An authentication protocol for low-cost RFID tags

Chia-Hui Wei

Department of Computer Science,
National Tsing Hua University,
101, Section 2, Kuang Fu Road,
Hsinchu 300, Taiwan
E-mail: chwei@cs.nthu.edu.tw

Min-Shiang Hwang*

Department of Computer Science
and Information Engineering,
Asia University,
500, Lioufeng Rd.,
Wufeng, Taichung 41354, Taiwan
E-mail: mshwang@nchu.edu.tw
*Corresponding author

Augustin Yeh-Hao Chin

Department of Computer Science,
National Tsing Hua University,
101, Section 2, Kuang Fu Road,
Hsinchu 300, Taiwan
E-mail: yhchin@cs.nthu.edu.tw

Abstract: Radio Frequency Identification (RFID) system can be read by radio wave within several metres without having direct contact. Many research focus on improving security and privacy problem. Recently, Rizomiliotis et al. point out that Song-Mitchell protocol is vulnerable to the denial of service attack, since the attacker can easily modify the data to make the server and the tag out of synchronisation without detection. However, this paper pointed out that Rizomiliotis et al.'s modification was not correct. Therefore, we designed a new authentication scheme, which inherits the advantage of Song-Mitchell protocol and Rizomiliotis et al.'s protocol, along with the assumption that the insecure channel exists between the reader and the server. Finally, this paper provides a security analysis to comparison with other protocols.

Keywords: authentication; RFID; security; privacy; denial of service; mobile communication.

Reference to this paper should be made as follows: Wei, C-H., Hwang, M-S. and Chin, A.Y-H. (2011) 'An authentication protocol for low-cost RFID tags', *Int. J. Mobile Communications*, Vol. 9, No. 2, pp.208–223.

Biographical notes: Chia-Hui Wei received the MS in Information Management from Chaoyang University of Technology, Taiwan. Currently, she is pursuing her PhD in Computer Science from National Tsing Hua University, Taiwan. Her current research interests include RFID security, information security and mobile communications.

Min-Shiang Hwang is a Professor in the Department of Computer Science and Information Engineering at Asia University, Taiwan. His research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing. He received his PhD in Computer and Information Science from National Chiao Tung University, Taiwan. He is a member of IEEE, the ACM, and the Chinese Information Security Association.

Augustin Yeh-Hao Chin received his PhD from the University of Texas at Austin. Currently, he is a Professor with the Department of Computer Science at the National Tsing Hua University, Hsinchu County, Taiwan. His current research interests include database systems, algorithm design, system programming and software engineering.

1 Introduction

RFID is one of the automatic identification systems different from barcode. The RFID components include the database server, the reader and the tag (Landt, 2005; Roberts, 2006; DeVries, 2008; Lai et al., 2010). The tag is a small microchip combined with an antenna component with limited processing units and limited memory. The tag can be read by radio wave within several metres without having direct contact or line of sight scanning, and then the reader can transfer the tag to the database server. The database server can authenticate whether the tag is legal or not. Although RFID is attractive in convenience and efficiency, its downside includes the security and privacy problems because the tag is being used in an open environment (Chen and Pfleuger, 2008; Juels and Weis, 2009; Lehou, 2009; Hwang et al., 2009a, 2009b; Lin et al., 2010). For example, an unauthorised reader near your bag could read the tag and knows what products you have bought. One way to solve the security and privacy problem is cryptography. However, traditional cryptography (Hwang et al., 2005, 2009a, 2009b, 2005; Ou et al., 2009) was not adopted in designing RFID because tag has limited computation and memory. On the other hand, many researchers have proposed various cryptographic operations in security mechanisms and attack models in RFID system, but some of these operations are not based on low-cost tag (Zhang and King, 2008; Cao and Shen, 2009; Juang and Wu, 2009).

The low-cost tag brings low cost on both computation and memory space in the tag, while performing simple arithmetic operations. In Song and Mitchell (2008) proposed an RFID authentication protocol for low-cost tags. The performance of this protocol is comparably better than others (Hwang et al., 2005; Lim and Takeyoung, 2006)) but the security is not. Rizomiliotis et al. (2009) have demonstrated that

Song-Mitchell's protocol is vulnerable to impersonating of a legitimate reader/server and de-synchronising attack between the reader/server and the tag. Therefore, Rizomiliotis et al. improved Song-Mitchell's protocol and inherited the advantages of the performance of Song-Mitchell's protocol. In Rizomiliotis et al.'s protocol, they claimed that their protocol is superior to Song-Mitchell's protocols with regard to security. The advantages of these two protocols are based on the performance, especially when the tag stores only one value that saves memory space. However, these two protocols assumed that secure channel does exist between the reader and the server. On the other hand, the server and the reader use wire to connect perpetually so such assumption is not considered practical in the real world. In fact, the goods are often distributed in a warehouse and it is usually big enough to store the entire inventory (Chen et al., 2011). The storekeeper uses the reader to move around the goods and to make an inventory for the warehouse. In such environment, the distance between the server and the reader is too long to be connected by wire. Hence, a more reasonable assumption is to assume that the channel is insecure between the reader and the server. Yeo et al. (2009) has made a similar assumption compared with ours, but with a different environment than that of Song-Mitchell's and Rizomiliotis et al.'s protocol. Yeo et al.'s research was based on the mobile agent in RFID environment. This protocol needs an extra mobile agent to design the authentication protocol while Song-Mitchell protocol and Rizomiliotis et al.'s protocol do not. Therefore, we inherited both the advantages of Song-Mitchell protocol and Rizomiliotis et al.'s improved protocol to design a low-cost tag with the assumption of the existing insecure channel between the reader and the server.

The remainder of this paper is organised as follows: Section 2 introduces the RFID protocol requirements to evaluate privacy and security problems. Section 3 describes a new low-cost tag authentication protocol in detail. Section 4 presents the comparison of our protocol with others. Conclusions are finally made in Section 5.

2 RFID protocol requirement

In this session, we proposed the following criteria for evaluation of the privacy and security of authentication protocol.

