# An efficient RFID mutual authentication scheme based on ECC

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

[1,3,4] Department of Information Management, Nanhua University, Taiwan

*: corresponding author: jschou@mail.nhu.edu.tw

wfdawu@gmail.com, chejtmmc@gmail.com

Tel: 886+ (0)5+272-1001 ext.56536

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

d949702@oz.nthu.edu.tw

## Abstract

Recently, Radio Frequency Identification (RFID) technique has been widely deployed in many applications, such as medical drugs management in hospitals and missing children searching in amusement parks. The applications basically can be classified into two types: non-public key cryptosystem (PKC)-based and PKC-based. However, many of them have been found to be flawed in the aspect of privacy problem. Therefore, many researchers tried to resolve this problem. They mainly investigated on how low-cost RFID tags can be used in large-scale systems. However, after analyses, we found those studies have some problems, such as suffering physical attack or de-synch attack. Hence, in this paper, we try to design an efficient RFID scheme based on Elliptic Curve Cryptography (ECC) to avoid these problems. After analyses, we conclude that our scheme not only can resist various kinds of attacks but also outperforms the other ECC based RFID schemes in security requirements, with needing only little extra elliptic curve point multiplications.

***Keywords:*** RFID, location privacy, forward secrecy, mutual authentication

## 1. Introduction

Radio Frequency Identification (RFID) systems can identify hundreds of objects in a contactless manner at one time. This benefit brings themselves the potential replacement of barcodes which are possibly scanned billion times worldwide in a day. They raise many new applications, such as EasyCard which is adopted for the payment by Taipei Rapid Transit System in Taiwan [3], anti-counterfeit drugs which is supported by Food and Drug Administration (FDA) in U.S.A, access control cards in safeguard, and supply chain management in commerce. Most recently, some scholars apply RFID technique in store management [2]. In this kind of application, whenever a reader eliminate queries in the store, each RFID tags attached on distinct articles will answer their information. But when out the store, they each will answer a meaningless number to prevent article information leakage. However, the reader can recognize the meaningless number and recover the tag information if the

article is returned by customer. Some scholars [10] use RFID system to search for missing children in amusement park or some public place. Study [11] applies RFID tags in mobile environment. This solution is suitable for ubiquitous society in the future. Study [22] presents a multi-context RFID infrastructure which allows a RFID tag to play different roles in different contexts. For example, in a specific context a tag can be a health insurance card identified by the backend server through a hospital reader and provide personal medical information, but for another context, it can be a financial card used to authenticate an individual's identity to a bank for acquiring financial services.

A typical RFID system has three parties: tag, reader, and back-end server, as shown in Fig. 1. When objects embedded with tags are to be identified, a reader emits an interrogation signal over the air. The tags in the range of the signal will answer the reader with authentication-related messages respectively. After receiving the responses, the reader passes it to the back-end server. The server, maintaining tags' information, will confirm the legality of each tag. In addition, the communication channel between the tags and reader is a radio interface and is supposed to be insecure. Whereas the channel between the reader and back-end server is a fixed infrastructure and assumed to be secure in general. The insecure channel is vulnerable to various threats such as eavesdropping, business espionage, and tag masquerading.

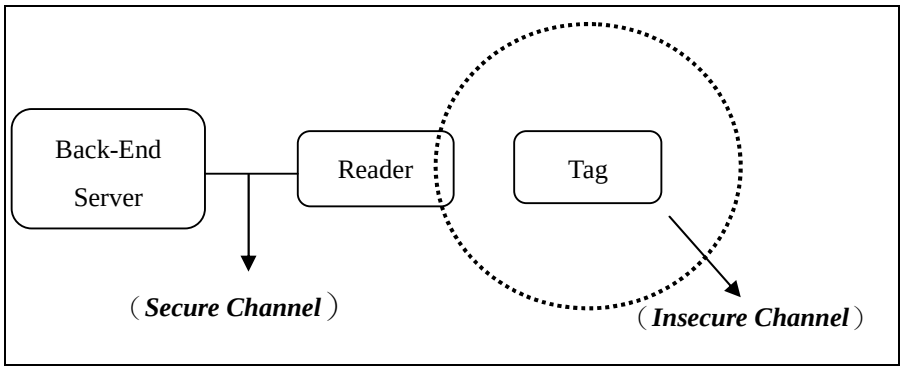


**Fig. 1. A typical RFID system**

To counter these threats, many secure RFID authentication systems have been proposed. [2, 5, 8, 12, 13, 21, 28, 30]. In the following, we define seven important security issues about RFID systems according these works.

(1) Location privacy [5]:
An adversary can trace the location of a tag if the tag always outputs a fixed value.

(2) Forward secrecy [12, 21, 30]:
Even if an adversary can know all the resident data of a tag, the adversary should not be able to trace the tag through past conversions.

(3) Replay attack [5]:
An adversary resends data, which is ever transmitted from a tag (server), to a server (tag), whereas the server (tag) unconsciously accepts the data and believes the attacker is a valid tag (server).

(4) Impersonation attack [30]:
An adversary can successfully impersonate a server (tag) to authenticate himself to the tag (server) when he doesn't know the server's (tag's) secret data.

(5) Man-in-the middle attack [28]:
An adversary stands between a server and a tag and modifies both server-to-tag and tag-to-server messages to make them believe that they are talking to the intended party, respectively.

(6) Physical attack [5,8,13]:
An attacker can use physical means to undermine a tag to obtain the stored secrecy and further analyze them to deduce other tag's privacy. Low-cost tag usually does not have a tamper-resistant device and thus cannot prevent this kind of attack. Hence, designing a secure RFID authentication protocol with physical-attack resistance becomes a desire task.

(7) Mutual authentication [2, 5]:
Mutual authentication means a tag should authenticate himself to a server, and vice versa.

We here classify the previous RFID authentication systems into non-public key cryptosystem (non-PKC)-based and PKC-based systems. Some non-PKC-based RFID systems use of simple bit-wise operations, like XOR, AND, OR, and rotation [7, 14-18], some support cyclic redundancy code (CRC) and pseudo random number generator, like the EPCglobal Class-1 Gen-2 RFID standard [9], some [2, 5, 11] adopt one-way hash functions, random number generating functions, or symmetric-key encryptions. However, the robustness of non-PKC-based RFID solutions using only simple bit operations, CRC, and PRNG are easily challenged. Moreover, they usually suffer from the scalability problem. The problem indicates that the back-end server always requires a linear search to identity a tag. In other words, the server must take $O(n)$ times to authenticate a tag, where $n$ is the number of tags in the system, and therefore demands more searching cost when $n$ is getting larger. In contrast, the PKC-based RFID approaches are not only stronger in the guard of both privacy and security but also easier in addressing the scalability problem. Nevertheless, a tag embedded with a PKC component will need more hardware cost. Because a low-cost tag, pricing \$0.01 to \$0.05, contains only 500 to 5K gates, but traditional PKC primitive costs at least 20K to 30K gates [13]. This causes many researchers deem the

PKC-based RFID systems to be infeasible at present. Fortunately, many studies try to cheap PKC primitive implementations [22, 28, 31]; for example, [28, 31] implement an ECC scalar multiplication component using about 12.5K gates, [24] implements Rabin's encryption with cost about 17K gates, and [22]'s NTRU public encryption costs only about 3K gates. Besides, with the same security level, a RSA public-key encryption would require a key length of 1024 bits while ECC only needs 160 bits. Therefore, researchers believe that ECC could be a viable solution for low-cost RFID tags [26, 27, 28, 29].

