A Novel RFID Authentication Protocol based on Elliptic Curve Cryptosystem

Yalin Chen[1], Jue-Sam Chou[2], Chi-Fong Lin[3], Cheng-Lun Wu[4]

[1] Institute of information systems and applications, National Tsing Hua University, Taiwan

yalin78900@gmail.com

[2,3,4] Department of Information Management, Nanhua University, Taiwan

jschou@mail.nhu.edu.tw, chejtmmc@gmail.com, wfdawu@gmail.com

Abstract

Recently, many researchers have proposed RFID authentication protocols. These protocols are mainly consists of two types: symmetric key based and asymmetric key based. The symmetric key based systems usually have some weaknesses such as suffering brute force, de-synchronization, impersonation, and tracing attacks. In addition, the asymmetric key based systems usually suffer from impersonation, man-in-the-middle, physical, and tracing attacks. To get rid of those weaknesses and reduce the system workload, we adopt elliptic curve cryptosystem (ECC) to construct an asymmetric key based RFID authentication system. Our scheme needs only two passes and can resist various kinds of attacks. It not only outperforms the other RFID schemes having the same security level but also is the most efficient.

Keywords: radio frequency identification, RFID, identification protocol, privacy, untraceability, location privacy, scalability, Elliptic Curve Cryptosystem,

1. Introduction

The barcode labels have been used for a long time. Although they are cheaper to apply, but not secure enough in some fields which needs more privacy and information protection. Another problem of the barcode labels is that they must be read by an optical reader in line-of-sight contact. These properties restrict their applications. Henceforth, Radio frequency identification (RFID) is developed to identify a specific target and transfer data by using radio signals without physical contacts between the reader and tags. The other characteristics of a RFID system include multiple reads at a time, easy and rapid read by the reader, and repeated use and high storage capacity of the tag. Because of these advantages, RFID systems are widely applied in many fields. Some of the most common applications are: electronic toll collection system, access management, animal identification, e-passport, medical applications, asset management, transportation and logistics management, etc [1, 5, 13].

According to whether equipped with a power supply, the tags can be categorized into three types:

(1). **Passive tag**: Due to lack of any power supply, the tag requires an external electric magnetic field to start a communication.

(2). **Semi-passive tag**: The tag is equipped with a power supply and needs the reader's signal to induce proper response.

(3). **Active tag**: With a battery, the tag can provide a wider range communication to communicate with the reader.

The most frequently used standard in RFID systems now is EPC (Electronic Product Code) global UHF Class 1 Generation 2. In 2004, the Class 1 Generation 2 interface was proposed by the Hardware Action Group to solve some vulnerabilities found in previous generations. The main property of EPC is that the tags conforming to the standard have the lowest cost [7]. Although EPC having this benefit, it still has some vulnerabilities to be overcome. For example, the privacy is sometimes more important than the other properties. If the applications of RFID systems violate privacy principle, like the personal information leakage or illegal tracing by a malicious person, it will keep us from applying them. To prevent this situation, a secure RFID protocol is usually embedded with authentication functions to protect the communication from an intentional adversary. To safely authenticate a tag's identity, some literatures pointed out that a RFID system should resist against the following attacks, as indicated in [16]:

- ✧ **Replay attack**: If an adversary *E* intercepts the information transmitted between the server and the tag. He can reuse the information to spoof the tag to be successfully authenticated by the server.
- ✧ **De-synchronization attack**: *E* sends spoofed messages to make the data stored in both the tags and the server de-synchronized. It can cause the communication between the tags and the server to be invalid temporarily or permanently.
- ✧ **Impersonation attack**: *E* utilizes the messages eavesdropped before to impersonate a legitimate tag (or server) to communicate with the server (or tag) and pass the authentication successfully.
- ✧ **Man-in-the-middle attack**: An active adversary modifies the transmitted messages between the tag and the server, making them believe that they are communicating to the intended party.
- ✧ **Physical attack**: An active adversary corrupts the tag and extracts the stored secrets, then uses those exposed secrets to launch various attacks on the other tags.

To prevent above mentioned attacks and protect the server and tags' privacy, many protocols [4, 8, 9] were proposed. However, they have been proved insecure enough, as indicated in [2, 3, 5, 13]. Although some improved protocols [3, 5, 11, 15] had been proposed subsequently; nevertheless in the following, we will show that there still exist some vulnerabilities in these improved protocols.

In 2007, Chien and Chen [4] proposed a RFID authentication protocol conforming to EPC Class 1 Generation 2 standard. They claimed that their protocol is secure against all possible attacks. However, in 2009, Han and Kwon [2] found that Chien and Chen's protocol is vulnerable to both the impersonate attack and de-synchronization attack. In 2010, Yeh *et al.* [15] proposed an improvement on Chien and Chen's protocol. Although their protocol can overcome the de-synchronization attack and be more efficient than Chien and Chen's protocol, it can not achieve the privacy property as they claimed. Because in their improvement, whenever the server sends a random number to the tag, the tag will respond with a message containing value C_i (used as an index) to the server for finding the corresponding record in the server's database with C_i kept unchanged until the tag is authenticated by the server successfully. This means if an adversary masquerades as the server by sending a request message N_R to the tag, the tag will reveal his C_i. But the adversary cannot respond with a correct message *M2* as the server does in the protocol, this makes C_i kept unchanged in the tag. Hence, the tag can be traced by an adversary next time when it responds to the server's request with parameters including value C_i.

In 2008, Burmester *et al.* [8] proposed a mutual authentication RFID protocol, (TRAP-3) which is compatible with the EPC Class2 Gen2 standard, to provide strong anonymity. But in 2010, Yeh *et al.* [5] found the protocol suffers the de-synchronization attack and hence proposed an improvement to modify the key updating mechanism. However, it also has the same vulnerability, suffering the de-synchronization attack. We will show the vulnerability in Section 3.1. In 2009, Peris-Lopez *et al.* [9] proposed a Gossamer RFID protocol to prevent Dos attack. But, in 2010, Tagra *et al.* [3] and Yeh *et al.* [5] both showed that the protocol suffers the de-synchronization DoS attack. They each proposed an improved protocol, and claimed that their improvements can successfully avoid the vulnerability. However, we found the pseudonym used is kept unchanged before a successful mutual authentication between the server and the tag. This means their protocols still suffer from the tracing attacks. This is because if an adversary sends a hello message to a tag, the tag will respond with its pseudonym. By this way, the adversary can easily distinguish two different tags and thus trace a tag. In 2010, Deng *et al.* [11] proposed an efficient RFID mutual authentication protocol. They claimed their protocol can avoid the de-synchronization attack. But we found it can not attain this goal. We will show the attack in Section 3.1. In 2011, Song *et al.* [21] proposed a scalable RFID security protocols supporting tag ownership transfer and claimed their protocol can avoid the de-synchronization attack. However, we found it also suffers from the de-synchronization attack. We will show this in Section 3.1 as well.

From the above mentioned, we can see that the proposed protocols using symmetric key cryptography although cost less but usually can not achieve the demanding security requirements of a RFID system. By using brute-force search, the particular tag can be found in almost all of those protocols. Moreover, symmetric key algorithm has the intrinsic shortcoming, lacking the scalability. About this, public key cryptosystem (PKC) seems a useful solution [6, 12, 14]. In PKC, elliptic curve cryptosystem (ECC) can provide the same security level with shorter keys. This makes ECC a suitable public key cryptosystem to be applied in RFID systems which has less powerful device such as tags [10]. Hence in this paper, we will base on ECC to propose a novel RFID authentication protocol.

The rest of this paper is organized as follows. In Section 2, the background and underline of our scheme, Theorem 1, are introduced. After that, some public key RFID protocols are reviewed in Section 3. The proposed protocol is presented In Section 4, and the security analyses and comparisons with other schemes are demonstrated in Section 5. The discussion is shown in Section 6 and finally a conclusion is given in Section 7.

2. Background

In this section, we give the definitions of the elliptic curve cryptography and demonstrate the underline of our scheme, Theorem 1, in Sec. 2.1 and Sec. 2.2, respectively.

2.1 Elliptic Curve Cryptography

In 1985 and 1987, Victor S. Miller and Neal Koblitz independently proposed the concepts of ECC [23]. Below, we roughly introduce ECC and Elliptic Curve Discrete Logarithm Problem [23, 25].

ECC

Suppose a and b are two field elements that define the curve of the equation $y^2 = x^3 + ax + b$. All points (x, y) satisfying the elliptic curve equation along with an infinite point O and an addition operation form a group G. The elliptic curve has the following properties:

- ✧ Suppose $P = (x, y)$, then define $–P = (x, -y)$.

- ✧ If *P* and *Q* are distinct, define $P \neq -Q$ and $P + (-P) = O$.
- ✧ If $P = (x,0)$, then $P+P=O$. Otherwise, draw a tangent line through *P*, the intersected point is defined as $-R$, then $P + P = 2P = R$.

ECDLP

If *P* is a base point of group *G*, *n* is a prime and is the order of *G*, and there is a point *Q* in *G*. To find the integer $l \in [0, n-1]$ such that $Q = lP$, is called an Elliptic Curve Discrete Logarithm Problem (ECDLP).

2.2 The underline of our scheme

From Diophantine equation [24], we have the following theorem.

Theorem 1: Given B_1, B_2, x_1, x_2 such that $B_1 = x_1A$ and $B_2 = x_2A$, where *A* is an element of G. Then, *A* can be easily computed if $gcd(x_1, x_2) = 1$.