2.1 Privacy

- *Tag tracing*: The tag could be traced if unauthorised reader obtained a link on all of the information response at the same tag from different location. Because the tag always broadcasts a fixed number or static ID to the reader, the unauthorised reader can collect and analyse this number. For example, when a user uses automatic road toll RFID system, the tag always response the static ID to the readers, which is distributed everywhere. The unauthorised reader can collect the response of the tag from different locations and then locate the matched static id. The information would expose the user's privacy such as the time and place of toll stations you have been to.

- *Individual data privacy*: The tag could expose individual data if the unauthorised reader has successfully counterfeited a legal reader. The counterfeited reader will attempt to query the tag to obtain the id. After gaining the tag's id, the counterfeited reader will be checked through successfully by the database server and then could acquire all of the individual data from the tag. For example, if the medical records are attached to the tag, the counterfeited reader could try to obtain ID from the tag and then respond the ID to the server. As a result, the counterfeited reader could be successfully verified by the server and then the tag would send user's medical record to the counterfeited reader.

2.2 Security

- *Tag cloning*: The illegal reader first queries the tag and gain the information from the tag. After collecting the related information, tag cloning is done by writing all the related data into a counterfeit tag, and then uses it to cheat the reader or the database server. For example, the tags are usually attached to products within open environments such as supermarkets, hospital and other public places. The illegal reader can furtively query the tag to collect the information from the tags followed by cloning a counterfeit tag to replace the genuine tag.
- *Eavesdropping*: Eavesdropping on RFID system is a major threat. The attacker surreptitiously listens to all the communications between the reader and the tag because they communicate via radio frequency, which is easy to be sniffed or eavesdropped. For example, a competitive company would hire spy for espionage. The tags are usually attached to clothes on sale within a store. When a user pays the bill for clothes, the attacker of competitive company could stealthily listen to the communication channel between the reader and the tag by eavesdropping. After collecting the desired information, the competitive company can analyse the information to find out its competitor's store policy.
- *Replay attack*: The attacker repeats or delays the message when valid data is transmitted. The tag sends ID to the reader to recognise its identity. Meanwhile, the attacker eavesdrop this message and keep the ID. After the tag and the reader have finished all the communications, the attacker retransmits ID to the reader, so the attacker can try to cheat and spoof the reader to pass verification.
- *Denial of Service*: The attacker sends a massive amount of message to the server and attempts to crash the server, which will result in the server unavailability to its intended tags and data inconsistency to respond to the tags. For example, the attacker sends large ID to the database server. The database server would spend time to search these pointless IDs to check if they are matched or not without the time to deal with the demands from the genuine tag. Besides this, the attacker interrupts the message to cause the tag not being able to update the secret value after the secret value of the server has been updated. Such situation would result in the secret value de-synchronisation because the server updates the secret while the tag does not.

- *Forward security*: The attacker can compromise a tag, obtain its current data, and possibly trace future transaction record. For example, when the tag is attached to a passport, the tag's ID is used to verify by the reader, and the attacker compromises the tag to obtain the ID. Hence, the attacker can trace records of that user's future boarding information that depends on whether the ID is matched or not.

3 A new authentication scheme for low-cost RFID tag

In this section, we present a new authentication protocol based on low-cost RFID tag. Our protocol inherits the advantage with low-cost RFID tag of Song-Mitchell's protocol and Rizomiliotis's protocol along with our own improvements it has low-cost RFID tag, and the assumption that the channel between the reader and the server is insecure.

3.1 Basic concept

The basic concept behind our protocol contains the adoption of a HF and Message Authentication Code (MAC) algorithm, or keyed HF, which are described in the following (Bakhtiari et al., 1996; Menezes et al., 1996).

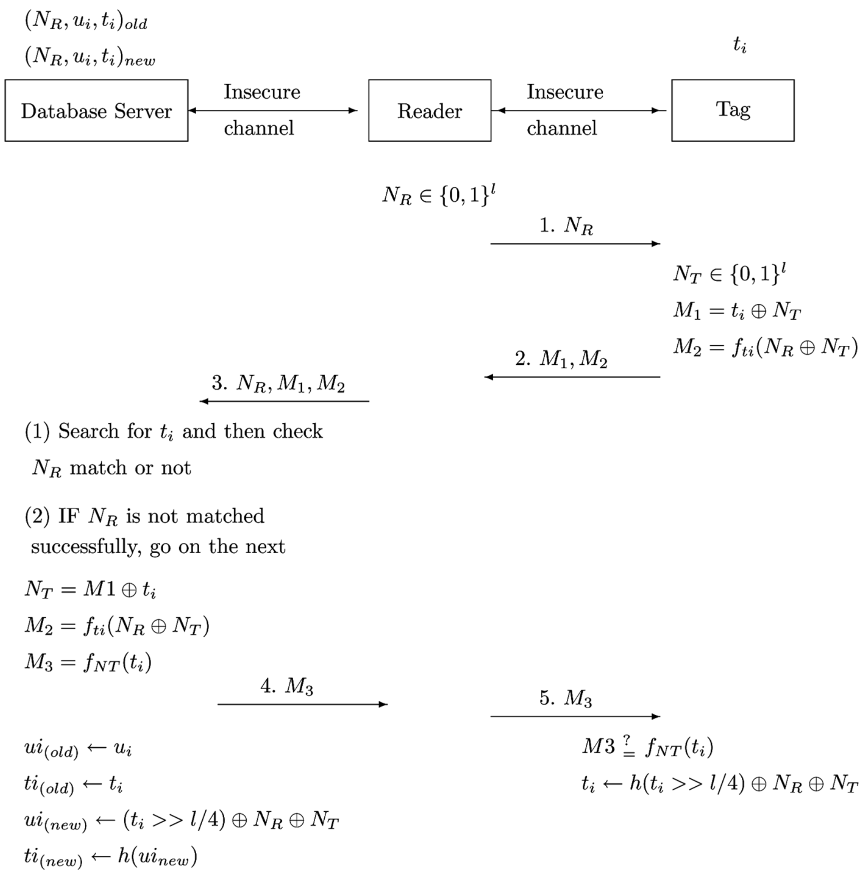
- *Hash Function*: Given h and an input x of arbitrary finite length, it can compute an output $h(x)$ of fixed length n. For output y, it is computationally infeasible to find an input x such that $h(x) = y$. For input x, it is computationally infeasible to find $x' \neq x$ such that $h(x') = h(x)$. For any two distinct inputs x, x', it is computationally infeasible to find out the same output such that $h(x) = h(x')$.
- *MAC algorithm*: Given a secret value k and an input x of arbitrary finite length, it can compute an output $f_k(x)$ of fixed length n.