Hence, in this paper, we will adopt ECC primitives [4] to design an efficient RFID mutual authentication protocol. Compared to previous related works, the proposed protocol is more efficient in both communicational and computational cost while achieving the same security level. The remaining of this paper is organized as follows. Section 2 reviews some recent RFID works and introduces the concept of ECC. We then present an efficient RFID mutual authentication scheme based on ECC in Section 3 and analyze its security in Section 4. The comparisons with other works and its discussion are presented in Section 5. Finally, a conclusion is given in Section 6.

## 2. Recent RFID studies

In this section, we briefly review recent RFID studies. We classify them into two types: non-PKC-based and PKC-based RFID systems, and discuss them in Section 2.1 and 2.2 respectively.

### 2.1 Non-PKC-based RFID systems

Ryu et al. [1] in 2009 proposed a hybrid solution for the privacy of RFID tags. In their design, the server stores the set of one-time values $\Delta=\{\alpha_1, ..., \alpha_m\}$ in tag's memory, where $\alpha_i = E_{pk}\{tagID||r\}$, $r$ is a random number, and $E_{pk}(.)$ is a RSA encryption using public key $pk$. For each authentication, the tag sends a fresh $\alpha_i$ to the server. The server can authenticate the tag in constant time $O(1)$ by decrypting $\alpha_i$ using its private key to obtain $tagID$. However, we consider that the size of $\alpha_i$ might be too large and thus impractical for memory-limited low-cost tags. Because if the tag memory size is 2K bits and the security level of the RSA used in Ryu et al.'s system is 512-bit, then the number of $\alpha_i$ a tag can store is only 4 (2048 / 512 = 4). This means the tag can be authenticated by the server at most 4 times.

Burnmester et al. [16] in 2007 proposed an optimistic forward-secure RFID authentication protocol, called O-FRAP for short. They devise their protocol in an attempt to reduce security overhead. In the protocol, a tag utilizes a pseudo-random value $r_{tag}$ (stored in the tag's memory), the server's challenge $r_{sys}$, and the tag's current key $k^a_{tag}$ to generate four values: one for pseudonym, one for authentication

token, and the other two for updating the old values, $r_{tag}$ and $k^a_{tag}$. However, we found if an adversary maliciously queries a tag twice, the tag will update its key twice without the server's updating its key simultaneously. This incurs the tag's current key $k^a_{tag}$ cannot be recognized by the server and results in a de-synchronization error. Burnmester [17] in 2009 modified O-FRAP to O-RAP. Although O-RAP needs not to update the authentication key $k_{tag}$; however, Duc et al. [7] in 2011 pointed that both O-FRAP and O-RAP are vulnerable to DOS (denial of service) attack. They also proposed a new solution. Unfortunately in their solution, we found that with only one physical attack on a tag, the common secret key $K_S$ shared among the server and all tags will be known. The attacker then can masquerade as the server, authenticate himself to a tag by employing $K_S$, and make the tag update its pseudonym with a wrong one. Of course, this wrong pseudonym cannot be recognized by the server anymore. .

Kang et al. [11] in 2008 proposed a secure RFID mutual authentication method for pervasive computing environments. Although they claimed their protocol is secure, we found it has a weakness. If an adversary eavesdrops on the reader-to-tag query, $G\_key_i \oplus r || G\_key_i \oplus TS || H(r||TS)$ in the first session and replays the query to the tag in the second. The tag will answer message, $\Delta r \oplus metaID \,||\, H(\Delta r \,||\, metaID)$. This message is the same as the first session one. Because when the replay attack occurs, $\Delta r$ will be the same as in the previous session (since $\Delta r$ is computed by *r-TS*, and *r* and *TS* are in the replayed query message) and *metaID* is a fixed value. Therefore, the adversary can easily track the location of the tag.

Liu et al. [2] in 2009 proposed a private authentication protocol for passive RFID tags in a retail store environment. However, we found their design makes the commodities traceable and leaks individual's location privacy. More precisely, an adversary can do the following steps to track a person who carries a product embedded with a Liu et al.'s tag. The adversary first launches a physical attack on a tag to obtain the secret key *k* shared between the reader and tag. Then, the other tags in the same group (the same category of commodities) will be insecure. This is because tags in the same group employ the same secret key *k*. Consequently, the adversary can use this key to trace any tag in the group.

Yeh et al. [18] in 2010 proposed a RFID authentication scheme conforming to EPC Class1 Generation 2. However, we found there exists a privacy problem in their scheme. An attacker can intercept and drop the server-to-tag authentication information and cause the tag not to update the value $C_i$ stored in its memory. This will result in the traceability of the tag by the attacker querying the tag twice, since the tag will answer with the same response $C_i$ as in the previous one.

Cho et al. [5] in 2010 proposed a hash-based RFID mutual authentication protocol, trying to prevent a brute-force attack. However, we found there is a scalability problem existed in their scheme. Because in the authentication phase, the server must first extract tag's ID ($ID_k$) and secrecy $s_j$ stored in its database for generating $\beta$ to compute $R_t \oplus \beta$, then utilize $s_j$ and $R_t$ to calculate *RID*, and generate $\alpha'$ by computing $h(ID_k \oplus R_t \oplus R_r \oplus RID)$ to check if equation $\alpha' = \alpha$ holds, where $\alpha$ is sent from the tag. It repeats this step for each tag stored in its database until a tag can satisfy the equation.

Song and Mitchell in 2011 [25] proposed a scalable RFID security protocol supporting tag ownership transfer. They claimed that their protocol requires only constant time to identify a tag. However, after a tag is maliciously queried more than *m* times and then queried by the true server, the server requires performing a linear search to identify the tag. Moreover, in the tag secrecy update phase, after receiving *r* and $M_s (= g_k(r||r_T) \oplus (s||k'||m'))$ from the server where, $g_k(.)$ is a hash function and $r_T$ is tag's one-time challenge, the tag computes $g_k(r||r_T) \oplus M_s$ to obtain $S||K'||M'$. It then uses *S* to compute *h* (*S*) and examines whether the computed *h*(*S*) is equal to the old *k*. If it is, the tag updates *k* and *c* to *K'* and *M'*, respectively. However, if an adversary intercepts $M_s$ and modifies the second *l* bits, obtaining $M_s'$, then when the tag computes $g_k(r||r_T) \oplus M_s'$, the second *l* bits of the computation result *K'* will be different from $k'$ owned by the sever. This makes the tag and the server desynchronized.

