^2AP is not secure in terms of the security functionality defined by the game. Otherwise, the protocol achieves the security functionality.

4.2.1 Tag Anonymity and Untraceability

Phan [26] defined the untraceability (UNT) of RFID protocol by using the game $\mathcal{G}$ played between an adversary $\mathcal{A}$ and a collection of protocol parties. The game $\mathcal{G}$ has following three phases:

Phase 1 (Learning). The adversary $\mathcal{A}$ can send any Execute, Send and Corrupt queries.

Phase 2 (Challenge). During $\mathcal{G}$, two fresh tags $\mathcal{T}_0$ and $\mathcal{T}_1$ (corresponding to identifiers ID_0 and ID_1) are given to $\mathcal{A}$ to be tested. And then $\mathcal{A}$ sends a Test query based on ID_0 and ID_1. After that, $\mathcal{A}$ is given a challenger identifier ID_b from the set $\{ID_0, ID_1\}$, where b is randomly chosen from $\{0, 1\}$. Then, A can again send any Execute, Send and Corrupt queries with restriction that $\mathcal{T}_0$ and $\mathcal{T}_1$ are not issued any corrupt query.

Phase 3 (Guessing). Finally, A outputs a guessed bit b' of the value of b.

The result of the $\mathcal{G}$ is denoted as:

$$\begin{aligned} \mathrm{Adv}_{\mathcal{A}}^{\mathrm{UNT}}(k) &= |\Pr[\mathcal{A} \text{ wins}] - \Pr[\text{random coin flip}]| \\ &= \left|\Pr[b' = b] - \frac{1}{2}\right| \end{aligned} \tag{12}$$

The protocol achieves tag anonymity and untraceability if $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{UNT}}(k) < \varepsilon(K)$, where $\varepsilon(K)$ is a negligible function. In such case, the adversary cannot win the game.

To break tag anonymity and untraceability of RFID protocol, an adversary can first send several queries to learn relation among the tag's *ID* and communication messages. If the adversary gets a probabilistic equation in the form of

$$[ID]_i = [M_1]_i \bullet [M_2]_i \ldots \bullet [M_n]_i \text{ with probability} p \tag{13}$$

where M_j is a communication message, $\bullet$ denotes arbitrary operator and p is larger than 1/2, the adversary wins the game. We explain the reason next.

We assume that the adversary gets an equation satisfying Eq. (13) with probability $p > 1/2$ in the Learning Phase of the game. Given two fresh tags $\mathcal{T}_0$, $\mathcal{T}_1$ and a random challenger identifier ID_b from the set $\{ID_0, ID_1\}$, the adversary first send Execute queries to get all communication messages of T_0 in a protocol session and then compute $[ID]_i'$ using Eq. (13) with these messages. If $[ID]_i'$ equals $[ID_b]_i$, the adversary outputs $b = 0$; otherwise the adversary outputs $b = 1$. In such case, $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{UNT}}(k) = |\Pr[\mathcal{A} \text{ wins}] - \Pr[\text{random coin flip}]| = \left|p - \frac{1}{2}\right| > \varepsilon(K)$ because of $p > 1/2$.

The fact that an adversary can establish such an probabilistic equation with probability larger than 1/2 is because unbalance operator(s) used in the protocol. Unbalance operator refers bitwise operator that the probabilities of outputting 0 and 1 are different. For example, operator OR is an unbalance operator because the probability of outputting 1 is 3/4 while 0 is 1/4.

In order to establish equation in the form of Eq. (13) of R^2AP, we get

$$[ID]_i = [Rec(Rec(K_2, K_3), Rec(n_1, K_1))]_i \oplus [C]_i \bullet [M_1]_i \bullet [M_2]_i \ldots \bullet [M_n]_i. \tag{14}$$

The adversary wins the game if he can rewrite Eq. (14) as

$$\begin{aligned}[ID]_i &= [Rec(Rec(K_2, K_3), Rec(n_1, K_1))]_i \oplus [C]_i \bullet [M_1]_i \bullet [M_2]_i \ldots \bullet [M_n]_i \\ &= [M_1']_i \bullet [M_2']_i \ldots \bullet [M_n']_i \text{ with probability } P > \frac{1}{2}. \end{aligned} \tag{15}$$