3.2 The initial stage and authentication stage

In our proposed protocol, the critical factor is that the server storage, both new and old, are random bit-string generated from the reader. The server successfully verifies the reader if the server have determined that the stored random bit-string and the received random bit-string are not the same so that the attacker cannot replay the message. The detailed steps are explained as follows. The notations that are used throughout the paper is listed in Nomenclature.

There are two stages in the protocol, namely the initial stage and the authentication stage. During the initial stage, each tag has a t_i, and it has a formula $t_i = h(u_i)$ with a string u_i composed of l bits. The length of l bits should be long enough so that an exhaustive search for finding u_i and t_i would be computationally infeasible. The server stores the current and previous tag identity pair and random number, immediate $(N_R, u_i, t_i)_{\text{new}}$, $(N_R, u_i, t_i)_{\text{old}}$. The values $(N_R, u_i, t_i)_{\text{new}}$ and $(N_R, u_i, t_i)_{\text{old}}$ are both set in the initial stage. There are five steps in the authentication stage as shown in Figure 1.

Figure 1 A new authentication protocol for low-cost RFID tag in mobile environment


$(N_R, u_i, t_i)_{old}$
$(N_R, u_i, t_i)_{new}$
t_i
Database Server
Insecure channel
Reader
Insecure channel
Tag
$N_R \in \{0,1\}^l$
1. N_R
$N_T \in \{0,1\}^l$
$M_1 = t_i \oplus N_T$
$M_2 = f_{ti}(N_R \oplus N_T)$
2. M_1, M_2
3. N_R, M_1, M_2
(1) Search for t_i and then check N_R match or not
(2) IF N_R is not matched successfully, go on the next
$N_T = M1 \oplus t_i$
$M_2 = f_{ti}(N_R \oplus N_T)$
$M_3 = f_{NT}(t_i)$
4. M_3
5. M_3
$ui_{(old)} \leftarrow u_i$
$ti_{(old)} \leftarrow t_i$
$ui_{(new)} \leftarrow (t_i >> l/4) \oplus N_R \oplus N_T$
$ti_{(new)} \leftarrow h(ui_{new})$
$M3 \overset{?}{=} f_{NT}(t_i)$
$t_i \leftarrow h(t_i >> l/4) \oplus N_R \oplus N_T$

1 The reader generates a random bit-string N_R:

$$N_R \in_R \{o, 1\}^l \tag{1}$$

and sends it to the tag.

2 The tag generates a random bit-string N_T and computes M_1 and M_2 as follows:

$$N_T \in_R \{o, 1\}^l \tag{2}$$
$$M_1 = t_i \oplus N_T \tag{3}$$
$$M_2 = f_{ti}(r_{\oplus}N_T). \tag{4}$$

Then, the tag responds the message (M_1, M_2) to the reader.

3 The reader transmits M_1, M_2, and its random bit-string N_R to the server.

4 The server will compute N_T and M_2 depending on whether if the server can find a matched t_i from database $(N_R, u_i, t_i)_{\text{new}}$ or $(N_R, u_i, t_i)_{\text{old}}$; otherwise, the message will fail, and then the server sends an error message to the reader, so the reader will forward it to the tag. If the server finds out the matched values for t_i, the server

will check whether N_R is matched or not. The server goes onto the next step if the N_R is not matched; otherwise, the server omits the message. The server will use u_i of the pair to compute $M_3 = f_{N_T}(t_i)$, sends M_3 to the reader, and then the reader will transmit it to the tag. The server updates $(u_i, t_i)_{\text{old}} \leftarrow (u_i, t_i)$ by the current t_i value and updates $(u_i)_{\text{new}} \leftarrow (t_i{>>}l/4) \oplus N_R \oplus N_T$, and $(t_i)_{\text{new}} \leftarrow h((u_i)_{\text{new}})$ for the next t_i. The detailed steps are as follows.

$$M_T = M_1 \oplus t_i \tag{5}$$

$$M_2 \overset{?}{=} f_{ti}(r_{\oplus}N_T) \tag{6}$$

$$M_3 = f_{NT}(t_i). \tag{7}$$

5 After the tag receives the message, the tag computes $f_{N_T}(t_i)$ by its t_i and verifies whether the value $M_3 \overset{?}{=} f_{N_T}(t_i)$. If the verification is successful, the tag will update $t_i \leftarrow h((t_i{>>}l/4) \oplus N_R \oplus N_T)$; otherwise, the tag t_i keeps it.

3.3 Performance analysis

RFID tag has limited capacity in memory, hence the memory space is not large enough to store the related data. For the storage cost, only one value t_i in the RFID tag while none is stored in the reader, and two pairs value $(N_R, u_i, t_i)_{\text{new}}$ and $(N_R, u_i, t_i)_{\text{old}}$ are stored in the server in our protocol. For the computational cost, the HF, MAC algorithm, XOR and random number generation (RG) are used in this study. The tag computes $1RG + 2MAC + 2XOR$, and the reader computes $1RG$ only, and the server computes $2MAC + 2XOR$. The tag and the server can also compute $1HF + 3XOR$ in the updated stage. For the communication cost, the 2 exchanged messages between the reader and the server are shown in Steps (3) and (4), and the 3 exchanged messages between the tag and the reader are shown in Steps (1), (2) and (5) in Figure 1. Although the server seems to have a large load, its memory is capable for computations and with available spaces to store the value.

3.4 Security analysis

The RFID protocol requirements are described in Section 2. In our protocol, we assumed that the insecure channel exists between the reader and the server and also between the tag and the reader. The security of the new low-cost RFID tag protocol relies on the tag secrets t_i, the random number equation (1) stored in the server, and the one-way HFs and MAC. Our protocol has the following privacy and security properties.