Proof: Since $gcd(x_1, x_2)=1$, from Euclidean algorithm [24], we can find a pair of (k_1, k_2) satisfying $k_1x_1 + k_2x_2 = 1$. Therefore, we have $A = (k_1x_1+k_2x_2) A = k_1x_1A + k_2x_2A = k_1B_1 + k_2B_2$.

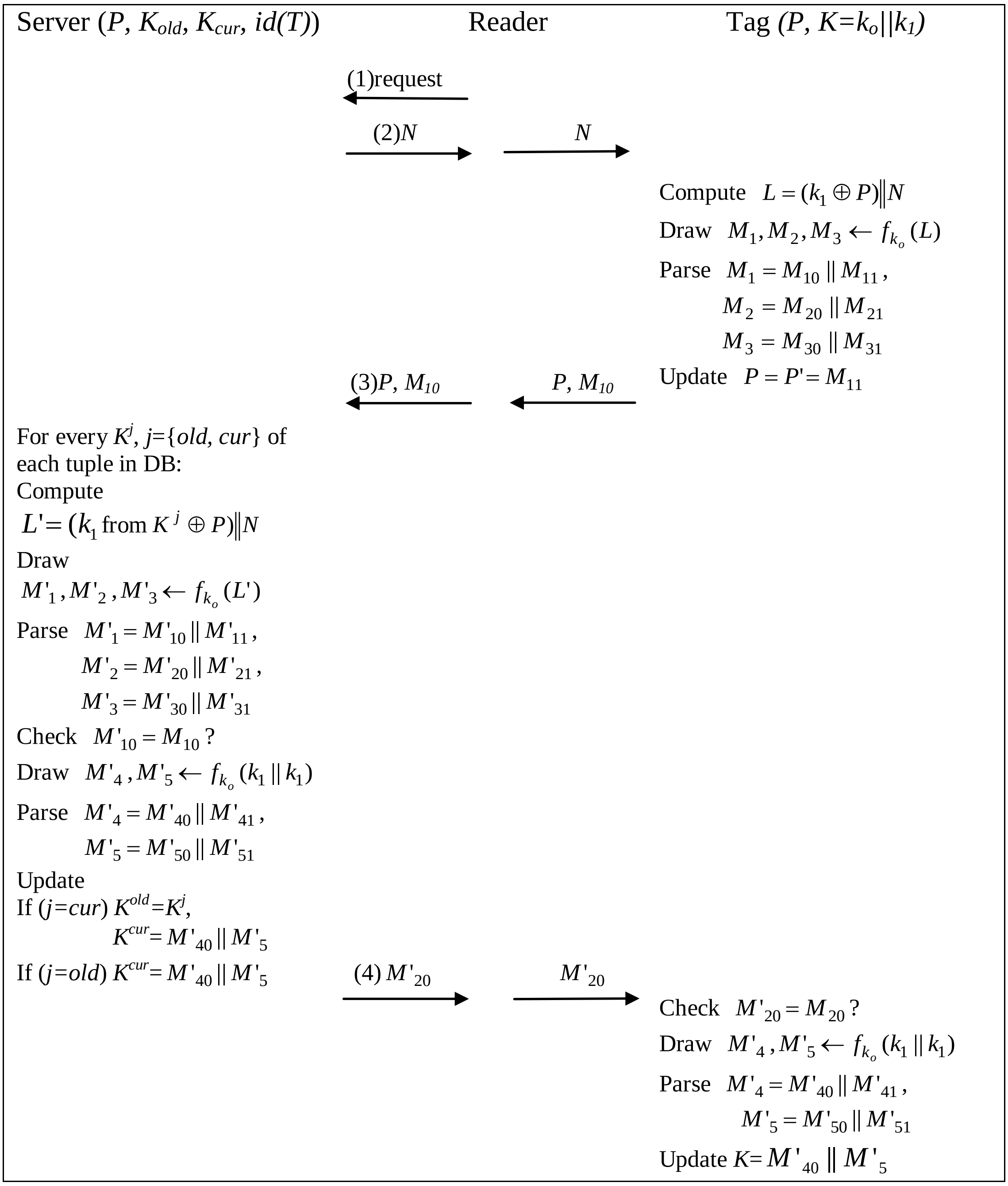
3. Review of some RFID authentication protocols

In this section, we review some RFID authentication protocols. We classify these RFID authentication protocols into two types: (a) non PKC-based RFID, and (b) PKC-based RFID. Below, in Section 3.1 and 3.2, we review these two types of protocols and show they each has some vulnerabilities.

3.1 Review of some non PKC-based RFID authentication protocols

In 2010, Yeh *et al.* [5] found TRAP-3 [22] is not secure and proposed a countermeasure. They modified the key updating mechanism, intending to solve the de-synchronization attack. However, we found their improvement still suffers the de-synchronization attack. We depict their improved scheme in Fig.1.(1) and demonstrate the attack in Fig.1.(2) through (4).

In Fig.1.(1), if there exists an adversary *E* and two legal readers, R_A and R_B, we can show the attack scenarios using Fig.1.(2) through (4). In Fig.1.(2), time T_0, a (server, tag) pair communicate with R_A in session A. Suppose *E* intercepts M'_{20A} and suspends the session. *E* then waits until time T_1, when the same (server, tag) pair communicate with another legal reader R_B in session B, he can launch the same attack by intercepting M'_{20B} and abandoning session B. After that, E resumes session A at T_2 and sends M'_{20A} to the tag. As a result, the keys stored in both the server ($K^{old}=K^j=k_0||k_1$, $K^{cur}=M'_{40B} \| M'_{5B}$) and the tag ($K=M'_{40A} \| M'_{5A}$) are different. The detail corresponding parameters according to the protocol are shown in Fig.1.(2) through (4). So, the improvement still suffers from the de-synchronization attack.



Tag:

Compute $L=(k_1 \oplus P)\|N$

Draw $M_1, M_2, M_3 \leftarrow f_{k_o}(L)$

Parse $M_1 = M_{10} \,||\, M_{11}$,
$M_2 = M_{20} \,||\, M_{21}$
$M_3 = M_{30} \,||\, M_{31}$

Update $P = P' = M_{11}$

Server:

For every K^j, $j=\{old, cur\}$ of each tuple in DB:
Compute
$L' = (k_1 \text{ from } K^j \oplus P)\|N$
Draw
$M'_1, M'_2, M'_3 \leftarrow f_{k_o}(L')$

Parse $M'_1 = M'_{10} \,||\, M'_{11}$,
$M'_2 = M'_{20} \,||\, M'_{21}$,
$M'_3 = M'_{30} \,||\, M'_{31}$

Check $M'_{10} = M_{10}$?

Draw $M'_4, M'_5 \leftarrow f_{k_o}(k_1 \,||\, k_1)$

Parse $M'_4 = M'_{40} \,||\, M'_{41}$,
$M'_5 = M'_{50} \,||\, M'_{51}$

Update
If ($j=cur$) $K^{old}=K^j$,
$K^{cur}= M'_{40} \,||\, M'_5$
If ($j=old$) $K^{cur}= M'_{40} \,||\, M'_5$

Tag:

Check $M'_{20} = M_{20}$?

Draw $M'_4, M'_5 \leftarrow f_{k_o}(k_1 \,||\, k_1)$

Parse $M'_4 = M'_{40} \,||\, M'_{41}$,
$M'_5 = M'_{50} \,||\, M'_{51}$

Update $K= M'_{40} \,||\, M'_5$

Fig.1.(1) Yeh's improvement of TRAP-3

In 2010, Deng *et al.* [11] proposed an efficient RFID mutual authentication protocol and claimed it can avoid the de-synchronization attack. However, after analyses, we found it can not avoid another kind of de-synchronization attack. We depict their scheme in Fig.2.(1) and show the attack in Fig.2.(2) though (3).

In their protocol, tag T_i and database both store the value *ctr* as a counter. If a malicious reader R_E broadcasts a challenge string C_A, then all the *n* tags, need to update their counters as $ctr_{i\text{A}}= ctr_i + 1$ (for i=1 to n). Then, R_E abandons the protocol, as shown in Fig.2.(2). This makes the counter value ctr_i stored in each tag different