In Eq. (14), the item $[Rec(Rec(K_2, K_3), Rec(n_1, K_1))]_i$ cannot be merged into another message since $Rec(Rec(K_2, K_3), Rec(n_1, K_1))$ only occurs in message C and all operators used in R^{2}AP are balance operators (from the truth value table, it is obvious that reconstruction is a balance operator). Thus, the adversary cannot find ways to identify the challenger identifier and we have $\text{Adv}_{\mathcal{A}}^{\text{UNT}}(k) = |\Pr[\mathcal{A} \text{ wins}] - \Pr[\text{random coin flip}]| = \left|\frac{1}{2} - \frac{1}{2}\right| = 0 < \varepsilon(K)$. Therefore R^{2}AP achieves tag anonymity and untraceability in terms of the attack form described in [26].

4.2.2 Data Integrity

Note that for any security system, it is always possible (with probability larger than 0) for an adversary to guess (although this probability may be very low) a secret because the length of a secret is limited. Therefore, we focus on discussing the data integrity functionality of a RFID protocol in views of whether the protocol can protect the data integrity under a reasonable number of attacks launched by an adversary.

Most RFID protocols use random numbers to keep randomness of communication messages. Because tags in ultralightweight RFID protocol are low-cost, random number generator can be only implemented in readers. Therefore, the first task of the reader in a protocol session is to transmit the generated random number to the tag. Thus, the data integrity of a RFID protocol mainly refers to the data integrity of the random number. Namely, the received random number in the tag side must be ensured to be the same as the random number generated by the reader. So we can simplify the definition of data integrity in an ultralightweight RFID protocol by only considering the data integrity of the random numbers.

We define the ability of an ultralightweight RFID protocol to keep data integrity using the game $\mathcal{G}$ played between an adversary $\mathcal{A}$, a tag $\mathcal{T}$ and a reader $\mathcal{R}$. The game $\mathcal{G}$ has following three phases:

Phase 1 (Intercepting): The adversary $\mathcal{A}$ sends Intercept($\mathcal{R}, \mathcal{T}, i, m$) query that m has the form of $m = m_r \oplus n$, where $n$denotes the generated random number.

Phase 2 (Modifying): The adversary $\mathcal{A}$ sends Send($\mathcal{R}, \mathcal{T}, i, m$') query where m' is computed by flipping some bits of m and we denote it as a function $m' = flip(m, k_1, k_2, \ldots k_n)$ where $k_1, k_2, \ldots, k_n$refer the indexes of bits needed to be flipped. In this phase, the adversary $\mathcal{A}$ tries to replace the message m by m' so that $\mathcal{T}$ will receive a modified n' and $n' = flip(n, k_1, k_2, \ldots, k_n)$.

Phase 3 (Challenge): The adversary $\mathcal{A}$ tries to forge all the remaining messages and send Send query for each forged message to continue the ongoing protocol session. $\mathcal{A}$ can repeat (assuming N times) these three phases until he finds a successive protocol session, which means that $\mathcal{A}$ has forged all the remaining messages to lead a successive protocol session in the case that $\mathcal{T}$ received a modified n'. Thus, $\mathcal{A}$ breaks the data integrity of the random number n with N times attempts since different random numbers are used in the tag $\mathcal{T}(n)'$) and the reader (n).

Finally, The game $\mathcal{G}$ outputs N denoting the average number of probing times needed. The adversary $\mathcal{A}$ wins $\mathcal{G}$ if

$$N < \alpha \times f(L), \tag{16}$$

Table 2 Security factors for some RFID protocols

Protocol	Security factor α	Attack form
LMAP [22], M^{2}AP [23]	1/48	[17]
EMAP [23]	1/72	[18]
SASI [9]	4	[7]
RAPP [29]	2	Break n_2 using method shown in [32]
RAPLT	1/12	[31]
R^{2}AP	768	–

Where α is a security factor, and L is the length of strings in protocol and $f(L)$ is a function with respect to L. A larger security factor α means that larger N is needed for an adversary to break the data integrity of the protocol. Thus the larger α used in a protocol, the more secure data integrity functionality the protocol achieves.