### 2.2 PKC-based RFID systems

For this type of RFID systems, we further classify them into two kinds: (a) non-ECC based schemes, and (b) ECC based schemes. We describe them as follows.

**(a) Non-ECC based schemes**

Chen et al. in 2008 [19] proposed a private mutual authentication scheme based on quadratic residues (also referred as Rabin cryptosystem). In the scheme, the tag uses Rabin's encryption to produce dynamic pseudonym and the server performs only one decryption to obtain the tag's identify without requiring linear search. However, in 2008, Cao et al. [20] found Chen et al.'s scheme suffers from impersonation attack. In addition, in 2010, Yeh et al. [21] further pointed that Chen et al.'s scheme suffers the location privacy problem and replay attack. They further proposed an improvement. However, the improvement requires three Rabin's encryptions and four hash operations on the tag side for each authentication. This computational overhead seems impractical for a power-limited low-cost tag.

**(b) ECC based schemes**

For introducing ECC based RFID schemes, in this sections, we first briefly introduce the concepts of elliptic curve cryptography (ECC) and elliptic curve discreet logarithm problem (ECDLP), then discuss some ECC based schemes and their weaknesses.

**(1) Elliptic curve cryptography and ECDLP**

In 1985, Koblitz and Miller independently proposed the concept of ECC [32] which assumes that there exists an elliptic curve $E$: $y^2=x^3+ax+b$ over a finite field, where $4a^3+27b^2 \neq 0$ and the finite field can be $Z_P$, $F_2$, or others. All $(x, y)$ points satisfying this elliptic curve equation along with one infinite point $O$ and an addition operation form a group $G$ which has the following properties.

- A point $P$ which can generate all points in group $G$ is called a generator or base point. In addition, if $nP = O$, then $n$ is the order of $G$.
- For a point $P = (x, y)$ in $G$, its inverse is defined as $-P = (x, -y)$.
- Addition rules:
  - For all $P \in G$, $P+O=O+P=P$, $P+(-P)=O$.
  - Let $P, Q \in G$, $P=(x_1, y_1)$, $Q=(x_2, y_2)$, and $P \neq -Q$. Then $P+Q=(x_3, y_3)$, where $x_3=\lambda^2-x_1-x_2$, $y_3=\lambda(x_1-x_3)-y_1$, and

$$\lambda=\begin{cases}\dfrac{y_2-y_1}{x_2-x_1} & \text{if } P \neq Q \\ \dfrac{3{x_1}^2+a}{2y_1} & \text{if } P=Q\end{cases}$$

  - Let $s, t \in Z_n$, for all $P \in G$, $(s+t)P = sP + tP$.
- Multiplication rules:
  - Let $k \in Z_n$, for all $P \in G$, $kP=P+P+\ldots P$, adding $P$ $k$ times.
  - Let $s, t \in Z_n$, for all $P \in G$, $s(tP) = stP$.

After introducing the concept of ECC, an ECDLP [32] can be defined as follows.

**Definition 1.** Let elliptic curve group $G$ be defined as above, which has generator $P$ and order $n$. Given $X \in G$ and $Y \in G$, to find the integer $c \in Z_n$ such that $Y = cX$, is called Elliptic Curve Discrete Logarithm Problem (ECDLP) and is considered as hard.

**(2) Some ECC based RFID schemes and their weaknesses**

Tuyls et al. [26] in 2006 proposed an ECC-based RFID identification scheme using Schnorr identification protocol, as shown in Fig. 2. They claimed their scheme can resist against tag counterfeiting, but Lee et al. [29] in 2008 pointed their protocol suffers a privacy problem. Because if an adversary eavesdrops and obtains $\{X_1, e, y_1\}$, he utilizes $e^{-1}$ to obtain $Z$ ($=-aP$) by computing $(X_1- y_1P)\ e^{-1}$, where $Z$ is a certain tag's public key and only known to the server. Hence, the adversary can use $Z$ to track the

tag. Moreover, we found another privacy problem in their scheme. If an adversary first eavesdrops on the communication between a reader and a specific tag and obtains three values, $X_1$ (= $r_1P$), $e$, and $y_1$ (= $ae+r_1$), where $a$ is the tag's private key. Then, after receiving $X_2$ (=$r_2P$), he replays the *challenge e'* (=$e$) to the unknown tag and obtains $y_2 = ae+r_2$. As a result, he can identify the unknown tag as the specific tag if $(y_2 - y_1)P$ equals to $X_2 - X_1$. Besides, we think that Tuyls et al.'s protocol lacks forward secrecy. This is because when an adversary, performing above-mentioned steps, obtains the public key $Z$(=$-aP$) of a tag, he can use $Z$ to track the past conversations of the tag. In addition, a scalability problem also exists in Tuyls et al.'s scheme. This is Because the server should fetch each tag's public key $Z$ from its database to compute $yP + eZ$ for comparing with the received $X_1$. This means the server requires linear search to identity each tag and thus increases considerable computational cost. Hence, their protocol lacks scalability.

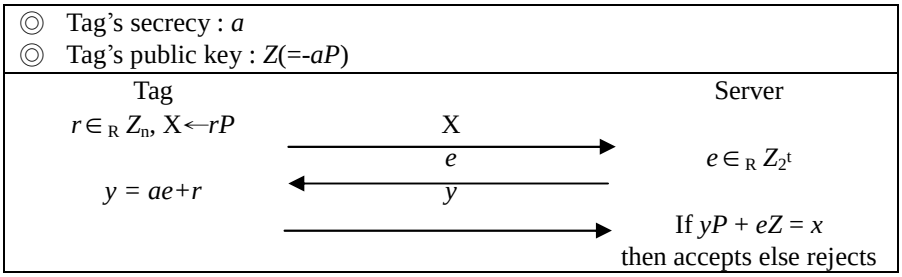


**Fig. 2. Tuyls et al.'s scheme**

Batina et al. in 2007 [27] proposed an ECC-based RFID identification protocol using Okamoto's identification, as shown in Fig. 3. Although they claimed their protocol can avoid active attacks, Lee et al. in 2008 [29] pointed their protocol exists a tracking problem. Because if an adversary obtains {$X$, $e$, $y_1$, $y_2$} and utilizes $e^{-1}$ to obtain $Z$ (= $-y_1P_1 - y_2P_2$) by computing $(X- y_1P_1- y_2P_2)\ e^{-1}$, where $Z$ is tag's public key, he can then use $Z$ to track the tag. Moreover, we also found a forward-secrecy violation and a traceability problem existing in their scheme. This is Because $X$, $e$, $y_1$ and $y_2$ are publicly transferred and both $P_1$ and $P_2$ are system parameters. If an adversary first eavesdrops on the communication between the tag and reader, he can obtain the value $e \cdot Z$ by computing $X - y_1P_1 - y_2P_2$. Then, in another conversation, when the tag transfers $X'$ to the reader, the adversary can impersonate the reader to communicate with the tag by sending a challenge $e'$=$e$+1 to the tag. If the tag answers $y'_1$ and $y'_2$, then the adversary can obtain the value $e' \cdot Z$ by computing $X' - y'_1P_1 - y'_2P_2$ and therefore can extract $Z = e'Z - eZ$. The adversary can then use $Z$ to distinguish the tag from the past conversations easily. Hence, their protocol does not achieve forward secrecy. Moreover, the adversary can also utilize $Z$ to trace the tag. This makes their scheme fails in untraceability.