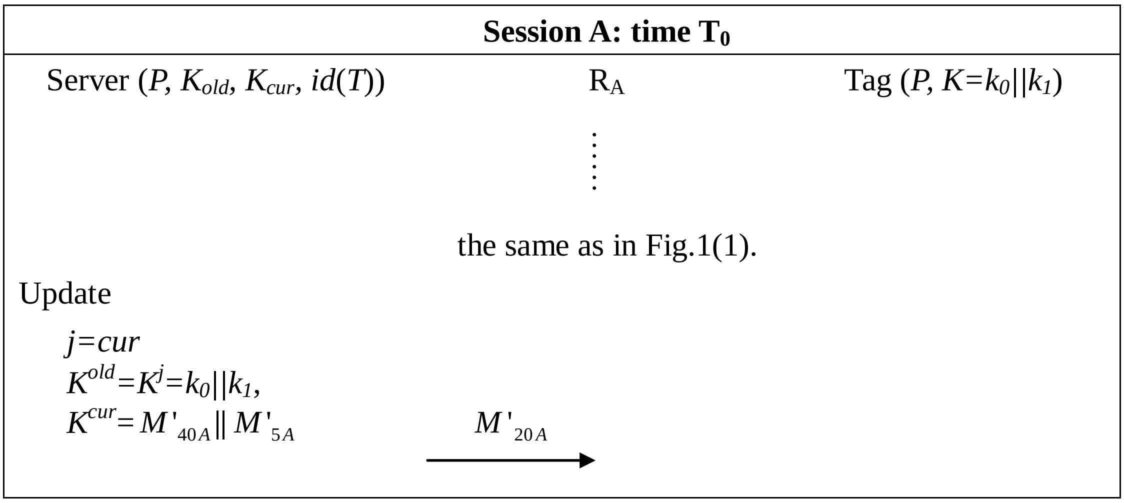


Fig.1.(2). E intercepts M'_{20A} and suspends this session

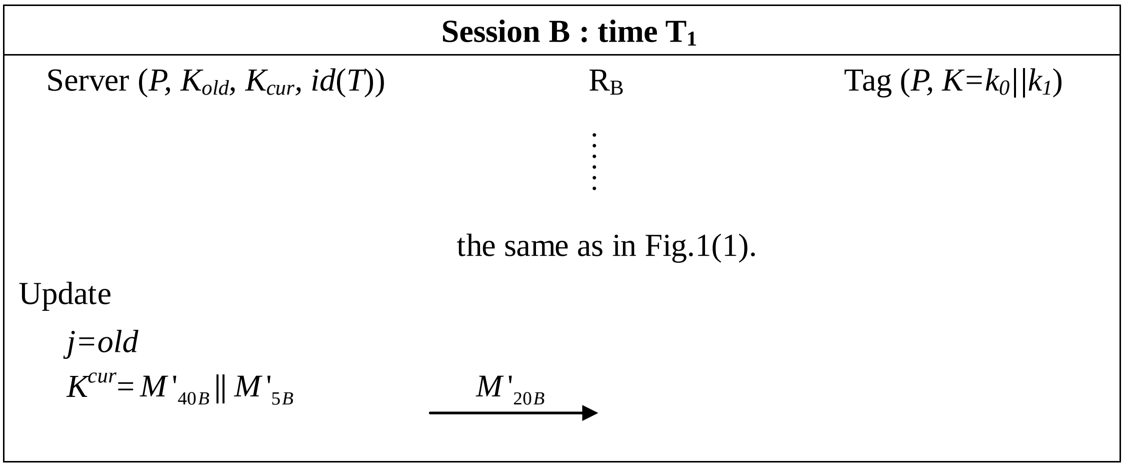


Fig.1.(3). E intercepts M'_{20B} and abandons this session

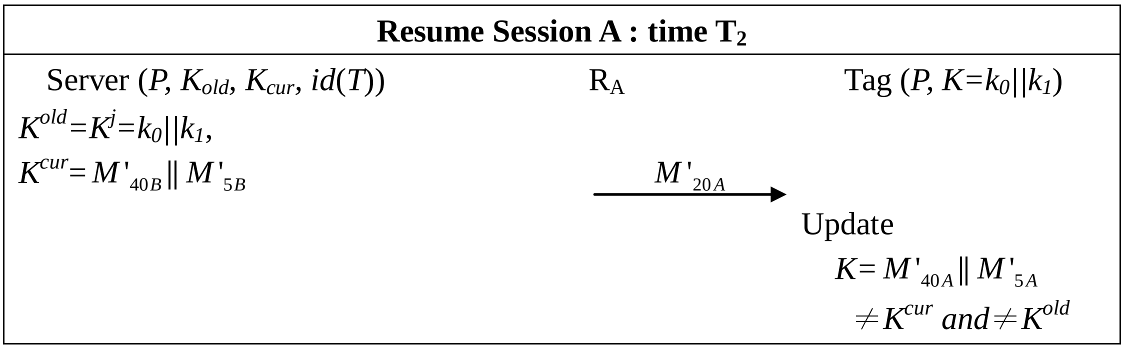

Resume Session A : time T_2
Server (P, K_{old}, K_{cur}, $id(T)$)
R_A
Tag (P, $K=k_0||k_1$)
$K^{old}=K^j=k_0||k_1$,
$K^{cur}=M'_{40B}\,||\,M'_{5B}$
M'_{20A}
Update
$K=M'_{40A}\,||\,M'_{5A}$
$\neq K^{cur}$ and $\neq K^{old}$

Fig.1.(4). E sends M'_{20A} to Tag

from the one stored in the server's database and thus incurs the de-synchronized problem. Thereafter, when the legitimate reader R broadcasts C_B to communicate with all the n tags, T_i (for I = 1 to n) will compute index $I_{iB}=F_k^0(ctr_{iA}\,||\,pad1)$ using the renewed ctr_{iA} in Session A and send message I_{iB} and r_{iTB} to the reader. After receiving

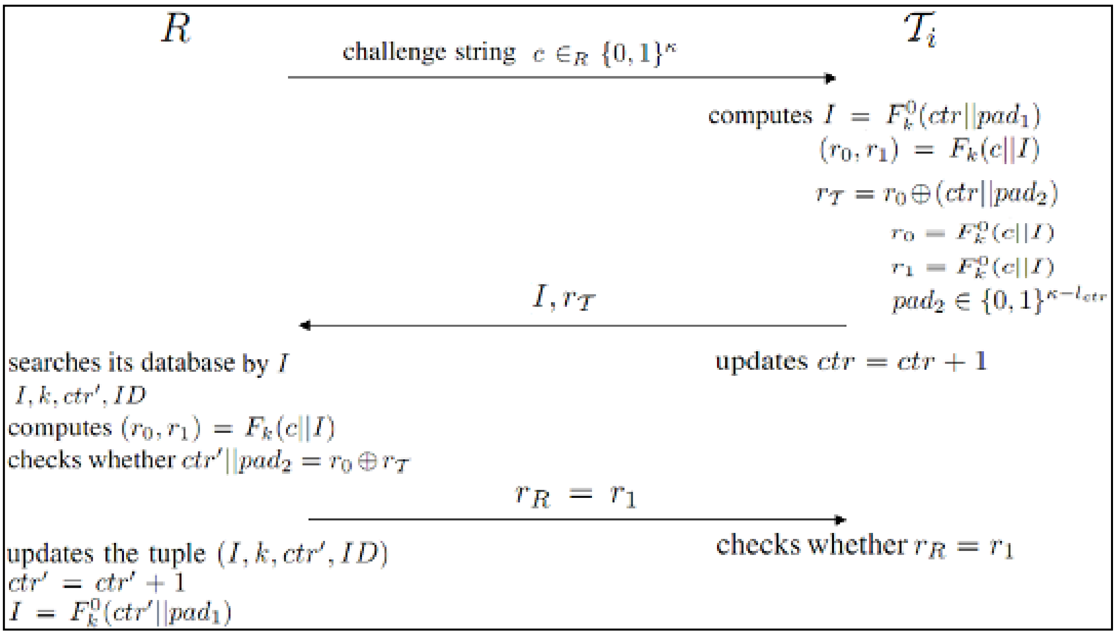

R
$\mathcal{T}_i$
challenge string $c \in_R \{0,1\}^\kappa$
computes $I = F_k^0(ctr||pad_1)$
$(r_0, r_1) = F_k(c||I)$
$r_\mathcal{T} = r_0 \oplus (ctr||pad_2)$
$r_0 = F_k^0(c||I)$
$r_1 = F_k^0(c||I)$
$pad_2 \in \{0,1\}^{\kappa - l_{ctr}}$
$I, r_\mathcal{T}$
updates $ctr = ctr + 1$
searches its database by I
I, k, ctr', ID
computes $(r_0, r_1) = F_k(c||I)$
checks whether $ctr'||pad_2 = r_0 \oplus r_\mathcal{T}$
$r_R = r_1$
checks whether $r_R = r_1$
updates the tuple (I, k, ctr', ID)
$ctr' = ctr' + 1$
$I = F_k^0(ctr'||pad_1)$

Fig.2.(1). Deng's RFID mutual authentication protocol

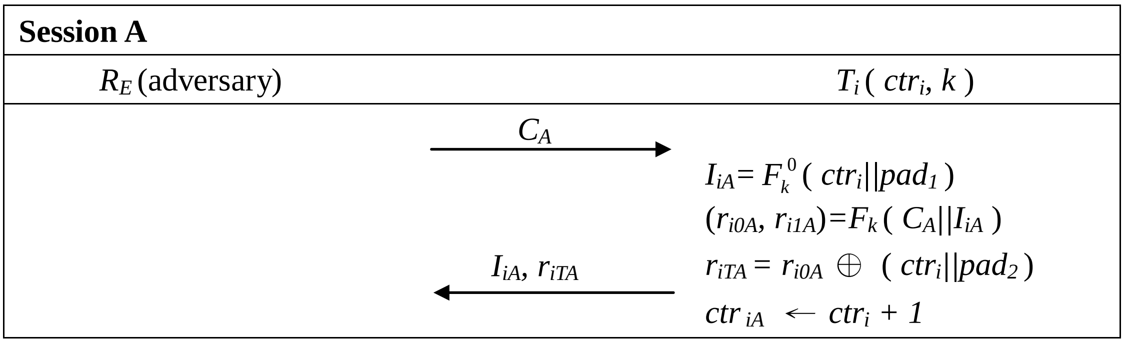

Session A
R_E (adversary)
T_i (ctr_i, k)
C_A
$I_{iA}= F_k^0$ ($ctr_i||pad_1$)
(r_{i0A}, r_{i1A})=F_k ($C_A||I_{iA}$)
I_{iA}, r_{iTA}
$r_{iTA} = r_{i0A} \oplus$ ($ctr_i||pad_2$)
$ctr_{iA} \leftarrow ctr_i + 1$