For convenience, we assume that $f(L) = L$. In most cases, the length of string in ultra-lightweight RFID system is 96 ($L = 96$). In such case, we computes the maximum security factor which should be used in some protocols so that these protocols can guarantee their data integrity. The results are shown in Table 2.

In R^{2}AP, it is relative easier for an adversary to try to break the data integrity of the random number n_2. The adversary can flip two neighboring bits of n_2 (the modified n_2 is denoted as n_2'), denoted as $[n_2]_i$ and $[n_2]_{i-1}$, respectively. It is obvious that the probability is 1/2 for $wt(n_2) = wt(n_2')$. Thus, two neighboring bits of message E will be flipped if the part $Rot(Rec(K_2, n_2), Rec(K_2, n_1))$ of E is unchanged. In such case, the adversary can correctly guess these two bits with probability 1/96. However, the part $Rot(Rec(K_2, n_2), Rec(n_1, n_2))$ will be changed at the same time due to n_2. On the basis of the rules for reconstruction, three bits of $Rec(K_2, n_2)$may be changed. Unfortunately, the adversary cannot know which bits are changed after$Rot(Rec(K_2, n_2), Rec(n_1, n_2))$ is performed and he still does not know how these three bits would be changed. The probability that the adversary can correctly guess which and how these bits will be changed is $1/(96 \times 8)$. Totally, we have that the adversary can try to break the data integrity of n_2 in R^{2}AP using the above way with probability 1/73728. Referring to (16) we get that the security factor of R^{2}AP is 768 which is a huge progress compared with other protocols mentioned in Table 2.

4.2.3 Resistance to Replay Attack

In a session of RFID protocol between a tag $\mathcal{T}$ and a reader $\mathcal{R}$, an adversary $\mathcal{A}$ can obtain all communication messages (some of the communication messages may be intercepted by the adversary) by eavesdropping the channel. In a later protocol session between $\mathcal{T}$ and $\mathcal{R}$, if $\mathcal{A}$ can lead the protocol session to be successful by only replaying the obtained messages in previous sessions, $\mathcal{A}$ implements the replay attack.

Typically, the adversary can try to impersonate the tag to deceive the reader for service. However, it is very difficult to forge messages that will be accepted by the reader because the adversary cannot get the random number generated by the reader easily. Therefore, most attackers try to impersonate the reader to deceive the tag. Generally, such an attack can lead the de-synchronization state between the tag and reader. We only discuss this form (deceive the tag) of replay attack in this paper.

The functionality of a ultralightweight RFID protocol to withstand replay attack is modeled by a game $\mathcal{G}$ played between an adversary $\mathcal{A}$, a tag $\mathcal{T}$ and a reader $\mathcal{R}$. The game $\mathcal{G}$ has following two phases:

Phase 1 (Obtaining): The adversary $\mathcal{A}$ can send any Execute, Intercept queries. Each obtained message is denoted as M_i, where $M \in \{A, B, C, D, E\}$ in R^{2}AP and i is the session number. The set M_{all} is used to represent all obtained messages.

Phase 2 (Challenge): The adversary $\mathcal{A}$ chooses a session number i and he can send any **Send**$(\mathcal{P}_1, \mathcal{P}_2, i, m)$ query where $m \in M_{all}$.

The adversary $\mathcal{A}$ wins the game $\mathcal{G}$ if he can find a subset M_{sub} of M_{all} which leads a successful protocol round in the ith session between $\mathcal{A}$ and $\mathcal{T}$.

In many ultralightweight RFID protocols, redundancy mechanism are used to improve their security functionalities. Indeed, replay attacks are generally derived from the defective redundancy mechanism. In R^{2}AP, the redundancy mechanism is implemented in the reader side. Last and next secret keys are both stored in the reader. In such case, a generic replay attack model can be described as follows:

Phase 1 (Setting): A tag $\mathcal{T}$ and a reader $\mathcal{R}$ share next session *ID* sid_i (In R^{2}AP, *IDS* can be considered as session *ID*), secret keys K_i. In addition, $\mathcal{R}$ stores information (sid_{i-1}, K_{i-1}) used in last protocol session. The communication messages are summarized as four messages: message sid_i (tag to reader) is used to start a new protocol session; challenge message c_i (reader to tag); authentication message a_i (tag to reader) and final message f_i (reader to tag).