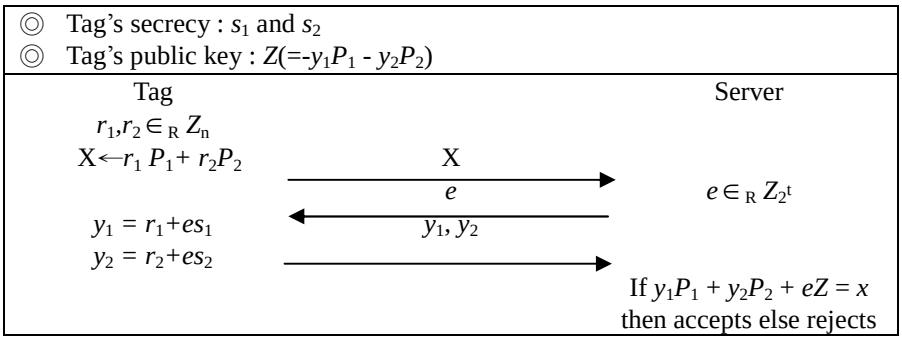


**Fig. 3. Batina et al.'s scheme**

Lee et al. in 2010 [28] proposed an ECC-based RFID authentication protocol, as shown in Fig. 4, to address the tracking problems existing in [26, 27]. In the figure, $\dot{r}_{s1} = x(r_{s1}P)$, indicating the $x$-coordinate of $r_{s1}P$, plays the key role to resist the possibility of linear operations on previous eavesdropped data to avoid privacy leakage like in [26, 27]. However, their protocol only considers tag-to-reader authentication, excluding reader-to-tag authentication. This makes tags easy to suffer malicious queries, because they are not capable of confirming whom they are talking to.

From the above mentioned, we know that there still lacks a secure RFID system to resist against possible attacks. Hence, in the following, we proposed a novel protocol possessing mutual authentication to resolve this problem.

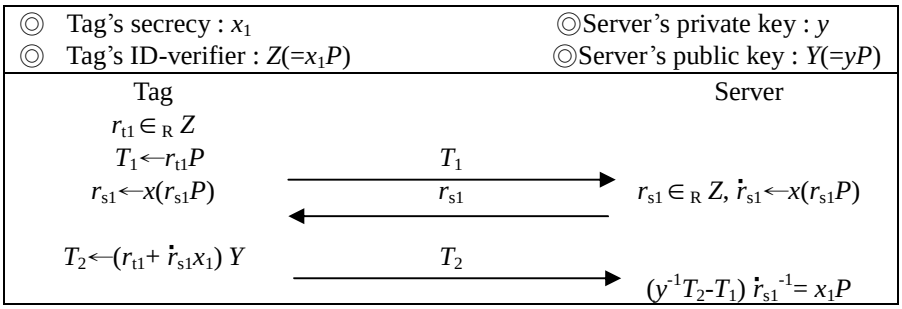


**Fig. 4. Lee et al.'s scheme**

## 3. The proposed scheme

In this section, we present an ECC based RFID scheme in Section 3.1 and exemplify it in Section 3.2. There are two roles: server/reader and tag in our RFID system. As in traditional RFID systems, we use server to stand for server/reader and assume that the communication between the server and tag is insecure. Our protocol consists of two phases: (1) setup phase, and (2) authentication phase. Before describing these two phases, we first introduce some notations used.

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

$P$: a primitive element of $G$,

$x_i$: $tag_i$'s private key,
$ID_i = X_i$ ($=x_iP$): $tag_i$'s identify,
$y$: server's private key,
$Y$ ($=yP$): server's public key,
$Cs$, C: two counters,
$r$, $k$: two random numbers in $Z_q$,
$h$: an one-way hash function

### 3.1 Our scheme

Our scheme consists of two phases. We demonstrate them as follows.

**(a) Setup phase**

In this phase, the server chooses a random number $y \in Z_q$ as his private key and sets $Y$ ($= yP$) as its public key. It also chooses $x_i \in Z_q$ as the private key for $tag_i$ and sets $X_i$ ($= x_iP$) as $tag_i$'s identity $ID_i$. In addition, the server stores each $tag_i$'s identity and information in its database, where the information includes the name of the tag and production number, etc., as shown in Table 1.

Finally, the server initializes a counter value $Cs$ to one and stores each $tag_i$'s data $[x_iP, Y, P, C]$ in the memory, where $C$ is $tag_i$'s counter initialized to zero.

**Table 1. Server's database containing tags' identity and information**

| identity | information |
|---|---|
| $ID_i=X_i$ ($=x_iP$) | name, production number |
| ⋮ | ⋮ |

**(b) Authentication phase**

In this phase, the server chooses a random number $r \in Z_q$ and computes $s_1P = (r + y + C_s)P$. It then sends $C_s$, $s_1P$ to the tag and then increments $C_s$ by $r$. After receiving the message, $tag_i$ checks whether $C_s > C$ holds. If so, it replaces $C$ by $C_s$ and picks a random number $k \in Z_q$. Then, it computes $C_1 = kP$, $C_2 = x_iP + s_1P + kY$, and $C_3 = h(x_iP, C_2)$, and sends these three values to the server. After receiving the values, the server utilizes its private key $y$ to compute $C_4 = C_2 - yC_1 - s_1P$ ($= x_iP + s_1P - s_1P$) $= x_iP$. Then, it searches $tag_i$'s identity $ID_i$($=X_i=x_iP$) in the database. If found, the server computes $h(x_iP , C_2)$ and compares it with the received $C_3$. If they are equal, the server confirms the tag is legitimate. Otherwise, it is illegitimate. If the tag is legal, the server sends back $C_5 = h(C_4+s_1P)$ to be authenticated by the tag. The tag computes $h(C_2-kY)$ to examine whether it is equal to the received $C_5$. If it is equal, the tag conforms that the server is authentic. We depict the process in Fig. 5.