Fig.2.(2). ***E*** **intercepts message** I_{iA}**,** r_{iTA} **and abandons this protocol run**

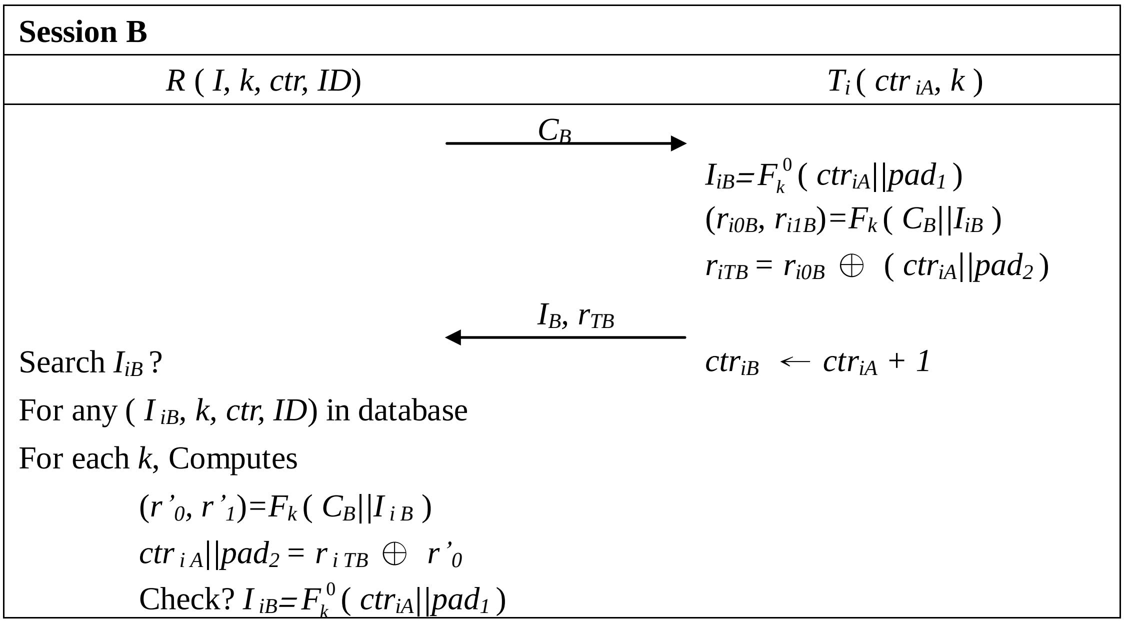
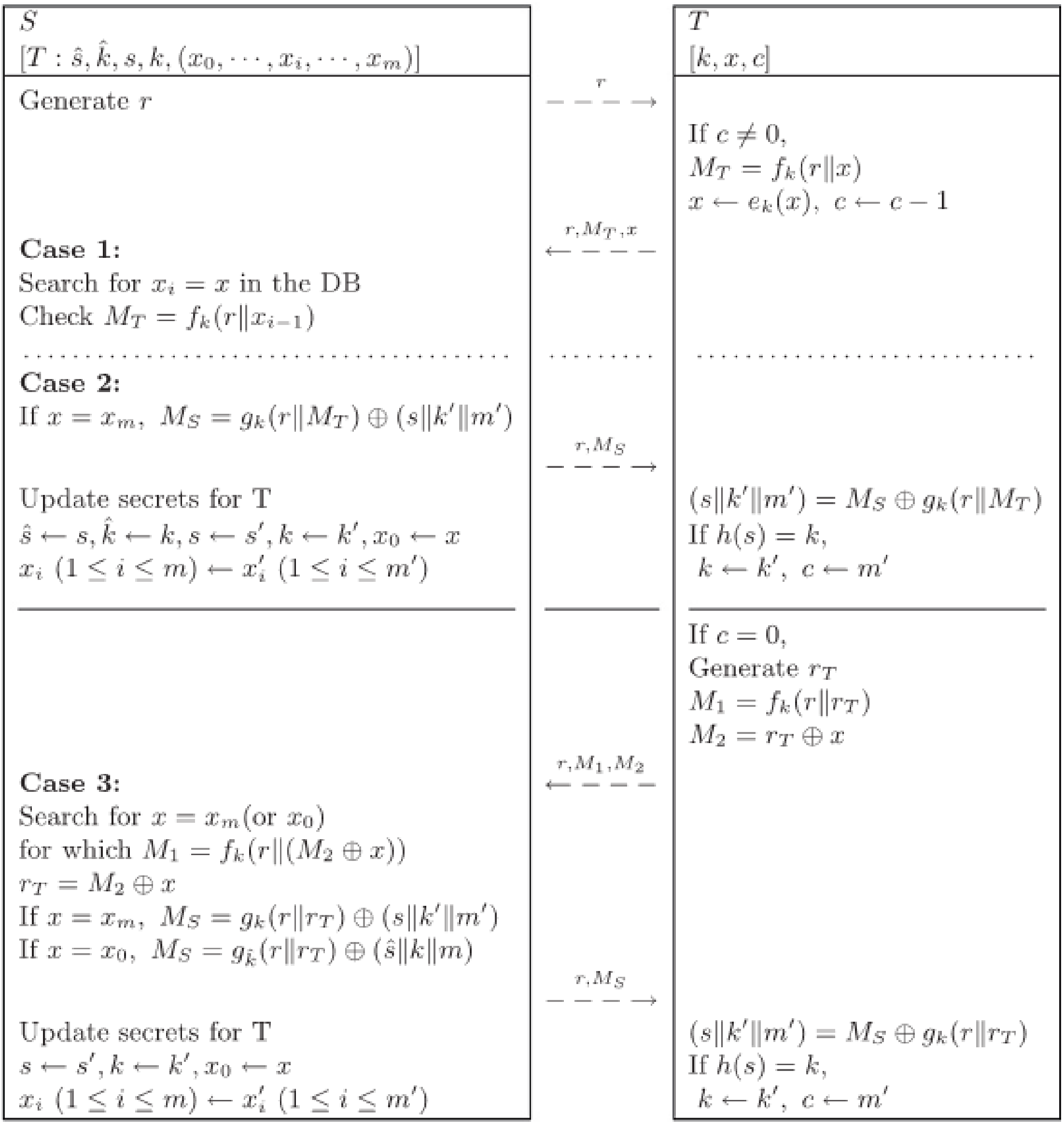

Session B
R (I, k, ctr, ID)
T_i (ctr_{iA}, k)
C_B
$I_{iB}=F_k^0$ ($ctr_{iA}||pad_1$)
(r_{i0B}, r_{i1B})=F_k ($C_B||I_{iB}$)
$r_{iTB} = r_{i0B} \oplus$ ($ctr_{iA}||pad_2$)
I_B, r_{TB}
Search I_{iB} ?
$ctr_{iB} \leftarrow ctr_{iA} + 1$
For any (I_{iB}, k, ctr, ID) in database
For each k, Computes
(r'_0, r'_1)=F_k ($C_B||I_{iB}$)
$ctr_{iA}||pad_2 = r_{iTB} \oplus r'_0$
Check? $I_{iB}=F_k^0$ ($ctr_{iA}||pad_1$)

Fig.2.(3). When legal R lunches session B, the counter values in both sides are different

the message, due to de-synchronization, the reader *R* can not find the corresponding value I_{iB} in server's database. It must use each tuple (*I*, *k*, *ctr*, *ID*) to compute $(r'_0, r'_1) = F_k(C_B||I_{iB})$ and then compute $r_{iTB} \oplus r'_0$ to find $ctr_{iA}||pad_2$. That is, on average, for each tag the server needs (1/2)(1+*n*) operations, with each operation including two computations and one verification to find out the right tag and its corresponding counter value. This not only makes the system lack of scalability, but also suffers from the de-synchronized attack. This is because the value *ctr* for T_i in server's side is ctr_i +1 rather than ctr_{iB}.

In 2011, Song *et al.* [21] proposed a scalable RFID security protocols to support tag ownership transfer, and claimed that their protocol can avoid the de-synchronization attack. Unfortunately, we found it hasn't the de-synchronization attack avoidance. We depict their scheme in Fig. 3 and show the attack as follows.