Phase 2 (Intercepting 1): In the ith genuine protocol session between $\mathcal{T}$ and $\mathcal{R}$, the adversary $\mathcal{A}$ sends Intercept($\mathcal{R}, \mathcal{T}, i, f_i$) query and notes down sid_i, c_i, a_i and f_i. Consequently, the reader updates its next (last) information as $sid_{i+1}(sid_i)$ and$k_{i+1}(k_i)$ while the tag remains sid_i and k_i.

Phase 3 (Intercepting 2): In the $i+1$th genuine protocol session between $\mathcal{T}$ and $\mathcal{R}$, the adversary $\mathcal{A}$ sends Intercept($\mathcal{R}, \mathcal{T}, i+1, f_{i+1}$) query. Thus, the reader updates its next(last) information as $sid_{i+2}(sid_i)$ and $k_{i+2}(k_i)$ while the tag remains sid_i and k_i.

Phase 4 (Replaying): In the $i+2$th protocol session of $\mathcal{T}$, the adversary $\mathcal{A}$ impersonates $\mathcal{R}$ to invoke a new protocol session. After receiving sid_i from $\mathcal{T}$, $\mathcal{A}$ sends Send($\mathcal{R}, \mathcal{T}, i+2, c_i$) query. $\mathcal{T}$ will response $\mathcal{A}$ with a_i. Last, $\mathcal{A}$ sends Send($\mathcal{R},\mathcal{T}, i+2, f_i$) query.

After successfully performing above four phases, the tag will update its secrets as sid_{i+1} and k_{i+1}, which are different with secrets stored in the reader. Obviously, $\mathcal{A}$ wins the game because $\mathcal{A}$ has found a subset $\{c_i, f_i\}$ of $\{sid_i, c_i, a_i, f_i, sid_{i+1}, c_{i+1}, a_{i+1}, f_{i+1}\}$ which could lead a successful protocol round and break the de-synchronization state between the reader and tag.

Next we analyze the ability of R^{2}AP to resist the replay attack described by the model. In R^{2}AP, an adversary can execute the first two phases of the replay attack model. Once $\mathcal{R}$ receives the old session *ID* sid_i in phase 3, $\mathcal{R}$ will response $\mathcal{T}$ with message c_i instead of c_{i+1} due to the same random numbers are used. Therefore, $\mathcal{A}$ cannot manipulate the communication between $\mathcal{T}$ and $\mathcal{R}$ so that $\mathcal{R}$ will update its information to sid_{i+2} (sid_i) and k_{i+2} (k_i) as $\mathcal{A}$ expected. So we can conclude that $\mathcal{A}$ cannot execute Phase 3 of the replay attack model successfully, which means that $\mathcal{A}$ cannot win the game in R^{2}AP.

From the above discussion, we have that R^{2}AP has the ability to withstand replay attack.

4.2.4 Resistance to De-synchronization Attack

Typically, two cases of de-synchronization attacks should be considered.

First, an adversary can try to break the data integrity of random numbers used in the protocol. Thus, the tag and reader will update their secrets by using different random numbers. Consequently, the tag cannot be authenticated by reader later since different keys are stored in them. This issue is closely related to data integrity which we have studied in the Sect. 4.2.2.

The other case is typically caused by the loss of final messages (messages D and E) in the final step, which leads to an unsuccessful updating for one of these two sides. In R^{2}AP, the side which may falls into such a trap and does not update its secrets is the tag side, because the final messages are transmitted from the reader. However, as our analysis made in *Resistance to replay attack*, such uncertain state in the tag side would not cause any insecure concerns because of the denial of the old *IDS* mechanism adopted in R^{2}AP.