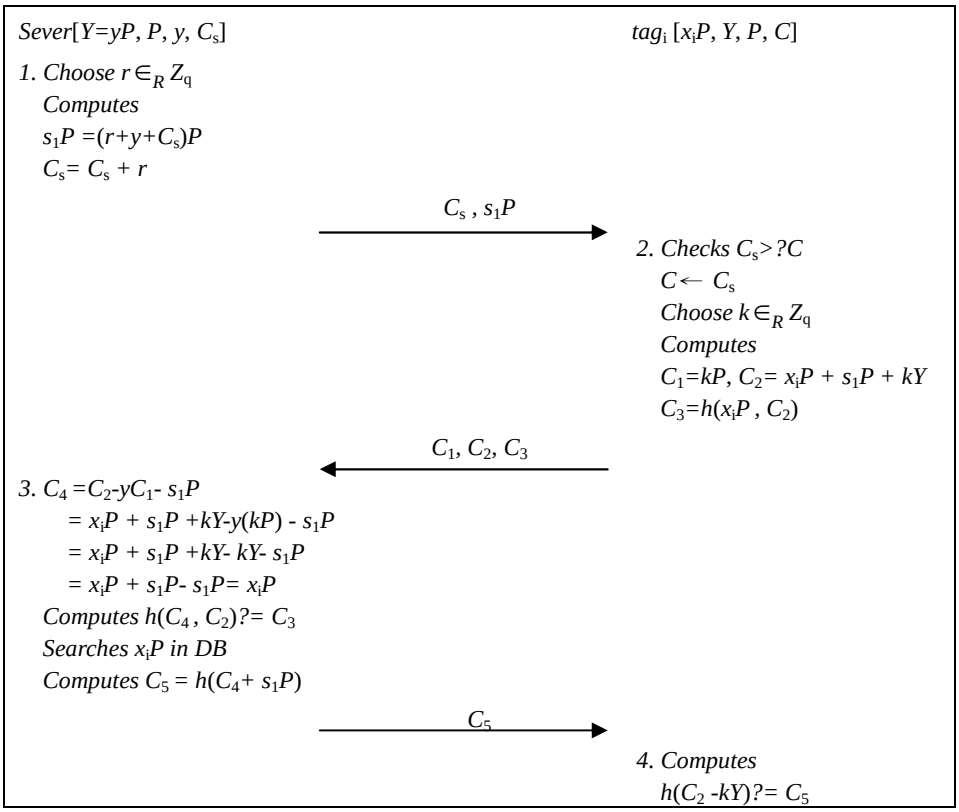


**Fig. 5. Proposed RFID mutual authentication scheme**

### 3.2 Example

In this section, we demonstrate the correctness of our scheme by running a program. Below, we first show the platform for running the program in part (a), then demonstrate the result in part (b).

**(a) Platform**

Our machine is equipped with Intel(R) Pentium(R) CPU 2.00 GHz with 2 GB of RAM on windows XP. In the experiment, the Pairing-Based Cryptography (PBC) [23] library is used for implementing the proposed scheme. It is a C library that can perform EC point multiplications and pairing-based calculation. In addition, we also use an one-way hash function HashMyFiles [6] in our scheme.

**(b) Exemplify**

We use PBC to design a program based on ECC for running our protocol. Due to that showing the complete result are too long to be fit in this paper, we only use the most significant 5 decimal digits to show x and y coordinates of the points and the most significant 5 hexadecimal digits to show the hash result in the example, as shown in Fig. 6 (The complete example can be seen in Appendix A). In the figure, we let the primitive element of G be $P$ = (18643, 38484), $tag_i$'s identify be $ID_i = X_i(=x_iP)$ = (72423, 65961), server's private key $y$ = 41882, and server's public key $Y$ ($=yP$) = (54290, 16087).

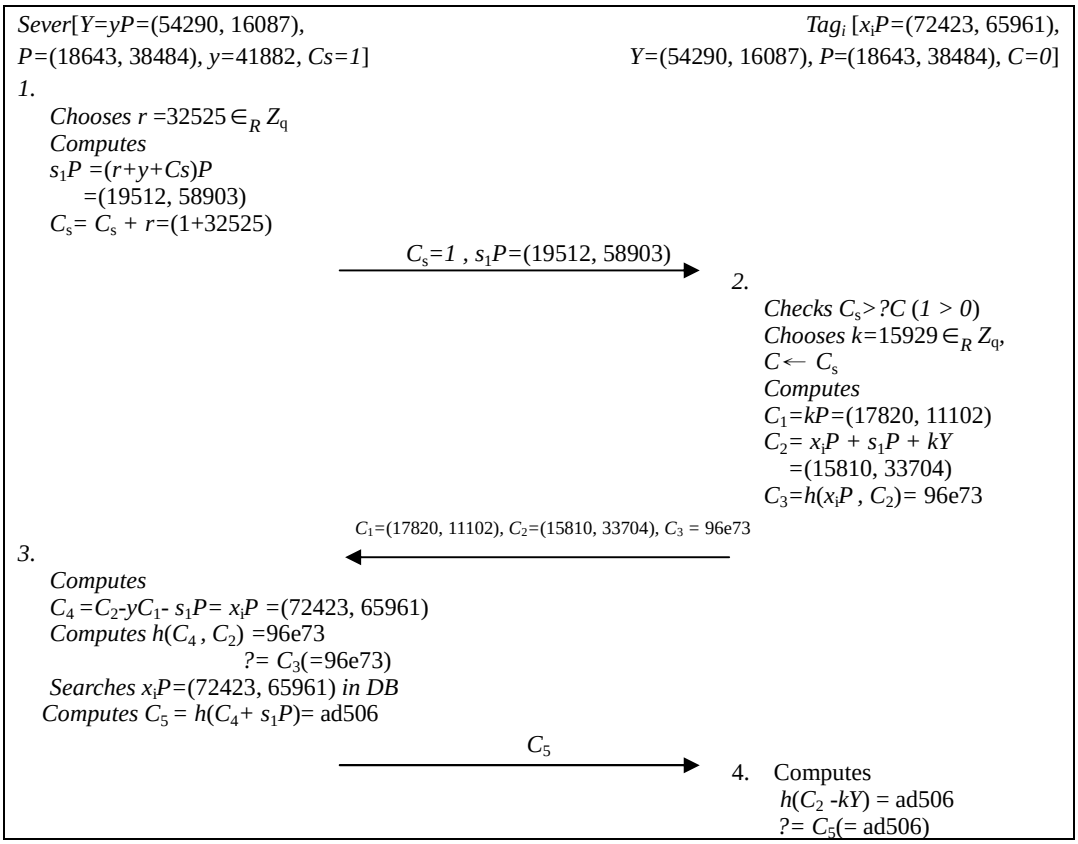


**Fig. 6. An example for the proposed RFID mutual authentication scheme**

## 4. Security analysis

In the following, we will probe some attacks, which often occur in a RFID system, on our scheme. After analyzing, we found no attack on our protocol can work. The analyses are shown in the following.