3.4.1 Privacy

- *Resistance towards tag tracing*: For tag tracing, the attacker attempts to get tag's ID from equations (3) and (4). The tag sends equations (3) and (4), which are not fixed value, because equations (3) and (4) are computed using secret key t_i and session random bit-string N_T, which is also not a fixed number. Even if an eavesdropper could obtain equations (3) and (4) many times from the same tag, it still cannot identify the tag since equations (3) and (4) are considered to be anonymous.

- *Individual data privacy*: In this protocol, the attacker can obtain individual data privacy when the attacker has been successfully verified by the server. To get an individual's data, the attacker queries the tag to obtain equations (1), (3) and (4), and then responds them to the server to acquire equation (7). The attacker would fail the verification when the received equation (1) is matched with computed equation (1) by the server. Therefore, the server omits the message. Besides, the message equation (7) is encrypted by N_T, which is a random number thus the attacker cannot get any useful information from this message.

3.4.2 Security

- *Resistance towards tag cloning*: For cloning the tag, the attacker needs to forge equations (3) and (4). However, the attacker cannot compute them because the attacker cannot obtain the secret value t_i and equation (2) from the tag. As a result, cloning the tag would eventually fail.
- *Resistance towards eavesdropping*: For eavesdropping, the attacker needs to listen to equations (1), (3), (4) and (7) surreptitiously between the tag and the server. However, the attacker would get nothing from equation (3) if the attacker cannot obtain secret key t_i and session secret N_T. In addition, equations (4) and (7) are computed by HF, which makes it impossible to inverse the value. Therefore, the attacker cannot obtain any useful information.
- *Resistance towards replay attack*: In a replay attack, the attacker attempts to reuse the previous equations (3)–(5). Equations (3)–(5), which used equations (1) and (2) to compute the value generated in each session. Even if the attacker tries to reuse equation (1), it would not pass the verification by the server since the server can only be checked through successfully if equation (1) is not equal with previous equation (1). In other words, the attacker using previous equation (1) cannot pass the check successfully. Besides, equation (2) kept secret, so the attacker cannot locate it and therefore not able to reuse it. This scenario proves that our protocol is capable of resisting replay attack.
- *Resistance towards denial of service*: The attacker can interrupt Step 5, equation (7), which would result in the update of secret data in the server but not in the tag. However, the server is still available with the tag in next session since the server stores the previous value $(N_T, u_i, t_i)_{\text{old}}$, hence the old value of the tag can be verified. After the server successfully verifies the new value, it is renewed and stored in the server therefore the attacker cannot break off synchronisation of the RFID system.
- *Resistance towards forward security*: For forward security, the attacker needs to know the tag's data. But, it is difficult to trace future transactions even if the attacker could compromise a tag and obtain t_i. Also, equation (2) is updated in each session, the attacker still does not know equation (2) even if it knows the current t_i, which will prevent the attacker from computing equation (4). Since the attacker cannot compute the next equation (2) in our protocol, we have proved that it can resist forward security.

3.5 Comparison

In this section, we will compare our method with other low-cost RFID tag and assumed that the insecure channel exists between the reader and the server in literature, using the criteria as stated in Section 2. In Song-Mitchell (2008) protocol stores secret value t_i in memory and the server stores $(u_i, t_i)_{\text{new}}, (u_i, t_i)_{\text{old}}, D_i$ in database, the detailed steps are as follows:

- The reader generates a random bit-string N_R and then sends it to the tag.
- After the tag receives the N_R, it generates a random bit-string N_T and then computes $M_1 = t_i \oplus N_T$ and $M_2 = f_{ti}(N_R \oplus N_T)$. The tag sends M_1 and M_2 to the reader. The reader receives the message, and forwards M_1, M_2, and N_T to the server.
- The server searches t_i in its database and computes M_1 and M_2 to check whether they matched with the received value or not. The server is stopped if the value is not matched; otherwise, the server goes onto the next step. The server computes $M_3 = u_i \oplus (N_T >> l/2)$ and updates the value $u_{i(\text{old})} = u_i$, $t_{i(\text{old})} = t_i$, $u_{i(\text{new})} = (u_i << l/4) \oplus (t_i >> l/4) \oplus N_R \oplus N_T$, and $t_{i(\text{new})} = h(u_{i(\text{new})})$. The server sends (D_i, M_3) to the reader, and the reader forwards M_3 to the tag.
- The tag computes u_i depending on $M_3 \oplus (N_T >> l/2)$ and checks if $h(u_i) \overset{?}{=} t_i$ is true or not. The tag updates $t_i = h((u_i << l/4) \oplus (t_i >> l/4) \oplus N_T \oplus N_R)$ if $h(u_i) \overset{?}{=} t_i$ is successfully checked.

Rizomiliotis et al.'s protocol is similar with Song-Mitchell protocol. But, Rizomiliotis et al. point out that Song-Mitchell protocol is vulnerable to the denial of service attack since the attacker can easily modify the data to make the server and the tag out of synchronisation without detection (Rizomiliotis et al., 2009). Therefore, Rizomiliotis et al. modified the message $M_3 = f_{N_T}(u_i)$ to resist denial of service attack. However, Rizomiliotis et al.'s modification is not correct due to $f_{N_T}(u_i) \neq f_{N_T}(h(t_i))$. We have proved $f_{N_T}(u_i) \neq f_{N_T}(h(t_i))$ by the following theorem.