S
$[T : \hat{s}, \hat{k}, s, k, (x_0, \cdots, x_i, \cdots, x_m)]$
T
$[k, x, c]$
Generate r
r
If $c \neq 0$,
$M_T = f_k(r||x)$
$x \leftarrow e_k(x),\ c \leftarrow c - 1$
r, M_T, x
Case 1:
Search for $x_i = x$ in the DB
Check $M_T = f_k(r||x_{i-1})$
Case 2:
If $x = x_m,\ M_S = g_k(r||M_T) \oplus (s||k'||m')$
r, M_S
Update secrets for T
$\hat{s} \leftarrow s, \hat{k} \leftarrow k, s \leftarrow s', k \leftarrow k', x_0 \leftarrow x$
$x_i\ (1 \leq i \leq m) \leftarrow x'_i\ (1 \leq i \leq m')$
$(s||k'||m') = M_S \oplus g_k(r||M_T)$
If $h(s) = k$,
$k \leftarrow k',\ c \leftarrow m'$
If $c = 0$,
Generate r_T
$M_1 = f_k(r||r_T)$
$M_2 = r_T \oplus x$
r, M_1, M_2
Case 3:
Search for $x = x_m$(or x_0)
for which $M_1 = f_k(r||(M_2 \oplus x))$
$r_T = M_2 \oplus x$
If $x = x_m,\ M_S = g_k(r||r_T) \oplus (s||k'||m')$
If $x = x_0,\ M_S = g_{\hat{k}}(r||r_T) \oplus (\hat{s}||k||m)$
r, M_S
Update secrets for T
$s \leftarrow s', k \leftarrow k', x_0 \leftarrow x$
$x_i\ (1 \leq i \leq m) \leftarrow x'_i\ (1 \leq i \leq m')$
$(s||k'||m') = M_S \oplus g_k(r||r_T)$
If $h(s) = k$,
$k \leftarrow k',\ c \leftarrow m'$

Fig. 3. Song's RFID authentication and secret update protocol

In cases 2 and 3 (secret update), when server *S* sends (*r*, *M_s*) to tag *T*, where *r* is a random number generated by *S* and $M_s = g_k(r \| r_T) \oplus (s \| k' \| m')$ is a $(2l+|m'|)$-bit string. On receiving (*r*, M_s), *T* should compute $(s \| k' \| m') = M_s \oplus g_k(r \| r_T)$ and check if $h(s)=k$ holds. If so, *T* updates its key *k* to k' and counter *c* to m'. However, if the adversary modifies the second *l* bits in M_s string, obtaining M_s' and sends M_s' to *T*. When *T* utilizes the receiving M_s' to compute $(s \| k' \| m')$, the second *l* bits in the computation result will be different from the value k' that *S* has, Thereafter, when using it to update his key, *T* will have a different key from k'. So, their protocol can not avoid the de-synchronization attack.

3.2 Review of some PKC-based RFID authentication protocols

In 2006, Tuyls et al. [10] proposed a Schnorr identification RFID protocol based on ECDLP.

We depict their Schnorr identification protocol in Fig. 4 and describe the interactions between the Prover-Tag and Verifier-Reader as follows:

- **Commitment by a Prover-Tag**: The tag picks a random number *r* and sends $X=rP$ to the reader.
- **Challenge from a Verifier-Reader**: After receiving *X*, the reader picks a random number *e* and sends it to the tag.
- **Response from a Tag**: After receiving *e*, the tag computes $y = ae + r$ and sends *y* to the reader. Upon receiving *y*, the reader computes $yP + eZ$ and checks if it is equal to *X*. If it is, the reader accepts.

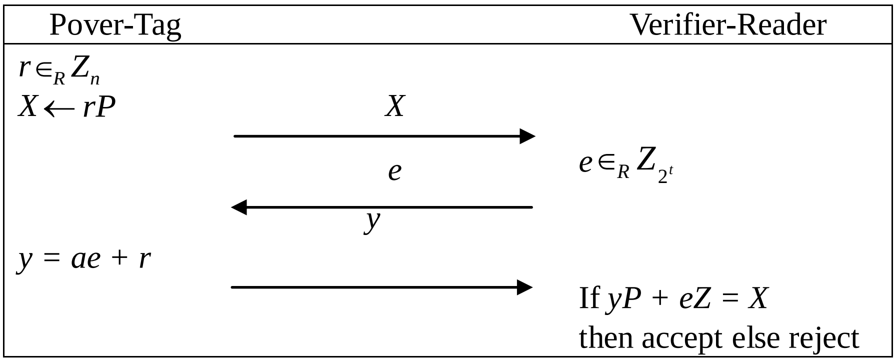
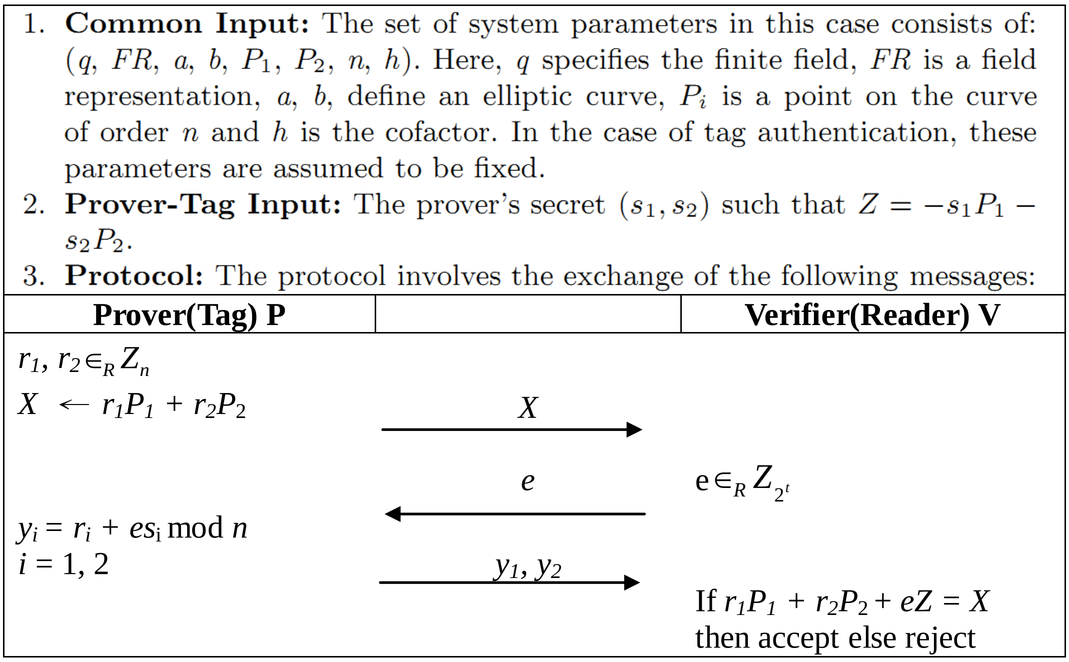


Fig. 4. Schnorr identification protocol

Although their protocol can prevent counterfeiting, we found it suffers from the tracing attack. The attack is shown as follows. Suppose an adversary eavesdrops on the communication between a specific tag and reader. He learns X_1 (= r_1P), *e*, and y_1 (= $ae+r_1$). If the adversary wants to trace this specific tag, he can pretend a legitimate reader to communicate with it. Once having received X_2 (= r_2P) from the tag in a protocol run, he sends *Challenge e'* (=*e*) back to the tag, and obtains y_2 (= ae+r_2). By doing this, the adversary can easily trace the tag by checking whether (y_2 - y_1)*P* equals to X_2 - X_1. The other problem of this protocol is that it lacks the forward secrecy. Because if an adversary obtains the prover's public key *Z*, he can know the messages *X* sent before and thus can trace the tag (prover) since *yp* + *eZ* = *X*. Besides, it lacks of scalability. When the reader receives the response from a certain tag, the reader can not judge from which tag the message is sent. Put it another way, the reader must use each tag's public keys *Z*s in its database for computing *yP* + *eZ* to compare with *X*. This means when the reader wants to communicate with larger numbers of tags, the reader needs more computations to identify the tag. This causes their protocol lack of scalability.

In 2007, Batina *et al.* [6] proposed Okamoto's identification RFID protocol based on ECDLP. We depict the Okamoto's identification protocol in Fig. 5 and describe the interactions between the Prover (Tag) and Verifier (Reader) as follows:

- ✧ **Commitment**: The tag (Prover P) picks two random numbers r_1, r_2 and computes X=r_1P_1+r_2P_2. Then, it sends *X* to the reader (Verifier V).
- ✧ **Challenge***:* Upon receiving *X*, the reader picks a random number *e* and sends back it to the tag.
- ✧ **Response***:* After receiving *e*, The tag computes $y_1 = r_1 + e \cdot s_1$ and $y_2 = r_2 + e \cdot s_2$, and sends them to the reader. The reader then computes $y_1P_1 + y_2P_2 + e \cdot Z$ to check if it is equal to *X*. If it is, the reader accepts.



1. **Common Input:** The set of system parameters in this case consists of: $(q, FR, a, b, P_1, P_2, n, h)$. Here, q specifies the finite field, FR is a field representation, a, b, define an elliptic curve, P_i is a point on the curve of order n and h is the cofactor. In the case of tag authentication, these parameters are assumed to be fixed.
2. **Prover-Tag Input:** The prover's secret (s_1, s_2) such that $Z = -s_1P_1 - s_2P_2$.
3. **Protocol:** The protocol involves the exchange of the following messages:

Prover(Tag) P		Verifier(Reader) V
$r_1, r_2 \in_R Z_n$		
$X \leftarrow r_1P_1 + r_2P_2$	$\xrightarrow{X}$	
	$\xleftarrow{e}$	$e \in_R Z_{2^t}$
$y_i = r_i + es_i \bmod n$ $i = 1, 2$	$\xrightarrow{y_1, y_2}$	
		If $r_1P_1 + r_2P_2 + eZ = X$ then accept else reject

Fig. 5. Okamoto's identification protocol

The authors consider that their identification protocol can protect against active adversaries. But we found it can not achieve the forward secrecy and thus is traceable. Since P_1, P_2 are system public parameters, and X, e, y_1 and y_2 are publicly transferred. If an adversary eavesdrops on one round of communication between the tag and reader, he can obtain value $e \cdot Z$ by computing $X - y_1P_1 - y_2P_2$. Thereafter, when the tag sends another commitment X', the adversary can impersonate the reader to communicate with the tag by sending challenge $e'=e+1$ to the tag. After receiving the responses y'_1 and y'_2 from the tag, the adversary can obtain the value $e' \cdot Z$ by computing $X' - y'_1P_1 - y'_2P_2$. Then, he can compute $Z = e'Z - eZ$. Thus, although X is randomized by r_1 and r_2 which are unknown to the adversary, the adversary can use the computed Z to find the tag. That is, if the adversary utilizes value Z to check all the messages (X, e, (y_1, y_2)) he eavesdropped before, he can easily identify the right tag. This incurs the tag to be traced by the adversary.