(a) Location privacy:

An attacker E cannot know the location of the tag in our scheme, since the tag randomly chooses a value $k$ in each session, so that each response $\{C_1, C_2, C_3\}$ to the server is different. If E eavesdrops on the transmitted information and gets two responding messages, $\{C_1, C_2, C_3\}$ and $\{C_1', C_2', C_3'\}$ where $C_1(=kP)$, $C_2(=x_iP+s_1P+kY)$, $C_3= h(x_iP, C_2)$, $C_1'(=k'P)$, $C_2'(=x_iP+s_1'P+k'Y)$, and $C_3' = h(x_iP, C_2')$. He tries to compute $C_2- s_1P-C_2'- s_1'P$ $(=kY- k'Y)$ to relate to a specific tag. But this affords no clue for him to attain this goal, since $k$ and $k'$ are twp random numbers.

(b) Replay attack:

In the latest session, the tag replaces $C$ with $C_s$. Hence, when E intercepted the message $C_s$, $s_1P=(r+y+C_s)P$ in a previous session and launches a replay attack by resending the message to the tag, the authentication cannot succeed. Because the tag will check if $C_s > C$ is correct. In addition, if the attacker intercepted the

second message flow $\{C_1=k'P, C_2=x_iP+ s_1'P - k'P, C_3= h(x_iP, C_2)\}$ in a previous session. He then replays $C_1$, $C_2$, and $C_3$ to the server. The server's authentication cannot succeed as well. Because in the server side, the computation result of $C_4 = C_2 - yC_1 - s_1P = x_iP+ s_1'P- s_1P$ yields no information about $x_iP$ in its database.

(c) Man-in-the middle attack:

Suppose E launches a man-in-the middle attack (MIMA) between the server and the tag, we describe the details as follows and also illustrate it in Fig.7.

Step 1: The sever chooses a random number $r$, computes $s_1P=(r+y+C_s)P$, and sends $\{C_s , s_1P\}$ to the tag.

Step 2: E intercepts the message $\{C_s , s_1P\}$. It chooses another random number $r'$ and utilizes $Y(=yP)$, and $C_s$ to compute $s_1'P=(r' +C_s)P+Y$ and impersonates the server by sending $\{C_s , s_1'P\}$ to the tag.

Step 3: After receiving the message, the tag computes $C_1(=kP)$, $C_2(=x_iP+ s_1'P + kY)$, and $C_3= h(x_iP, C_2)$, and then sends $\{C_1, C_2, C_3\}$ to the server.

Step 4: E intercepts the message $\{C_1, C_2, C_3\}$ sent by the tag to the server. However, E does not know $tag_i$'s secret $x_iP$. E chooses a new $x_i'$ and random number $k'$ to compute $C_1'(=k'P)$, $C_2'(=x_i'P+ s_1'P + k'Y)$, and $C_3'= h(x_i'P, C_2')$, and masquerades as the tag by sending $\{C_1', C_2', C_3'\}$ to the server.

Step 5: After receiving the message, the server computes $C_4= C_2'-yC_1' - s_1P$ $(=x_i'P+ s_1'P- s_1P \neq x_iP)$. Obviously, the server cannot use $C_4$ to find out $Tag_i$'s identity $ID_i$ $(=x_iP)$. It means the server's authentication fails for $Tag_i$. As a result, the server will not compute $C_5= h(C_4+s_1P) = h(x_iP +s_1P)$ to send it to the tag for authentication.

Step 6: Even receiving $C_5$ from the server, if E wants to impersonate server to the tag, he must generate $C_5'= h(x_iP+s_1'P)$ for the tag's authentication. However, not having tag's secret $x_iP$, E cannot compute and send $C_5'$ to the tag. It means E cannot pass tag's authentication. Therefore, the MIMA fails.

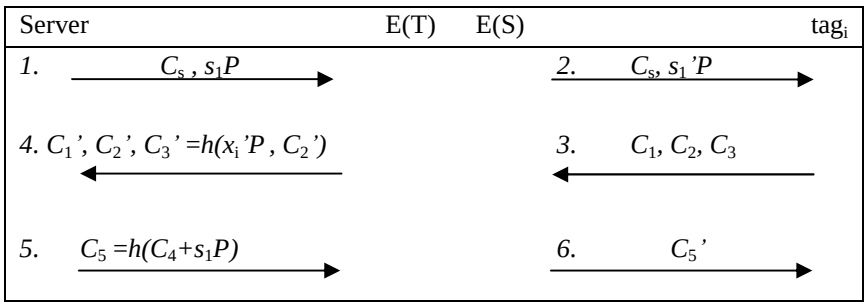


**Fig. 7. Man-in-the middle attack**

(d) Impersonation attack:

If E wants to impersonate the tag to the server, he will fail. This is because E must use the $tag_i$'s secret $x_iP$ to compute valid $C_2$ and $C_3$ for the server's authentication. Conversely, if the attacker wants to impersonation the server to the tag, he must use $x_iP$ to compute $C_5$. However, without the server's computed value $C_4(=x_iP)$, he cannot have value $C_5$ $(= h(C_4+s_1P))$ to pass $tag_i$'s authentication.

(e) Physical attack:

Even if E uses physical means to obtain the secret $x_1P$ of $tag_1$, he still cannot know the secret of the other tag, $tag_2$. For example, if E eavesdrops on the communication between $tag_2$ and the server. He can obtain $\{C_s', s_1'P\}$, and $tag_2$'s transmitted values $\{C_1'$ $(= k'P)$, $C_2'$ $(= x_2P+s_1'P+k'Y)$, $C_3'$ $(= h(x_2P, C_2'))\}$. However, by using these values, E cannot obtain $x_2P$. For example, if E tries to compute $C_2'- s_1'P - x_1P=x_2P +s_1'P+ k'Y- s_1'P -x_1P$, he still cannot obtain $x_2P$, the secret of $tag_2$.

(f) Forward secrecy:

If E compromises $tag_i$'s resident data $x_iP$, he cannot identify the $tag_i$ by tracing though any previous communications. For example, if E eavesdrops on two successive rounds and obtains $C_2(=x_iP+s_1P+kY)$, $s_1P(=(r+y+C_s)P)$ and $C_2'(=x_iP+s_1'P+k'Y)$, $s_1'P(=(r'+y+C_s')P)$ respectively. He wants to utilize the secrecy $x_iP$ of $tag_i$ to determine whether the two messages $\{C_1, C_2, C_3(= h(x_iP, C_2))\}$ and $\{C_1', C_2', C_3'(= h(x_iP, C_2'))\}$ come from the same tag by calculating $C_2- s_1P - x_iP=kY(=kyP)$ and $C_2'- s_1'P - x_iP= k'Y(=k'yP)$. E will fail. Since $k$ and $k'$ are two random numbers without any relationship existing between them.