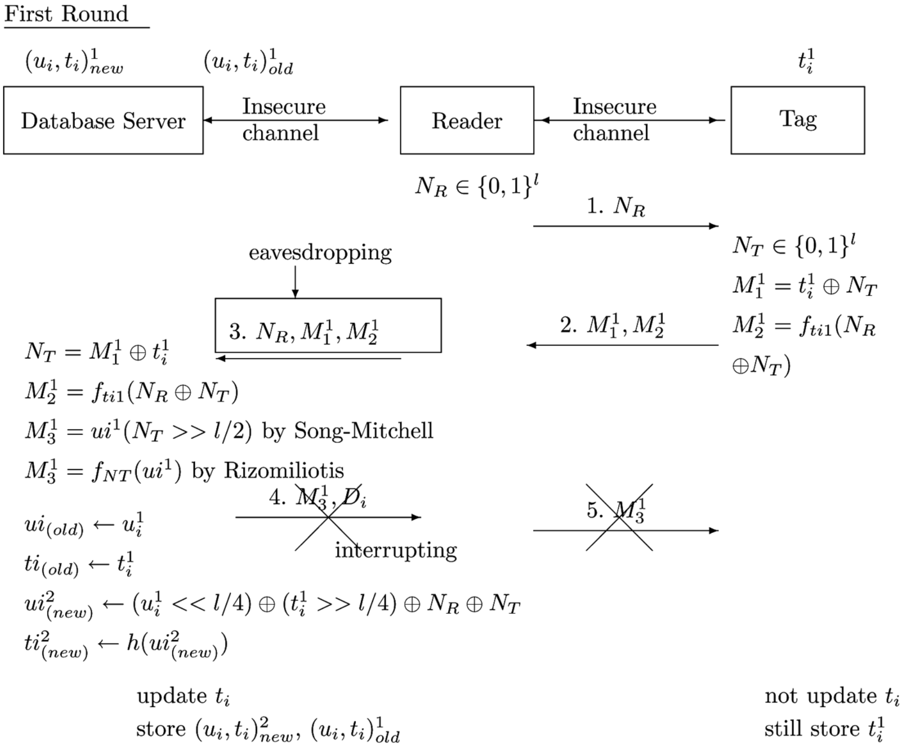
Theorem 3.1: *Let f_{N_T} and $h(t_i)$ have the same definition as stated in Section 3.1. Then, $f_{N_T}(u_i) \neq f_{N_T}(h(t_i))$.*

Proof: The value t_i is computed from $h(u_i)$, namely $t_i \longleftarrow h(u_i)$. It means that the output t_i has a fixed length n instead of the original length, and t_i is computationally infeasible to find u_i, namely $u_i \neq h(t_i)$. Therefore, $f_{N_T}(u_i) \neq f_{N_T}(h(t_i))$. □

Example 3.2: We assumed that input u_i has a length of 1049 bits, secret key N_T has a length of 256 bits, and output t_i has a fixed length of 128 bits by MD5. Therefore, the length of $t_i \longleftarrow h(u_i)$ is 128 bits, and it is computationally infeasible to inverse the value $u_i \longleftarrow h(t_i)$. In other words, t_i's length, 128 bits, is impossible to generate the original length of 1049 bits as u_i.

The Song-Mitchell and Rizomiliotis et al.'s protocol have low memory space and no extra device to design the authentication protocol. However, we have described that an attacker can impersonate a reader and permanently de-synchronise the secret value t_i when the insecure channel between the server and the reader is used in Song-Mitchell and Rizomiliotis et al.'s protocol as shown in Figures 2–4.

Figure 2 Weakness in Song-Mitchell and Rizomiliotis et al.'s protocol – the first round

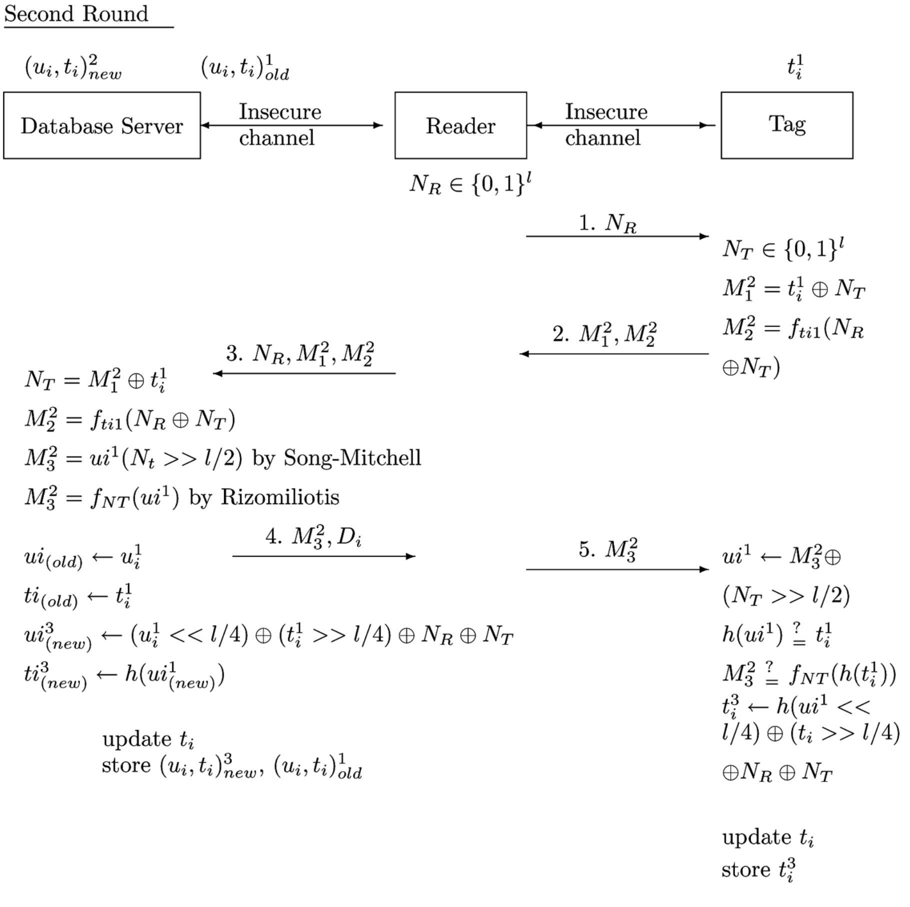

First Round
$(u_i,t_i)^1_{new}$
$(u_i,t_i)^1_{old}$
t_i^1
Database Server
Insecure channel
Reader
Insecure channel
Tag
$N_R \in \{0,1\}^l$
1. N_R
$N_T \in \{0,1\}^l$
$M_1^1 = t_i^1 \oplus N_T$
$M_2^1 = f_{ti1}(N_R \oplus N_T)$
eavesdropping
3. N_R, M_1^1, M_2^1
2. M_1^1, M_2^1
$N_T = M_1^1 \oplus t_i^1$
$M_2^1 = f_{ti1}(N_R \oplus N_T)$
$M_3^1 = ui^1(N_T >> l/2)$ by Song-Mitchell
$M_3^1 = f_{NT}(ui^1)$ by Rizomiliotis
4. M_3^1, D_i
interrupting
5. M_3^1
$ui_{(old)} \leftarrow u_i^1$
$ti_{(old)} \leftarrow t_i^1$
$ui^2_{(new)} \leftarrow (u_i^1 << l/4) \oplus (t_i^1 >> l/4) \oplus N_R \oplus N_T$
$ti^2_{(new)} \leftarrow h(ui^2_{(new)})$
update t_i
store $(u_i,t_i)^2_{new}$, $(u_i,t_i)^1_{old}$
not update t_i
still store t_i^1