In 2008, Lee *et al.* [18] proposed EC-RAC protocol to resist against such tracing attacks. But in 2009, Lee *et al.* [19] found that EC-RAC in [18] still suffers both tracing attacks and replay attacks and further proposed a revised EC-RAC protocol, denoted as EC-RAC II, to avoid those vulnerabilities. In 2010, Lee *et al.* [17] found EC-RAC II suffers the man-in-the-middle attack. They proposed a solution denoted as EC-RAC IV to solve this deficiency. We depict EC-RAC IV in Fig. 6 and roughly describe it using the following steps.

Step 1: Tag generates a random number r_{t1}, computes $T_1 = r_{t1}P$, and sends message T_1 to the server.

Step 2: Upon receiving T_1, the server generates and sends r_{s1} to the tag. Then the server computes $\dot{r}_{s1} = x(r_{s1}P)$, where $x(r_{s1}P)$ is the x-coordinate of $r_{s1}P$.

Step 3: When receiving challenge r_{s1} from the server, the tag computes an authentication message T_2=$(r_{t1} + \dot{r}_{s1}x_1)Y$ and sends T_2 to the server.

Step 4: After receiving T_2, the server computes $(y^{-1}T_2 - T_1)\dot{r}_{s1}^{-1}$ to obtain the tag's identifier (x_1P) and checks to see if it is in the server's database.

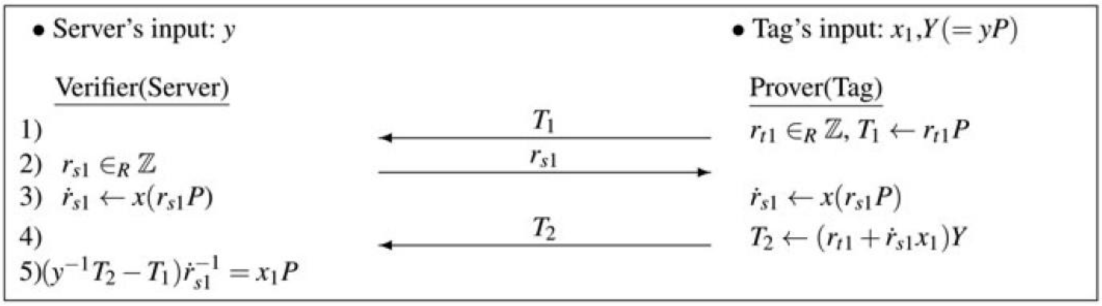

• Server's input: y
• Tag's input: $x_1, Y(=yP)$
Verifier(Server)
Prover(Tag)
1)
T_1
$r_{t1} \in_R \mathbb{Z}$, $T_1 \leftarrow r_{t1}P$
2) $r_{s1} \in_R \mathbb{Z}$
r_{s1}
3) $\dot{r}_{s1} \leftarrow x(r_{s1}P)$
$\dot{r}_{s1} \leftarrow x(r_{s1}P)$
4)
T_2
$T_2 \leftarrow (r_{t1} + \dot{r}_{s1}x_1)Y$
5)$(y^{-1}T_2 - T_1)\dot{r}_{s1}^{-1} = x_1P$

Fig. 6. EC-RAC IV

Basically, we haven't found any flaw in their scheme yet. But, we found if it is further reduced to a two-pass scheme (to be comparable with our scheme) for efficiency consideration, as shown in Fig. 7, the reduced protocol will suffer from tag impersonation attack. We demonstrate this attack in the following. First, if an adversary eavesdrops on two successful protocol runs between the server and a specific tag, obtaining the values of $\left(r_{s1},\left(T_1,T_2\right)\right)$ and $\left(r_{s1}',\left(T_1',T_2'\right)\right)$. Since T_1 and T_1' are equal to $r_{t1}P$ and $r_{t1}'P$, and T_2 and T_2' are equal to $(r_{t1}+\dot{r}_{s1}x_1)Y$ and $(r_{t1}'+\dot{r}_{s1}'x_1)Y$, respectively. The adversary can compute and obtain the value $r_{s1}''=r_{s1}'-r_{s1}$, $\dot{r}_{s1}''=x(r_{s1}''P)$, $T_1''=T_1'-T_1=(r_{t1}'-r_{t1})P$ and $T_2''=T_2'-T_2=(r_{t1}'+\dot{r}_{s1}'x_1)Y-(r_{t1}+\dot{r}_{s1}x_1)Y=(r_{t1}'-r_{t1})Y+(\dot{r}_{s1}'-\dot{r}_{s1})x_1Y$. We found that T_1'' and T_2'' will satisfy $(y^{-1}T_2''-T_1'')\cdot\dot{r}_{s1}''^{-1}=((r_{t1}'-r_{t1})P+(\dot{r}_{s1}'-\dot{r}_{s1})x_1P-(r_{t1}'-r_{t1})P)\cdot\dot{r}_{s1}''^{-1}=(\dot{r}_{s1}'-\dot{r}_{s1})x_1P\cdot(\dot{r}_{s1}'-\dot{r}_{s1})^{-1}=x_1P$. This means by using the tuple $(\dot{r}_{s1}'',T_1'',T_2'')$, the tag will be successfully authenticated by the server. Under this way, the adversary can generate more and more legitimate authentication messages at his will to impersonate any specific tag successfully. Therefore, the attempt to reduce EC-RAC IV to fewer passes fails.

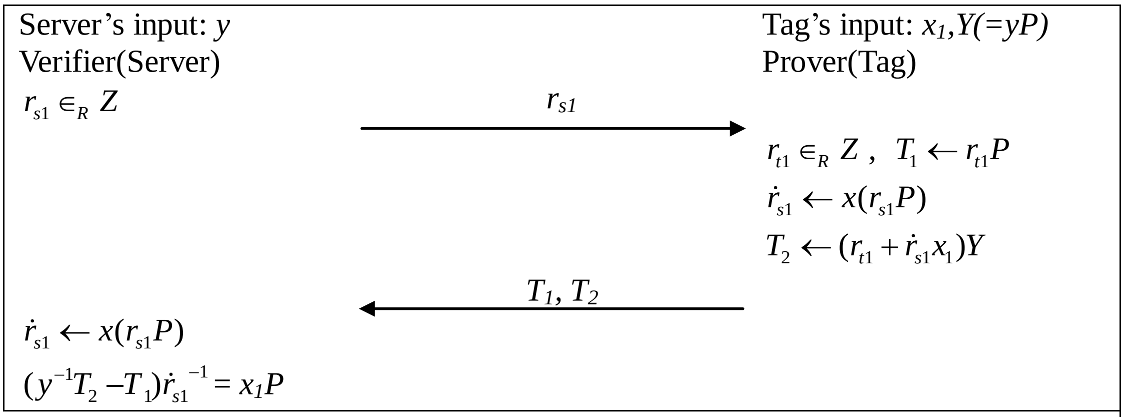

Server's input: y
Verifier(Server)
$r_{s1}\in_R Z$
r_{s1}
Tag's input: $x_1,Y(=yP)$
Prover(Tag)
$r_{t1}\in_R Z$, $T_1\leftarrow r_{t1}P$
$\dot{r}_{s1}\leftarrow x(r_{s1}P)$
$T_2\leftarrow(r_{t1}+\dot{r}_{s1}x_1)Y$
T_1, T_2
$\dot{r}_{s1}\leftarrow x(r_{s1}P)$
$(y^{-1}T_2-T_1)\dot{r}_{s1}^{-1}=x_1P$

Fig. 7. EC-RAC IV(reduced version)

From the above mentioned, we see that up to date RFID protocols based on ECC either flawed or need at least 3 passes, we therefore propose a novel secure ECC based RFID protocol which needs only two passes.

4. The proposed scheme

As in EC-RAC IV, there are only two participants in the proposed scheme, namely the server and the tag. The scheme consists of two phases: (1) initialization phase, and (2) authentication phase. We describe the two phases in Section 4.1 and Section 4.2, respectively. Before describing it, we define some used notations.

G: a group of order *q* on an elliptic curve,

P: a primitive element of G,

d_i: tag$_i$'s private key,

ID: tag's identify,

s: server's private key,

Y: server's public key,

h: an one-way hash function mapping from $G \times G$ to Z_q,

H: an one-way hash function mapping from $\{0, 1\}^*$ to G,

t_s, t_i: two timestamps,

r_s, k: two random numbers in Z_q,

R_i: a random element in G,

4.1 Initialization phase

In this phase, server S generates a random number s and computes $Y=sP$. It then sets s/Y as his private/public key pair. Then, S generates a random number d_i as Tag_i's private key, and computes $ID_i=d_iP$ for every tag. After that, S distributes ID_i, P, Y and t_i to Tag_i over a secure channel, where t_i is a timestamp.