(g) Mutual authentication:

In our scheme, the server computes $C_4= C_2-yC_1-s_1P = x_iP$ to search $x_iP$ in the database, to check whether $tag_i$ is legal or not. Conversely, $tag_i$ also computes $h(C_2- kY)$ and checks if it is equal to the received $C_5$ (sent from the server) to see whether the server is legal. We know that only legal $tag_i$ has valid $x_iP$ in $C_2$ to let server deduce $x_iP$ and search it in the database, and only legal server has the right $x_iP$ stored in its database and sends back $h(x_iP+s_1P)$ to pass the tag's examination. Thus, our protocol can achieve mutual authentication.

## 5. Comparisons and discussions

### (a) Comparisons

In this section, we use Table 2 and Table 3 to show the comparison results of our scheme with related works in both security and efficiency. From Table 2, we can see that our scheme is the most secure. As for efficiency, our scheme is the most efficient among [26, 27, 28] in EC point multiplications with 2 extra hash operations at both

sides, as shown in Table 3.

**Table 2. Comparisons of security properties**

| | Tuyls et al. [26] | Batina et al. [27] | Lee et al. [28] | Ours |
|---|---|---|---|---|
| 1.Location privacy | No | No | Yes | Yes |
| 2.Replay attack | Yes | Yes | Yes | Yes |
| 3.Man-in-the middle attack | No | No | Yes | Yes |
| 4.Impersonation attack | Yes | No | Yes | Yes |
| 5.Physical attack | No | No | Yes | Yes |
| 6.Forward secrecy | No | No | Yes | Yes |
| 7.Mutual authentication | No | No | No | Yes |

**Table 3. Comparisons of EC point multiplications**

| | | Tuyls et al. [26] | Batina et al. [27] | Lee et al. [28] | Ours |
|---|---|---|---|---|---|
| (ECm, h) | Server | (2, 0) | (4, 0) | (3, 0) | (2, 2) |
| | Tag | (1, 0) | (2, 0) | (3, 0) | (2, 2) |

ECm: EC point multiplication, h: hash operations

**(b) Discussion**

In our original design, the server sends $C_s$, and $s_1$ to the tag and the tag stores [$x_i$, $Y$, $P$, $C$] in its memory. This makes the tag needs four EC point multiplications. To reduce tag's computational cost to two EC point multiplications, we modify $s_1$ to $s_1P$ and the tag's private key $x_i$ to $x_iP$ in our final design. Besides, we demonstrate a practical example for showing the correctness of our scheme by running a program on windows XP with Intel(R) Pentium(R) CPU 2.00 GHz platform.

## 6. Conclusion

In this paper, we adopt ECC to design an efficient RFID mutual authentication protocol under the consideration of tag's limited computational ability. After analyzing, we conclude that our scheme possesses the properties of location privacy, forward secrecy, and mutual authentication, and can resist replay attack, man-in-the middle attack, impersonation attack and physical attack. Compared to previous related works, the proposed protocol is the most secure and efficient. We have demonstrated this by using Table 2 and 3. That is, our protocol outperforms the other ECC based RFID schemes. Besides, we also have shown a practical example to demonstrate the correctness of our scheme.

## Appendix A

## Experiment Result

**A primitive element of G is *P* =**

(186432448643485022778351936862994643288850414061303718052589937079869591145717822427675982310639640467155950411685064127611068763687818597296963230218232655312750197738640995486298720270479637886923902506575504791971874009767484497897183276559708197352622981308533331665267495540315725751801122707494977153 2,

384847198117127231829572358520498064755111148950740532223855243823223351997049293435323451401649318854849414109916114857990611859761569259532450727824818870122742918399163363371415810767694690477641028005923934761548429251228286173661574304409980803400768807280860207024464468693129124308661985411675966211)

**Tag's identify is *ID*$_i$ = *X* $_i$(=$x_i$*P*) =**

(724232279839055261692843006538409821366805632933712973293928895581533043094447201256903693720629729347590016769573955056769666463922944052429462204338974955611316126161952769533032044024266305879390776111543830100501245867632925341950694410028643927649312732632528314978569279640292835535039489271102183950 2,

659619932569363133618623783900513885594210883471994262096267492502282721170541826950904200035044224627398972119634278419967662166656836635879542572406241313018689135235374645378394415172722578621912659681428828564395424337700945693661194210783286800122708532273902365676669690859608417136875162573039263204 3)

**Server's private key is *y* =** 41882037575774069350681235042483970156764663112 2

**Server's public key is *Y* (=*yP*) =**

(542904049846810916744764458569560352171490061479900862061706119808016928936317115804737439098288144341725949616255421186544551097055922496720554520259244303180352379276340236529801877616076450807422442809450598246633247553751763385082116113501627380995894894376885398286181342601039973968262699059159707823 1,

160874200292419527255584201798567324634978844803802030536554471264075076418016537584990676402171258849219521424129423551239327062011437231328195402800169218023127941780808604201890381509160698387868287119132549993643581791113213468504210509779771663275569158402911493650337737737551110326235951860070574547 0)

**Server's *C*$_s$ = 1**

**Server choose random number *r* =** 3252559105985707474023443766503840375019062893 52

**Server computes *s1P* = (*r*+*y*+*Cs*)*P* =**

(195123879036041015336270881719189342473888002039215036977684227101797911728116386255218218320503094899707635363892082696724427538890550138251913244394868448506145088849564031252293106533053890377840469033793513249163520581545088557660164286475462572620464225705007919081367124481143113450632292505047931852 5,

589033971996792221871016640689337743768862241495010357291544107910117213839635362351 6

6088631733171898427096148911352935487918193498709004678359634179036459034247210153061492796745128612664881772243599012522482360092703752740569035530004106679587344146206278047128301929657101425768256596687824898203663501600)

**Tag's $C$** = 0

**Tag choose random number $k$** = 159291522284677507776829529633828697130120254054

**Tag computes $C_1 = kP$** =

(178201580933873671632792674350722492960209348497427011344290207635354874959268696632899265118201622739657523839336549040304026221678633266461234237461786550666375585029914577992229648519776934379290559378014792545412605238573594584318894252555760411971141675784096128196555932239874092404186369772967566700,111025635900655347376024803029624238139625969813127338186698246999169350627064074304606079941527382632761493298602930379332058901246643985781116497046981762624703333140448925742887664695140458598754285881982634764573231154852169469229530973941374510023090439046101603542409453727298727773180101150320191843 1)

**Tag computes $C_2 = x1P+s1P+kY$** =

(158109441256312244815382772640153325192021115822223138379222714621953184755562168320859204299749206725335829618884124641636374664049140415780454568099300207685133369305484784424088776716528434802690253947701267763269730428297903511443907223732530698015228852792603122451962585032896183281035567517786894204 9,337042585342312400374054071530776015840000670982031445091152038227990805837897099349156659598399191433976096299423179942832822969381014409222357916668689875952628332871932807032185693258445848727677565893380461001438759971759087855699600472263727661305869512542882449053420146148016674895415111085636165981 8)