Step 1: In the first round (see Figure 2), the reader sends a random bit-string N_R and queries to the tag. The tag computes message (M_1^1, M_2^1) by t_i^1 and responds it to the reader. The reader forwards (M_1^1, M_2^1, N_R) to the server. The attacker eavesdrops the message (M_1^1, M_2^1, N_R) and stores it in the attacker's database, and then interrupts the message M_3^1 at the end of the protocol so that the tag will not update its secret value t_i. The tag still holds the previous value t_i^1 however, the server is updated along with its database $(u_i,t_i)_{\text{old}} = (u_i,t_i)^1$ and $(u_i,t_i)_{\text{new}} = (u_i,t_i)^2$.

Step 2: In the second round, the authentication protocol takes place normally. We allowed the reader and the tag to run the protocol again without intervention. The server can identify the tag because the value t_i^1 of the tag is still stored in the server $(u_i,t_i)_{\text{old}} = (u_i,t_i)^1$, thus the server can compute the value N_T, M_1^2, and M_2^2. Next, the server sends the message M_3^2 to the tag. The server would update its database $(u_i,t_i)_{\text{old}} = (u_i,t_i)^1$ and $(u_i,t_i)_{\text{new}} = (u_i,t_i)^3$, because the random bit-string is changed to replace the previous one. Therefore, $(u_i,t_i)_{\text{new}} = (u_i,t_i)^3$ is also updated. The secret value t_i^3 of the tag is updated and stored as shown in Figure 3.

Step 3: In the third round (see Figure 4), the attacker forges a legal reader to send the previous message (M_1^1, M_2^1, N_R), which was eavesdropped in Step 1 from the reader. The server can verify the message (M_1^1, M_2^1, N_R) because the server can locate the value t_i^1 from the database $(u_i, t_i)_{\text{old}} = (u_i, t_i)^1$. Therefore, the server refreshes its database $(u_i, t_i)_{\text{old}} = (u_i, t_i)^1$, and $(u_i, t_i)_{\text{new}} = (u_i, t_i)^2$. After updating the database, the attacker interrupts the message M_3^1 so the value t_i of the tag is still t_i^3, which is not updated. Since the value $(u_i, t_i)^2_{\text{new}}$ and $(u_i, t_i)^1_{\text{old}}$ of the server is de-synchronised, the genuine tag is caused by not being able to implement the authentication protocol permanently .

Figure 3 Weakness in Song-Mitchell and Rizomiliotis et al.'s protocol – the second round

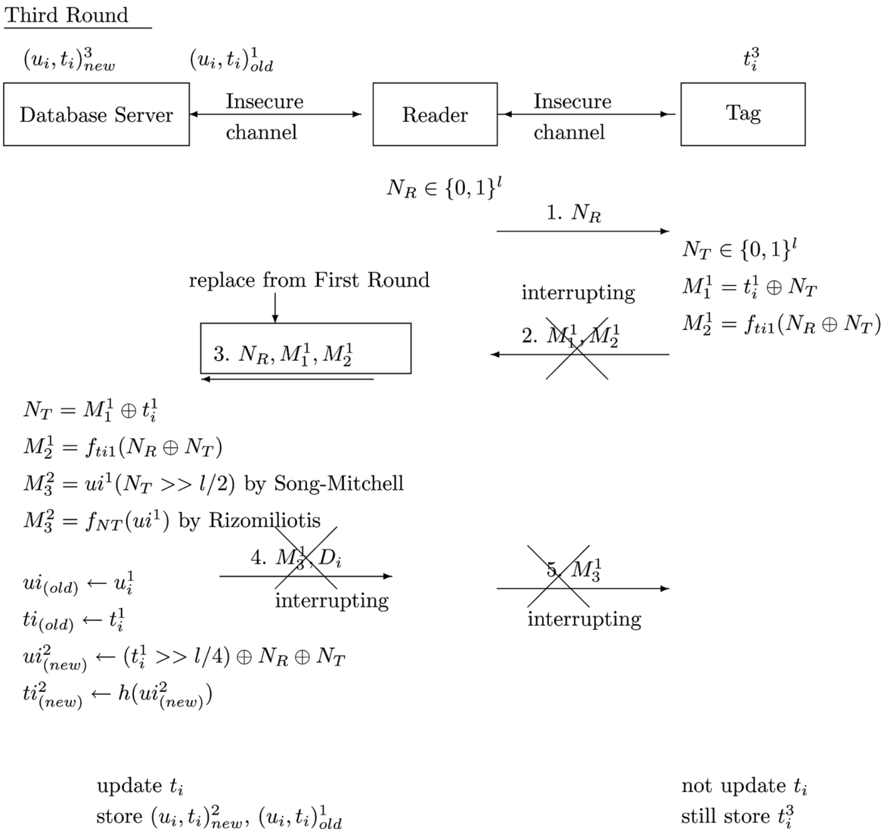

Second Round
$(u_i, t_i)^2_{new}$
$(u_i, t_i)^1_{old}$
t_i^1
Database Server
Insecure channel
Reader
Insecure channel
Tag
$N_R \in \{0,1\}^l$
1. N_R
$N_T \in \{0,1\}^l$
$M_1^2 = t_i^1 \oplus N_T$
$M_2^2 = f_{ti1}(N_R \oplus N_T)$
2. M_1^2, M_2^2
3. N_R, M_1^2, M_2^2
$N_T = M_1^2 \oplus t_i^1$
$M_2^2 = f_{ti1}(N_R \oplus N_T)$
$M_3^2 = ui^1(N_t >> l/2)$ by Song-Mitchell
$M_3^2 = f_{NT}(ui^1)$ by Rizomiliotis
4. M_3^2, D_i
5. M_3^2
$ui_{(old)} \leftarrow u_i^1$
$ti_{(old)} \leftarrow t_i^1$
$ui^3_{(new)} \leftarrow (u_i^1 << l/4) \oplus (t_i^1 >> l/4) \oplus N_R \oplus N_T$
$ti^3_{(new)} \leftarrow h(ui^1_{(new)})$
update t_i
store $(u_i, t_i)^3_{new}$, $(u_i, t_i)^1_{old}$
$ui^1 \leftarrow M_3^2 \oplus (N_T >> l/2)$
$h(ui^1) \overset{?}{=} t_i^1$
$M_3^2 \overset{?}{=} f_{NT}(h(t_i^1))$
$t_i^3 \leftarrow h(ui^1 << l/4) \oplus (t_i >> l/4) \oplus N_R \oplus N_T$
update t_i
store t_i^3