4.2.5 Mutual Authentication

R^{2}AP is a mutual authentication protocol, which means that both the reader and the tag sides need to authenticate each other for guaranteeing the validity of opposite side. Based on messages A, B, D and E, a valid tag can authenticate a valid reader because of the shared secret keys. For the same reason, a legitimate reader can authenticate a tag with message C.

4.2.6 Resistance to Disclosure Attack

Besides *IDS*, any secret used in R^{2}AP is hidden with at least two other secrets. Thus, it is impossible for an adversary to get any secret from any single session directly. Moreover, the rules of our new operation reconstruction can increase the complexity for an attacker to compromise all secrets in a tag.

4.2.7 Forward Security

If a tag is compromised by an attacker, the attacker can get all secrets used for mutual authentication in the next session. However, because of the updating for all secrets and the *IDS* are based on two random numbers, an attacker cannot find any clue for previous secrets used at all.

4.3 Performance Evaluation

In R^{2}AP, tags implement three operations: exclusive or (XOR), Hamming weight based rotation (*Rot*) and reconstruction (*Rec*). XOR and *Rot* can be implemented on low-cost tags because they are very efficient bitwise operations. Actually, the time complexity of implementation of reconstruction has the same complexity level with the implementation of XOR. In the implementation of *Rec*, the basic operation is bits comparison. We briefly describe implementation procedures of *Rec*:

Step 1: Two pointers P_1 and P_2 are used to index the MSBs of the first and second parameters of *Rec*, respectively. Another pointer P_r refers to the MSB of the output string.

Step 2: The values (denoted as V_1 and V_2) pointed by P_1 and P_2 are compared. If $V_1 = V_2$, V_1 is used to set the value pointed by P_r and then P_1, P_2 and P_r move to next position. If $V_1 \neq V_2$ and the larger one is represented by $V_L (L \in \{1, 2\})$, P_1 and P_2 move to next position and the value pointed by P_L is copied to the value pointed by P_r. Last, P_r goes to next position. This step is repeated until the LSBs of parameters are processed.

Obviously the above algorithm scans each string (two parameters and output) once, which is same with the bitwise operation XOR. We also have pointed out that the basic operation

Table 3 Comparisons among all protocols motioned in the paper

	LMAP [22]	M^{2}AP [23] EMAP [21]	RAPLT [12]	SASI [9]	David-Prasad [10]	Gossamer [24]	Lee et al. [15]	RAPP [29]	ours
Resistance to full-disclosure attack	No	No	Yes	No	No	No	No	Yes	Yes
Resistance to man-in-the-middle attack	–	–	No	–	–	–	–	No	Yes
Resistance to de-synchronization attack	–	–	No	–	–	–	–	No	Yes
Resistance to replay attack	–	–	No	–	–	–	–	No	Yes
Resistance to trace attack	–	–	No	–	–	–	–	No	Yes
Mutual authentication	–	–	No	–	–	–	–	No	Yes
Forward security	No	No	Yes	No	No	No	No	Yes	Yes
Total no. of communications in a single session	4	4	4	4	5	4	4	5	5
Total messages for authentication	6	7	7	6	7	6	5	7	7
Memory size on tag	$6L$	$6L$	$5L$	$7L$	$5L$	$4L$	$3L$	$5L$	$5L$
Memory size for each tag on backend database	$6L$	$6L$	$5L$	$4L$	$4L$	$4L$	$3L$	$9L$	$11L$

of *Rec* is bits comparison, which is exactly the basic operation of XOR. Therefore, we can conclude that *Rec* has the same time complexity with XOR. Thus, reconstruction is indeed an efficient operation.

In R^2AP, the number of communications in a single session, number of messages for authentication and memory size on tag are kept in a reasonable range compared with other protocols shown in Table 3. The memory size for each tag on backend database increases in our protocol because additional old secrets and random numbers are stored.

Table 3 gives a particular comparison among our work and other researchers' works.

For any protocol which is vulnerable to full-disclosure attack, we don't consider any other attack because an attacker can get all secrets of a tag in such protocol, so that s/he can implement others easily. Since the full-disclosure attack shown in [30] needs a massive of protocol sessions, we think that RAPP can resist the full-disclosure attack. We assume all strings used in protocols have the length of L.