**Tag computes $C_3=h(x_iP, C_2)$ =** 96e73d889070ad0196320be96cf9c538

**Server computes $C_4=C_2-yC_1- s_1P$ = tag's ID =$x_iP$** =

(724232279839055261692843006538409821366805632933712973293928895581533043094447201256903693720629729347590016769573955056769666463922944052429462204338974955611316126161952769533032044024266305879390776111543830100501245867632925341950694410028643927649312732632528314978569279640292835535039489271102183950 2,659619932569363133618623783900513885594210883471994262096267492502282721170541826950904200035044224627398972119634278419967662166656836635879542572406241313018689135235374645378394415172722578621912659681428828564395424337700945693661194210783286800122708532273902365676669690859608417136875162573039263204 3)

**Server computes $h(C_4,C_2)$=** 96e73d889070ad0196320be96cf9c538

**Server checks $h(C_4,C_2)= C_3$**

**Server searches $x_iP$ to find out the tag's ID in DB**.

**Tag's ID** =

(724232279839055261692843006538409821366805632933712973293928895581533043094447201 2

5690369372062972934759001676957395505676966646392294405242946220433897495561131612616195276953303204402426630587939077611154383010050124586763292534195069441002864392764931273263252831497856927964029283553503948927110218395022,659619932569363133618623783900513885594210883471994262096267492502282721170541826950904200035044224627398972119634278419967662166656836635879542572406241313018689135235374645378394415172722578621912659681428828564395424337700945693661194210783286800122708532273902365676669690859608417136875162573039263204 3)

**Server computes $C_5=h(C_4+s_1P)$ =** ad506dfba13a3ccfb24fade2cd9210a9

**Tag computes $h(C_2-kY)$=** ad506dfba13a3ccfb24fade2cd9210a9

**Tag checks $h(C_2-kY) = C_5$**

## References


[1] E.-K. Ryu, and T. Takagi, "A hybrid approach for privacy-preserving RFID tags," *Computer Standards & Interfaces*, Vol. 31, 2009, pp. 812-815.

[2] Alex X. Liu, and LeRoy A. Bailey, "A privacy and authentication protocol for passive RFID tags," *Computer Communications*, Vol. 32, 2009, pp. 1194-1199.

[3] EasyCard,http://www.easycard.com.tw/.

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

[5] J.S. Cho, S.S. Yeo, and S.K. Kim, "Securing against brute-force attack：A hash-based RFID mutual authentication protocol using a secret value," *Computer Communications*, Vol. 34, 2011, pp. 391-397.

[6] NirSoft, http://www.nirsoft.net/.

[7] D. N. Duc, and K. Kim, "Defending RFID authentication protocols against DoS attacks," *Computer Communications*, 2010.

[8] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, V. Khandelwal, "Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications," *in: Proceedings of the IEEE International Conference on RFID*, April 2008, pp. 58–64.

[9] EPCglobal, http://www.epcglobalinc.org.

[10] X. Lin, R. Lu, D. Kwan, and X. (Sherman) Shen, "An RFID-based privacy-preserving children tracking scheme for large amusement parks," *Computer Networks*, 2010.

[11] S.-Y. Kang, D.-G. Lee, and I.-Y. Lee, "A study on secure RFID mutual authentication scheme in pervasive," *Computer Communications*, Vol. 31, 2008, pp. 4248-4254.

[12] M. Ohkubo, K. Suzuki and S. Kinoshita, "Cryptographic Approach to "Privacy-Friendly" tags," RFID Privacy Workshop 2003, MIT, November 2003.

[13] S. Weis, S. Sarma, R. Rivest and D. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," *Security in Pervasive Computing*, March, 2003.

[14] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, "Scalable RFID Systems: a Privacy-Preserving Protocol with Constant-Time Identification," *Security in Pervasive Computing*, March, 2003.

[15] B. Alomair, and R. Poovendran, "Privacy versus Scalability in Radio Frequency

Identification Systems," *Computer Communications*, 2010.
[16] T. V. Le, M. Burnmester, and B. de Medeiros, "Universally Composable and Forward-secure RFID Authentication and Authenticated Key Exchange," *The Proceedings of the Second ACM Symposium on Information, Computer and Communications Security*, March 2007, pp. 242–252.
[17] M. Burnmester, "Universally Composable RFID Identification and Authentication Protocols," *ACM Transactions on Information and Systems Security*, January 2009.
[18] T. Yeh, Y. Wanga, T. Kuo, and S. Wang, "Securing RFID systems conforming to EPC Class 1 Generation 2 standard," *Expert Systems with Applications*, Vol. 37, 2010, pp. 7678–7683.
[19] Y. Chen and J.S. Chou, and H.M. Sun, "A novel mutual authentication scheme based on quadratic residues," *Computer Networks*, Vol. 52, 2008, pp. 2373–2380.
[20] T. Cao, and P. Shen, "Cryptanalysis of some RFID authentication protocols," *JOURNAL OF COMMUNICATIONS*, VOL. 3, NO. 7, DECEMBER 2008.
[21] T.C. Yeh, C.H. Wua, Y.M. Tseng, "Improvement of the RFID authentication scheme based on quadratic residues," *Computer Communications*, Vol. 34, 2011, pp. 337–341.
[22] S. V. Kaya, E. Savaş, A. Levi and Ö. Erçetin, "Public key cryptography based privacy preserving multi-context RFID infrastructure," *Ad Hoc Networks* , Vol. 7, 2009, pp. 136–152.
[23] PBC Library, http://crypto.stanford.edu/pbc/.
[24] G. Gaubatz, J. P. Kaps, E. Ozturk, and B. Sunar, State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks, Proc. in the Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'05), 2005.
[25] B. Song, C. J. Mitchell, "Scalable RFID security protocols supporting tag ownership transfer," *Computer Communications* , Vol. 34, 2011, pp. 556–566.
[26] P. Tuyls, L. Batina, "RFID-tags for Anti-Counterfeiting," *Lecture Notes in Computer Science, Volume 3860, 2006, Pages 115-131*.
[27] 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*
[28] Y. K. Lee, L. Batina, D. Singelee, B. Preneel, I. 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.*
[29] Y. K. Lee, L. Batina, I. Verbauwhede, "EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID authentication protocol," *IEEE International Conference on RFID, 2008, Pages 97-104.*
[30] Y. K. Lee, L. Batina, I. Verbauwhede, "Untraceable RFID authentication protocols: Revision of EC-RAC," *IEEE International Conference on RFID, 2009, Pages 178-185.*
[31] Y. K. Lee, K. Sakiyama, and I. Verbauwhede, Elliptic-Curve-Based Security Processor for RFID, *IEEE Trans. on Computers*, 57(11), Nov., 2008.

[32] H. Y. Chien, “SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity” *IEEE Trans. Dependable and Secure Computing*, vol. 4, no. 4, 2007, pp. 337-340.