Previous similar protocols are briefly reviewed later and security characteristic comparison is shown in Table 1. Han et al. (2007) proposed a mutual authentication protocol based on synchronised secret information. However, this protocol still could not resist desynchronising (Pei, 2009). Chen et al. (2008) proposed a protocol based on quadratic residue to enhance security and privacy protection but the protocol is

Table 1 Security characteristic comparison

	Han et al. (2007)	*Chen et al. (2008)*	*Lim et al. (2008)*	*Song and Mitchell (2008)*	*Rizomiliotis et al. (2009)*	*Ours*
Tag tracing	O	X	O	O	O	O
Individual data privacy	O	O	X	O	O	O
Tag cloning	O	O	O	O	O	O
Eavesdropping	O	X	O	O	O	O
Replay attack	X	X	O	O	O	O
Denial of service	O	O	O	X	X	O
Forward security	X	X	O	O	O	O

Figure 4 Weakness in Song-Mitchell and Rizomiliotis et al.'s protocol – the third round


Third Round
$(u_i, t_i)^3_{new}$
$(u_i, t_i)^1_{old}$
t_i^3
Database Server
Insecure channel
Reader
Insecure channel
Tag
$N_R \in \{0,1\}^l$
1. N_R
$N_T \in \{0,1\}^l$
$M_1^1 = t_i^1 \oplus N_T$
$M_2^1 = f_{ti1}(N_R \oplus N_T)$
replace from First Round
interrupting
2. M_1^1, M_2^1
3. N_R, M_1^1, M_2^1
$N_T = M_1^1 \oplus t_i^1$
$M_2^1 = f_{ti1}(N_R \oplus N_T)$
$M_3^2 = ui^1(N_T >> l/2)$ by Song-Mitchell
$M_3^2 = f_{NT}(ui^1)$ by Rizomiliotis
4. M_3^1, D_i
interrupting
5. M_3^1
interrupting
$ui_{(old)} \leftarrow u_i^1$
$ti_{(old)} \leftarrow t_i^1$
$ui^2_{(new)} \leftarrow (t_i^1 >> l/4) \oplus N_R \oplus N_T$
$ti^2_{(new)} \leftarrow h(ui^2_{(new)})$
update t_i
store $(u_i, t_i)^2_{new}$, $(u_i, t_i)^1_{old}$
not update t_i
still store t_i^3

vulnerable to location privacy and it suffers from replay attack (Yeh et al., 2010). Lim et al. (2008) designed a complete mutual authentication protocol based on Ohkubo et al. (2004) and Lee et al. (2005) hash chain protocols since Ohkubo's protocol cannot resist replay attacks while Lee's protocol cannot resist tracking. Lim's is designed to repair these flaws but we pointed out that Lim's protocol cannot resist replay attack and denial of service since the secret value stored in the tag is not updated. An adversary could obtain the last message and counterfeit a random number to cheat the backend database. After this last message has been successfully verified, the ID could be updated and interrupted by the message, which updates the tag, and this procedure is being executed twice. Thus, denial of service attack has been successfully executed.

4 Conclusion

To design a powerful new authentication based on low-cost RFID tag, we added the assumption that the insecure channel exists between the reader and the server as shown in Section 3. In terms of the criteria for evaluating the privacy and security of authentication protocol, our protocol has shown resistance towards the privacy and security attack and success with practical assumption in real environment as well. In this paper, our contributions are listed as follows.

- The last step of Rizomiliotis et al.'s protocol is not correct in HF. We have proved its mistake in Section 4.
- We have proved that Song-Mitchell and Rizomiliotis et al.'s protocol cannot resist denial of service attack when the insecure channel exists between the server and the reader. Such attack would permanently result in the secret value's de-synchronisation.
- The proposed protocol inherits low-cost RFID tag of Song-Mitchell's and Rizomiliotis et al.'s protocol and improves its defect. From the system managerial standpoint, designing synchronisation with updated data is important to avoid system failure. The original authentication protocol by Song-Mitchell et al. cannot resist the de-synchronisation attacks and the impersonation of legitimate reader. Rizomiliotis et al.'s protocol is also not correct since it violates the properties and definition of HF. Our protocol does more than resisting impersonation and de-synchronisation attacks by adding the assumption that the insecure channel exists between the reader and the server. However, our proposed protocol does not fit with large scalar tag. The server of both Rizomiliotis et al.'s protocol and ours have large computation load when multi-tag needs verification simultaneously. We can use index or other methods to solve this problem in future study.

Acknowledgements

The authors acknowledge the many helpful suggestions of this paper. We also thank the Editor of this Journal.

References

Bakhtiari, S., Safavi-Naini, R. and Pieprzyk, J. (1996) 'Keyed hash functions', *Lecture Notes in Computer Science*, Vol. 1029, pp.201–214.

Cao, T. and Shen, P. (2009) 'Cryptanalysis of two RFID authentication protocols', *International Journal of Network Security*, Vol. 9, pp.95–100.

Chen, C.L., Lai, Y.L., Chen, C.C., Deng, Y.Y. and Hwang, Y.C. (2011) 'RFID ownership transfer authorization systems conforming EPCglobal class-1 generation-2 standards', *International Journal of Network Security*, Vol. 13, No. 1, pp.41–48.