5 Conclusions

In this paper, we have proposed a new ultralightweight RFID authentication protocol R^2AP based on a new bitwise operation named reconstruction. Reconstruction has three very important properties: Hamming weight unpredictability, irreversibility and effectiveness. Part of these three properties are absent in previous operations used in ultralightweight RFID protocols and the absence has caused many insecure issues in RFID system. The proposed protocol R^2AP takes advantage of reconstruction to guarantee its security functionality.

We also extended the UNT model [13] so that it can be used to analyze the abilities of the protocol to withstand traceability attack, to protect the data integrity, to withstand replay attack. Using this new model, our security analysis and performance evaluations demonstrate that R^2AP is indeed a secure and effective ultralightweight RFID protocol.

References

1. Ahmadian, Z., Salmasizadeh, M., & Aref, M. R. (2012). *Desynchronization attack on RAPP ultralightweight authentication protocol*, Cryptology ePrint Archive, Report 2012/490.
2. Ahmadian, Z., Salmasizadeh, M., & Aref, M. R. (2013). Recursive linear and differential cryptanalysis of ultralightweight authentication protocols. *IEEE Transactions on Information Forensics and Security*, *8*, 1140–1151.
3. Alomair, B., Lazos, L., & Poovendran, R. (2007). Passive attacks on a class of authentication protocols for RFID. In K.-H. Nam & G. Rhee (Eds.), *International conference on information security and cryptology-ICISC 2007*. Seoul, Korea: Lecture notes in computer science.
4. Avoine, G., Carpent, X., & Martin, B. (2012). Privacy-friendly synchronized ultralightweight authentication protocols in the storm. *Journal of Network and Computer Applications*, *25*, 826–843.
5. Avoine, G., & Carpent, X. (2012). *Yet another ultralightweight authentication protocol that is broken, in pre-proceeding of RFIDsec*. Netherlands: Nijmegen.
6. Barasz, M., Boros, B., Ligeti, P., Loja, K., & Nagy, D. A. (2007). Breaking LMAP. In: *Conference on RFID security*, Malaga, Spain.
7. Cao, T., Bertino, E., & Lei, H. (2009). Security analysis of the SASI protocol. *IEEE Transactions on Dependable and Secure Computing*, *6*, 73–77.
8. Castro, H., Tapiador, M. E., Lopez, P., & Quisquater, J. (2008). *Cryptanalysis of the SASI ultralightweight RFID authentication protocol with modular rotations*, arXiv preprint arXiv:0811.4257.
9. Chien, H. Y. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transaction on Dependable and Secure Computing*, *4*, 337–340.
10. David, M., & Prasad, N. R. (2009). Providing strong security and high privacy in low-cost RFID networks, In: *Proceedings of security and privacy in mobile information and communication systems, MobiSec 2009*, Heidelberg.