4.2 Authentication phase

In this phase, if server S wants to anonymously authenticate Tag_i to access some stored information. They together perform the following steps which also depicted in Fig.8.

Step1: S generates a random number r_s and timestamp t_s, and then broadcasts the message $\{r_s, t_s\}$.

Step2: Upon receiving the message $\{r_s, t_s\}$, Tag_i checks to see whether $t_s > t_i$, where t_s is a timestamp in Tag_i. If it does not hold, Tag_i ignores the message. Otherwise, it generates a random number $k \in Z_q^*$, selects a random element R_i *in* G, computes $C_1 = r_s\ kP$ and $C_2 = R_i + r_skY$, and computes $x_1 = h(R_i, C_1)$ and $x_2 = h(R_i, C_2)$. If $gcd(x_1, x_2)=l$, Tag_i resets values x_1 and x_2 to x_1 / l and x_2 / l, respectively. It then computes $B_1= x_1H(ID_i)$ and $B_2= x_2H(ID_i)$, and sends message $\{C_1, C_2, B_1, B_2, B_3\}$ to the server.

Step3: Upon receiving the message $\{C_1, C_2, B_1, B_2, B_3\}$, S computes $R_i = C_2 - sC_1$, $x_1= h(R_i, C_1)$, and $x_2 = h(R_i, C_2)$. If $gcd(x_1, x_2)=l$, S resets values x_1 and x_2 to x_1 / l and x_2 / l, respectively. After that, from Euclidean algorithm [24] S can easily find an integer pair (k_1 ,k_2) such that $k_1x_1 + k_2x_2 = 1$. By computing $k_1B_1 + k_2B_2 = (k_1x_1 + k_2x_2)\ H(ID_i)$, S can obtain Tag_i's public identity $H(ID_i)$ and thus relate to Tag_i's identity ID_i.

Server		*Tag$_i$*
$Y=sP$, P		$ID_i=d_iP$, P, Y, t_i
Generates $r_s, t_s \in_R Z_q$	$\xrightarrow{r_s,\ t_s}$	Checks $t_s ?> t_i$ Chooses $R_i \in G$, $k \in_R Z_q$ $C_1 = r_s kP$ $C_2 = R_i + r_s kY$ Computes $x_1 = h(R_i, C_1)$ $x_2 = h(R_i, C_2)$ Lets $gcd(x_1, x_2)=l$ $x_1 \leftarrow x_1 / l$ $x_2 \leftarrow x_2 / l$ Computes $B_1= x_1 H(ID_i)$ $B_2= x_2 H(ID_i)$ $B_3=h(R_i, C_1, r_s)$
	$\xleftarrow{C_1,\ C_2,\ B_1,\ B_2,\ B_3}$	
Computes $R_i = C_2 - sC_1$ $= R_i+r_skY - sr_skP$ $= R_i+r_skY -r_sk(sP)$ $= R_i+r_skY -r_skY$ $B'_3=h(R_i, C_1, r_s)$ Compares B'_3 with B_3, if they doesn’t equal, abort. Computes $x_1 = h(R_i, C_1)$ $x_2 = h(R_i, C_2)$ Lets $gcd(x_1, x_2)=l$ $x_1 \leftarrow x_1 / l$ $x_2 \leftarrow x_2 / l$ uses Euclidean algorithm to find k_1 ,k_2 such that $k_1x_1 + k_2x_2 = 1$ computes $H(ID_i)= k_1B_1 + k_2B_2$ $= k_1x_1H(ID_i) + k_2x_2H(ID_i)$ $=(k_1x_1 + k_2x_2) H(ID_i)$		

Fig. 8 Our RFID authentication protocol based on ECC

5. Security analyses and comparisons

5.1 Security analyses

In the following, we show why our protocol can resist against various attacks.

(a) Replay attack

In the proposed scheme, the timestamp t_s received in the tag side must bigger than Tag_i's timestamp t_i. If an attacker replays the server's message which he intercepted before, the tag will ignore the message, since the replayed $t_s < t_i$.

Now assume that the server sends out r_s, t_s to identify Tag_i. If an attacker impersonates Tag_i and replays the tag's message (C_1, C_2, B_1, B_2, B_3) which he intercepted before. However, without the knowledge of server's secret s, the attacker can not obtain R_i to form valid B_3 to pass server's verification. Hence, the replay attack is doomed to fail.

(b) De-synchronization attack

Our scheme overcomes the de-synchronization attack. Because the authentication data stored in both the tag and the server is $H(ID_i)$ which does not change after every successful communication. Therefore, if an adversary launches a de-synchronization attack on our scheme, he cannot succeed.

(c) Impersonation attack

This attack indicates that an attacker wants to impersonate the server to communicate with Tag_i by sending r_s and t_s. After receiving the responding messages from the tag, the attacker can not compute R_i and henceforth x_1 and x_2 without the knowledge of the server's private key s. Not to mention, he can find the right pair k_1 and k_2 satisfying $k_1x_1 + k_2x_2 = 1$ to compute Tag_i's public identity $H(ID_i)$.

Conversely, if an attacker wants to impersonate the tag to communicate with the server, he must generate the legitimate responding message C_1, C_2, B_1, B_2 and B_3. However without the server's secret s, the attacker can not compute R_i to form valid B_3 for passing server's verification as well. This was demonstrated in part (a), the replay attack. Therefore, the impersonation attack can not work in our scheme.

(d) Man-in-the-middle attack

Assume that an attacker E wants to launch a man-in-middle attack by masquerading as both Tag_i to server S and server S to Tag_i. After receiving the message (r_s, t_s) from S, E modifies it and masquerades as S by sending (r'_s, t'_s) to Tag_i. After Tag_i sending out (C_1, C_2, B_1, B_2, B_3), E also modifies it and masquerades as Tag_i by sending (C'_1, C'_2, B'_1, B'_2, B'_3) to S. For more clarity, we briefly show this scenario in Fig.9. On receiving message 2, (r'_s, t'_s), Tag_i computes (C_1, C_2, B_1, B_2, B_3), and sends it to S. E intercepts it and generates $C'_1 = r_s k' P$, $C'_2 = R'_i + r_s k' Y$, $B'_3 = h(R'_i, C'_1, r_s)$. He can also produce valid $x'_1 = h(R'_i, C'_1)$, $x'_2 = h(R'_i, C'_2)$.

However, without *Tag*$_i$'s identity, *E* cannot produce valid $B'_1 = x'_1 H(ID_i)$, $B'_2 = x'_2 H(ID_i)$ to be successfully authenticated by *S*. Hence, the man-in-the-middle attack fails.

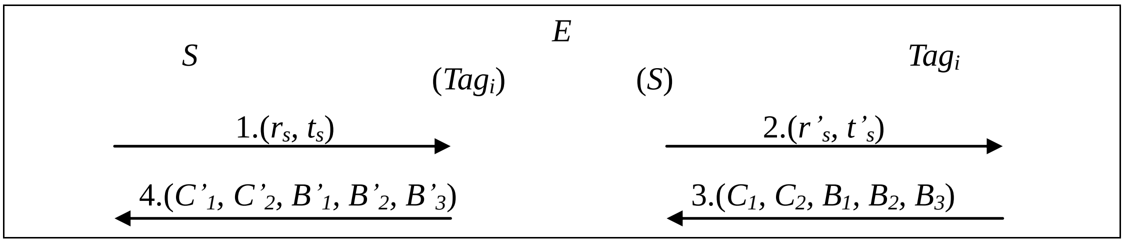


Fig.9 Man-in-the-middle attack

(e) Physical attack

If an attacker obtains *Tag*$_1$'s secrecy, $ID_1=d_1P$, by using physical attack and wants to utilize it to attack *Tag*$_2$. However, there is no relationship between the secrecy of any two tags. Hence, even if the attacker knows *Tag*$_1$'s secrecy, he can not deduce any secrecy of the other tag. Therefore, the effect of the physical attack is confined to *Tag*$_1$.

After analyzing the security features of our scheme, we show the comparison result according to these security features among our protocol and other ECC based RFID schemes in Table 1.

Table 1: Comparisons of resistance against various attacks among some ECC protocols and ours

Schemes	Ours	Schnorr [10]	Batina's [6]	Lee's [17] (EC-RAC IV)
Replay attack	Yes	Yes	Yes	Yes
De-synchronization attack	Yes	Yes	Yes	Yes
Impersonation attack	Yes	Yes	No	Yes
Man-in-the-middle attack	Yes	No	No	Yes
Physical attack	Yes	No	No	Yes

From Table 1, we see that our protocol is as secure as the most recent ECC based RFID method, EC-RAC IV.

5.2 Properties and Comparisons

In the following, we first show the properties which our protocol possesses. In addition, we show the comparison result according to these properties and needed number of passes among our scheme and other related works in Table 2.

(a) Brute search and Scalability

Jamming attack is a kind of deniable of service (DoS). This attack will cause the communication fail between the tag and the reader. If the reader can not deduce the identity from the received message, he must draw each data stored in his database to compare with the authentication value. This condition is termed as brute search. The protocols adopting this method are easy to suffer from jamming attack and hence are non-scalability. Because if a large number of tags suffering Jamming attack communicate with the server simultaneously, the server's DoS will be triggered. Our protocol can avoid the phenomenon since in our protocol the server can use the received messages to compute the tag's public identity (*ID*), so that the server can find the identity directly in its database. Even if the number of tags' suffering Jamming attack is huge, it has no effect on the server for finding the tag. In other words, the proposed protocol is scalable.