Chen, J.V. and Pfleuger Jr., P.P. (2008) 'RFID in retail: a framework for examining consumers' ethical perceptions', *International Journal of Mobile Communications*, Vol. 6, No. 1, pp.53–66.

Chen, Y., Chou, J.S. and Sun, H.M. (2008) 'A novel mutual authentication scheme based on quadratic residues for RFID systems', *Computer Networks*, Vol. 52, No. 12, pp.2373–2380.

DeVries, P.D. (2008) 'The state of RFID for eRective baggage tracking in the airline industry', *International Journal of Mobile Communications*, Vol. 6, No. 2, pp.151–164.

Han, S., Potdar, V. and Chang, E. (2007) 'Mutual authentication protocol for RFID tags based on synchronized secret information with monitor', *Lecture Notes in Computer Science*, Vol. 4707, pp.227–238.

Hwang, M.S. and Liu, C.Y. (2005) 'Authenticated encryption schemes: current status and key issues', *International Journal of Network Security*, Vol. 1, pp.61–73.

Hwang, M.S., Lee, C.C. and Tzeng, S.F. (2009a) 'PA new Knapsack public-key cryptosystem based on permutation combination algorithm', *International Journal of Applied Mathematics and Computer Sciences*, Vol. 5, No. 1, pp.33–38.

Hwang, M.S., Wei, C.H. and Lee, C.Y. (2009b) 'Privacy and security requirements for RFID applications', *Journal of Computers*, Vol. 20, No. 3, pp.55–60.

Hwang, M.S., Chong, S.K. and Chen, T.Y. (2005) 'DoS-resistant ID-based password authentication scheme using smart cards', *Journal of Systems and Software*, Vol. 83, pp.163–172.

Juang, W.S. and Wu, J.L. (2009) 'Robust and efficient authenticated key agreement in mobile communications', *International Journal of Mobile Communications*, Vol. 7, No. 5, pp.562–579.

Juels, A. and Weis, S.A. (2009) 'Defining strong privacy for RFID', *ACM Transactions on Information and System Security*, Vol. 12, No. 1, pp.7:1–7:23.

Lai, C.L., Fang, K. and Chien, S.W. (2010) 'Enhanced monitoring of tuberculosis patients by using RFID technologies', *International Journal of Mobile Communications*, Vol. 8, No. 2, pp.244–256.

Landt, J. (2005) 'The history of RFID', *IEEE Potentials*, Vol. 24, No. 4, pp.8–11.

Lee, S.M., Hwang, Y.J., Lee, D.H. and Lim, J.I. (2005) 'Efficient authentication for low-cost RFID systems', *Lecture Notes in Computer Science*, Vol. 3480, pp.619–627.

Lehlou, N. (2009) 'An online RFID laboratory learning environment', *IEEE Transactions on Learning Technologies*, Vol. 2, pp.295–303.

Lim, C.H. and Takeyoung, K. (2006) 'Strong and robust RFID authentication enabling perfect ownership transfer', *Lecture Notes in Computer Science*, Vol. 4307, pp.1–20.

Lim, J., Oh, H. and Kim, S. (2008) 'A new hash-based RFID mutual authentication protocol providing enhanced user privacy protection', *Lecture Notes in Computer Science*, Vol. 4991, pp.278–289.

Lin, I.C., Yang, C.W. and Tsaur, S.C. (2010) 'Nonidentifiable RFID privacy protection with ownership transfer', *International Journal of Innovative Computing, Information and Control*, Vol. 6, No. 5, pp.2341–2351.

Menezes, A.J., Oorschot, P.C.V. and Vanstone, S.A. (1996) *Handbook of Applied Cryptography, Chapter 9 Hash Functions and Data Integrity*, CRC Press, Canada.

Ou, H.H., Lin, I.C., Hwang, M.S. and Jan, J.K. (2009) 'TK-AKA: Using temporary key on authentication and key agreement protocol on UMTS', *International Journal of Network Management*, Vol. 19, pp.291–303.

Ohkubo, M., Suzuki, K. and Kinoshita, S. (2004) 'Efficient hash-chain based RFID privacy protection scheme', *Proceedings of the Sixth International Conference on Ubiquitous Computing (UbiComp'04)*, Nottingham, England, pp.1–5.

Pei, J.P. (2009) *A Study of Counter Inconsistency Problems in RFID Authentication Protocols*, Master Thesis, pp.1–49.

Rizomiliotis, P., Rekleitis, E. and Gritzalis, S. (2009) 'Security analysis of the song-mitchell authentication protocol for low-cost RFID tags', *IEEE Communications Letters*, Vol. 13, No. 4, pp.274–276.

Roberts, C.M. (2006) 'Radio frequency identification (RFID)', *Computers & Security*, Vol. 25, No. 1, pp.18–26.

Song, B. and Mitchell, C.J. (2008) 'RFID authentication protocol for low-cost tags', *Proceedings of the First ACM Conference on Wireless Network Security*, Alexandria, VA, USA, pp.140–147.

Yeh, T.C., Wu, C.H. and Tseng, Y.M. (2010) 'Improvement of the RFID authentication scheme based on quadratic residues', *Computer Communications*, Vol. 34, No. 3, pp.337–341.

Yeo, S.S., Kim, S.C. and Kim, S.K. (2009) 'Protecting your privacy with a mobile agent device in RFID environment', *Wireless Personal Communications*, Vol. 51, No. 1, pp.165–178.

Zhang, X. and King, B. (2008) 'Security requirements for RFID computing systems', *International Journal of Network Security*, Vol. 6, pp.214–226.

Nomenclature

h	A hast function
f_K	A keyed Hash Function
N	The number of tag
l	The bit-length of a tag identifier
N_T	The random number generated by the tag
N_R	The random number generated by the reader
u_i	A string of l bits assigned to T_i
t_i	T_i's l-bit identifier, which is $t_i = h(u_i)$
x_{new}	The new (refreshed) value of x
x_{old}	The most recent value of x
r	A random string of l bits
$\bigoplus$	XOR operator
$\leftarrow$	Substitution operator
$x >> y$	Right circular shift operator, rotates all bits of x to the right by y bits
$x << y$	Left circular shift operator, rotates all bits of x to the left by y bits
$\in_R$	The random choice operator, randomly selects an element from a finite set using a uniform probability distribution