11. Eghdamian, A., & Samsudin, A. (2011). A secure protocol for ultralightweight radio frequency identification (RFID) tags. In *Information engineering and information science—ICIEIS 2011, Vol. 251 of communication in computer and information science*, Kuala Lumpur, Malaysia.
12. Jeon, I., & Yoon, E. (2013). A new ultra-lightweight RFID authentication protocol using merge and separation operations, *7*, 2583–2593.
13. Juels, A., & Weis, S. A. (March 2007). Defining strong privacy for RFID. In *Proceedings of fifth annual ieee international conference pervasive computing and communications*, pp. 342–347.
14. Klimov, A., & Shamir, A. (2003). A new class of invertible mappings. In *Proceedings of cryptographic hardware and embedded systems*, Lecture Notes in Computer Science, Vol. 2523, pp. 470–483.
15. Lee, Y. C., Hsieh, Y. C., You, P. S., & Chen, T. C. (2009). A new ultralightweight RFID protocol with mutual authentication. In *Proceedings of the 2009 WASE international conference on information engineering, vol. 1* (pp. 58–61). IEEE Computer Society.
16. Lee, Y. C. (2012). Two ultralightweight authentication protocols for low-cost RFID tags. *Applied Mathematics and Information Sciences*, *6*, 425–431.
17. Li, T. & Wang, G. (2007). Security analysis of two ultra-lightweight RFID authentication protocols. In *Proceedings of 22nd IFIP TC-11 Int'1 information security conference*, Sandton, Ganteng, South Africa.
18. Li, T., & Deng, R. (2007). Vulnerability analysis of EMAP-an efficient RFID mutual authentication protocol. In *Proceedings of second international conference on availability, reliability, and security (AReS'07)*, Vienna, Austria.
19. Lo, N. W., Yeh, K.-H., & Chen, H.-Y. (2012). Analysis against secret redundancy mechanism for RFID authentication protocol. In *2012 IEEE international conference on, communication, network and satellite (ComNetSat)*. IEEE.
20. Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Li, T. (2010). Quasi-linear cryptanalysis of a secure RFID ultralightweight authentication protocol. In *6th China international conference on information security and cryptology-Inscrypt'10*. Shanghai, China: Springer.
21. Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. (2006). EMAP: An efficient mutual authentication protocol for low-cost RFID tags. In O. T. M. Federated (Ed.) *Conferences and workshop: IS workshop*, Montpellier, France.
22. Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. (2006). LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In *Proceedings of 2nd workshop RFID security*. Graz, Austria: Ecrypt.
23. Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. (2006). M^2AP: a minimalist mutual-authentication protocol for low-cost RFID tags. In *Proceedings of 2006 international conference on ubiquitous intelligence and computing*, Wuhan and Three Gorges.
24. 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 *Information security applications*, pp. 56–68.
25. Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & van der Lubbe, J. C. A. (2010). Security flaws in a recent ultralightweight RFID protocol, In: *Workshop on RFID security-RFIDSec Asia'10*, Singapore.
26. Phan, R. C. W. (2009). Cryptanlysis of a new ultralightweight RFID authentication protocol-SASI. *IEEE Transactions on Dependable and Secure Computing*, *6*, 316–320.
27. Sun, H. N., Ting, W. C., & Wang, K. H. (2011). On the Security of Chien's ultralightweight RFID authentication protocol. *IEEE Transactions on Dependable and Secure Computing*, *8*, 315–317.
28. Tagra, D., Rahman, M., & Sampalli, S. (2010). Technique for preventing DoS attacks on RFID systems, In: *18th international conference on software telecommunications and computer networks-SoftCOM'10*. Bol, Island of Brac, Croatia: IEEE Computer Society.
29. Tian, Y., Chen, G., & Li, J. (2012). A new ultralightweight RFID authentication protocol with permutation. *IEEE Communications Letters*, *16*, 702–705.
30. Wang, S., Han, Z., Liu, S., & Chen, D. (2012). *Security analysis of RAPP: an RFID authentication protocol based on Permutation*, Cryptology ePrint Archive, Report 2012/327.
31. Zhuang, X., Zhu, Y., & Chang, C. C. (2013). *Security Analysis of Ultralightweight RFID Protocols*. Technique Report.
32. Zhuang, X., Wang, Z. H., Chang, C. C., & Zhu, Y. (2013). Security analysis of a new ultra-lightweight RFID protocol and its improvement. *Journal of Information Hiding and Multimedia Signal Processing*, *4*, 165–180.

Xu Zhuang received his B.S. degree in Computer Science from Southwest Jiaotong University (SWJTU), Chengdu, China, in 2006, and is currently pursuing the Ph.D. degree in the Software Engineering Laboratory in Southwest Jiaotong University (SWJTU), Chengdu, China. His research interests include data mining, data hiding and information security.

Yan Zhu received her B.S. and M.S. degrees in Computer Science from Southwest Jiaotong University (SWJTU), Chengdu, China, in 1986 and 1989, respectively. She received her Ph.D. degree in Computer Science from Darmstadt University of Technology, Germany in 2004. Yan Zhu is currently a professor of the School of Information Science and Technology, SWJTU and the director of the Laboratory of Software Engineering. Her research interests include data mining, Web information security, and Web spam detection.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. He is a Fellow of IEEE and a Fellow of IEE, UK. His research interests include database design, computer cryptography, image compression and data structures.