(b) Forward secrecy

Forward secrecy means if the tag's secrecy is revealed, the attacker can not use the revealed secrecy to trace any of the tag's previous communications. In our protocol, the values C_1, C_2, B_1, B_2 and B_3 are obviously different from those generated in any previous protocol run, because of the random values r_s, k and R_i which are different in each protocol run. Thus, even an attacker obtains tag's secrecies in a communication between the tag and server, he can not use them to identify any messages sent by the tag before.

(c) Anonymity

Anonymity means an adversary can not distinguish a tag among all the tags from the eavesdropped messages. In the proposed scheme, C_1 and C_2 are two random-liked values; it means the adversary can not utilize C_1 and C_2 to know what the tag's identity is. Although B_1 and B_2 are related to the tag's identity *ID*, the adversary still can not distinguish what messages are generated by the specific tag. Because $B_1= x_1H(ID_i)$, $B_2= x_2\ H(ID_i)$, and $B_3 = h(R_i, C_1, r_s)$ which are also all random points in *G*. $H(ID_i)$ cannot be found in B_1 and B_2 since it is well known that solving Elliptic Curve Discrete Logarithm Problem (ECDLP) is hard.

(d) Untraceability

If the message sent by the tag is partially by the same as the previous sent message, then this tag can be traced by an attacker. In our protocol, the values k and R_i are reset by the tag in each protocol run such that when receiving the server's message, the responding message C_1, C_2, B_1, B_2 and B_3 sent by the tag is distinct from all the messages he responded before. This makes the attacker cannot use the message to trace a specific tag. Below in Table 2, we show the comparison result of our protocol with the other related works using the above mentioned properties along with needed number of passes which is a dominant factor in efficiency consideration. From the table, we can see that our scheme not only possess the same properties as EC-RAC IV but also is the most efficient.

Table 2: Some properties of some ECC protocols and ours

Schemes		Ours	Schnorr	Batina’s	Lee’s (EC-RAC IV)
Brute search		No	Yes	Yes	No
Scalability		Yes	No	No	Yes
Forward secrecy		Yes	No	No	Yes
Privacy	Anonymity	Yes	Yes	Yes	Yes
	Untraceability	Yes	No	No	Yes
no. of passes		2	3	3	3

6. Discussion

In the original design of the proposed scheme, x_1 *and* x_2 are computed by $h(R_i, r_sC_1)$ and $h(R_i, t_sC_2)$, respectively, to prevent C_1 *and* C_2 from being modified. However, to reduce the tag’s computation overhead, we slightly modify it to let x_1 and x_2 be computed by $h(R_i, C_1)$ and $h(R_i, C_2)$ respectively which has no effect on any of the demanding properties. In the future, trying to reduce the computation overhead of the tag is the main target.

In addition, to achieve a higher level security requirement of a RFID authentication protocol, the mutual authentication function is often needed. If we modify the proposed protocol to three passes, we can attain this goal by letting the server generate a ECDSA signature on H(ID_i), then Tag_i can verify the identity of the server. In other words, with little modification, our scheme can accommodate mutual authentication without needing any extra pass. For example, the server can choose a random number $u_1 \in Z_q$ and computes $V=u_1P$, $W= u_1 + s \cdot H(r_s, t_s)$ (s is the private key of the server). Then, the server broadcasts the message (r_s, t_s, V, W). Upon receiving the message, Tag_i computes WP and checks whether it is equal to $V+ H(r_s, t_s) \cdot Y$ or not (Y is the public key of the server). If it does not hold, Tag_i ignores the message. Otherwise, it continues this protocol run. By doing this way, Tag_i can easily authenticate the identity of the server.

7. Conclusion

In this paper, we have reviewed several ECC based RFID authentication protocols and shown that those protocols are flawed. We therefore propose a novel RFID authentication scheme in this aspect (based on ECC), and show that the proposed protocol can resist against various kinds of attacks. Moreover, it also possesses the demanding properties of a RFID system. After comparisons, we conclude that the proposed scheme not only has the same security level as EC-RAC IV but also is the most efficient in some recent ECC based RFID schemes.

Reference

[1] Ahmad-Reza Sadeghi, Ivan Visconti, Christian Wachsmann, “Enhancing RFID Security and Privacy by Physically Unclonable Functions,” *Information Security and Cryptography, Part 5, 2010, Pages 281-305.*

[2] Daewan Han, Daesung Kwon, “Vulnerability of an RFID authentication protocol conforming to EPC Class 1 Generation 2 Standards,” *Computer Standards & Interfaces, Volume 31, Issue 4, June 2009, Pages 648-652.*

[3] Deepak Tagra, Musfiq Rahman, Srinivas Sampalli, “Technique For Preventing DoS Attacks On RFID Systems,” *Software, Telecommunications and Computer Networks, Issue 23-25, September 2010, Pages 6-10.*

[4] Hung-Yu Chien, Che-Hao Chen, “Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards,” *Computer Standards & Interfaces, Volume 29, Issue 2, February 2007, Pages 254-259.*

[5] Kuo-Hui Yeh, N.W. Lo, “Improvement of Two Lightweight RFID Authentication Protocols,” *Information Assurance and Security Letters, Volume 1, 2010, Pages 6-11.*

[6] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, I. Verbauwhede, “Public-Key Cryptography for RFID-Tags,” *Fifth IEEE International Conference on Pervasive Computing and Communications Workshops, 2007, Pages 217-222*

[7] Mark Roberti “A 5-Cent Breakthrough,” *RFID Journal, 2007, http://www.rfidjournal.com/article/articleview/2295/1/128/.*

[8] Mike Burmester1, Breno de Medeiros, “The security of EPC Gen2 compliant RFID protocols,” *ACNS'08 Proceedings of the 6th international conference on Applied cryptography and network security, 2008, Pages 490-506.*

[9] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. E. Tapiador, Arturo Ribagorda, “Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol,” *Information Security Applications, Lecture Notes in Computer Science, Volume 5379, 2009, Pages 56-68.*

[10] Pim Tuyls, Lejla Batina, “RFID-tags for Anti-Counterfeiting,” *Lecture Notes in Computer Science, Volume 3860, 2006, Pages 115-131.*

[11] Robert H. Deng, Yingjiu Li, Moti Yung, Yunlei Zhao, “A New Framework for RFID Privacy,” *Computer Science, Volume 6345, 2010, Pages 1-18.*

[12] Selim Volkan Kaya, Erkay Savas, Albert Levi, Ozgur Ercetin, “Public key cryptography based privacy preserving multi-context RFID infrastructure,” *Ad*

Hoc Networks, Volume 7, Issue 1, 2009, Pages 136-15.

[13] Selwyn Piramuthu, “RFID mutual authentication protocols,” *Decision Support Systems, Volume 50, Issue 2, 2011, Pages 387-393.*

[14] Serge Vaudenay, “On Privacy Models for RFID,” *Advances in Cryptology, Lecture Notes in Computer Science, Volume 4833, 2007, Pages 68-87.*

[15] Tzu-Chang Yeh, Yan-Jun Wang, Tsai-Chi Kuo, Sheng-Shih Wang, “Securing RFID systems conforming to EPC Class 1 Generation 2 standard,” *Expert Systems with Applications, Volume 37, Issue 12, 2010, Pages 7678-7683.*

[16] Yalin Chen, Jue-Sam Chou, Hung-Min Sun, “A novel mutual authentication scheme based on quadratic residues for RFID systems,” *Computer Networks, Volume 52, 2008, Pages 2373-2380.*

[17] Yong Ki Lee, Lejla Batina, Dave Singelee, Bart Preneel, Ingrid Verbauwhede, “Anti-counterfeiting Untraceability and Other Security Challenges for RFID Systems- Public-Key-Based Protocols and Hardware,” *Information Security and Cryptography, Part 5, 2010, Pages 237-257.*

[18] Yong Ki Lee, Lejla Batina, Ingrid Verbauwhede, “EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID authentication protocol,” *IEEE International Conference on RFID, 2008, Pages 97-104.*

[19] Yong Ki Lee, Lejla Batina, Ingrid Verbauwhede, “Untraceable RFID authentication protocols: Revision of EC-RAC,” *IEEE International Conference on RFID, 2009, Pages 178-185.*

[20] Yongquan Cai, Xiuying Li, “Identity-based Conference Key Distribution Scheme Using Sealed Lock,” *IEEE/ACIS International Conference on Computer and Information Science, 2008, Pages 282-286.*

[21] Boyeon Song, Chris J. Mitchell, “Scalable RFID security protocols supporting tag ownership transfer,” *Computer Communications, Volume 34, Issue 4, 1 April 2011, Pages 556-566.*

[22] Mike Burmester, Breno de Medeiros, “The security of EPC Gen2 compliant RFID protocols,” *ACNS'08 Proceedings of the 6th international conference on Applied cryptography and network security, 2008, Page 490-506.*

[23] Wenbo Mao, *Modern Cryptography - Theory And Practice*, 2003, Prentice Hall, Pages 196-203.

[24] Ralph P. Grimaldi, *Discrete and Combinatorial Mathematics - An Applied Introduction, Fourth Edition*, 1999, Addison-Wesley, Pages 202-205.

[25] Douglas R. Stinson, *Cryptography – Theory and Practice*, 1995, CRC Press Inc.