


# Elliptic Curve Cryptography-Based RFID Authentication Resisting Active Tracking

Hung-Yu Chien<sup>1</sup> 

Published online: 28 September 2016  
© Springer Science+Business Media New York 2016

**Abstract** The challenge of authentication for radio frequency identification (RFID) with low computing capacities call for computation-efficient authentication that can achieve mutual authentication, anonymity, and tracking resistance. The excellent performance of elliptic curve cryptography (ECC) including its strong security, its small key size and efficient computation has attracted many researchers' attention in designing RFID authentication. Recently there are several promising ECC-based RFID authentication schemes aimed at achieving the above functions. Despite of their good performance in terms of computation and general security properties, we find that they all fall in the same security pitfall-being vulnerable to active tracking. In this paper, we identify these weaknesses and then propose a new ECC-based RFID authentication which conquers the weakness and even improves the computational performance.

**Keywords** RFID · Authentication · Anonymity · Elliptic curve cryptography · Tracking · Diffie–Hellman

## 1 Introduction

Radio frequency identification (RFID) systems, thanks to their low cost and their convenience in identifying an object without physical contact, have found many applications in manufacturing, supply chain management, parking garage management, and inventory control. RFID is also one of the key technologies that facilitate the development of Internet of Things (IoT). An RFID system consist of radio frequency (RF) tags, readers and backend servers, where readers can inquire tags of their identifications and contents by

---

✉ Hung-Yu Chien  
hychien@ncnu.edu.tw

<sup>1</sup> Department of Information Management, National Chi-Nan University, Nantou, Taiwan, ROC

broadcasting an RF signal, and then read or update the corresponding data in backend servers.

The widespread deployment of RFID systems not only enhances the efficiency and convenience in our daily life but also exposes potential security threats and risks either to corporations or individuals. Forging of participated entity (either tag or reader) is one key threat, and disclosure of sensitive data is another. In addition, as the co-related information of tags labeled on products might be utilized to reveal an user's identity, his location, his movement, and his habits. Therefore, a desirable RFID authentication solution should protect identity privacy (anonymity) and tracking resistance (un-linkability). However, as most popular tags (like Mifare, Suicard, ISO 15693, EPC Gen2 [1]) have cost pressure from the market, they all call for computationally lighter algorithms.

The RFID authentication has been extensively studied like [2–19], and readers are referred to Avoine's RFID Security and Privacy Lounge [2] for a comprehensive list of related works. Among them, solution based on ECC has recently attracted many researchers' attention [20–27], owing to Elliptic Curves Cryptography's (ECC) excellent performance in terms of strong security, smaller key size and lighter computation. Some [20–23] of these schemes achieved only basic authentication functions while others [27] aimed at achieving full-fledged security functions like mutual authentication, anonymity, tracking resistance and denial-of-service (DOS) attack resistance. Liao and Hsiao [27] recently did a critical survey of these ECC-based schemes and proposed a security-improved solution. However, we find one key security weakness of these schemes and it has been neglected: most of the previous schemes only considered passive tracking and fell victim to active tracking attack. An attacker in passive-tracking attacks passively eavesdrops on the transmission to track RFID tags, while an active-tracking attacker would track RFID tags through various active involvements like intercepting, modification, replaying message, and so on. As RFID communicates via wireless radio frequency and the devices are cheap, it is practically feasible to conduct various active attacks, and these threats should be carefully deterred.

In this paper, we describe our active-tracking attacks on several recent publications and then propose our scheme to conquer the weakness. The rest of this paper is organized as follows. In Sect. 2, we introduce some preliminaries of ECC, related hard problems and bilinear pairing computations which will be used to facilitate the attacks. We review several ECC-based schemes in Sect. 3.1, and show the attacks in Sect. 3.2. In Sect. 4, we propose our new scheme, and the security analysis and performance evaluation are conducted in Sect. 5. Finally, Sect. 6 states our conclusion.

## 2 Preliminaries

In this section, we give some preliminaries on ECC, related hard problems, and bilinear pairing.

**Elliptic curves over  $GF(p)$ :** A non-supersingular elliptic curve  $E(Fp)$  is the set of points  $P = (x, y)$ , for  $x, y \in Z_p$  satisfying the equation  $y^2 \equiv x^3 + ax + b \pmod{p}$ , where  $a, b \in Z_p$  are constants such that  $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ , together with the point  $O$  called the *point at infinity*. Two points  $P = (x_1, y_1)$  and  $Q = (x_2, y_2)$  on the elliptic curve  $E$  can be added together using the following rule: if  $x_2 = x_1$  and  $y_2 = -y_1$ , then  $P + Q = O$ ; otherwise,  $P + Q = (x_3, y_3)$  where:  $x_3 = \lambda^2 - x_1 - x_2 \pmod{p}$ ,  $y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p}$ , and  $\lambda = (y_2 - y_1)/(x_2 - x_1)$  if  $P \neq Q$  or  $\lambda = (3x_1^2 + a)/(2y_1)$  if  $P = Q$ .

**Definition 1** The computational elliptic curve Diffie–Hellman problem (ECDHP) [28] is: given an elliptic curve over a finite field  $F_p$ , a point  $P \in E(F_p)$  of order  $q$ , and points  $A = aP, B = bP \in \langle P \rangle$ , find the point  $C = abP$ .

**Definition 2** The elliptic curve discrete logarithm problem (ECDLP) [28] is: given an elliptic curve over a finite field  $F_p$ , two points  $P, Q \in E(F_p)$ , find a number  $k$  such that  $Q = kP$ .

It is believed that both the ECDLP and the ECDHP are hard problems for proper parameter setting, and many security systems have been proposed based on them [28].

**Definition 3** (*Non-degenerate, bilinear, computable map*) [29] Let  $G_1$  and  $G_2$  be cyclic groups of prime order  $q$ , where  $G_1$  is an additive group on elliptic curves and  $G_2$  is multiplicative. Let  $e: G_1 \times G_1 \rightarrow G_2$  be a map with the following properties below.

- (1) Non-degenerate: There exists  $X, Y \in G_1$  such that  $e(X, Y) \neq 1$ .
- (2) Bilinear:  $e(X_1 + X_2, Y) = e(X_1, Y) \cdot e(X_2, Y)$  and  $e(X, Y_1 + Y_2) = e(X, Y_1) \cdot e(X, Y_2)$ .  
 Computable: There is an efficient algorithm for evaluating  $e$ .
- (3) Computability: There exist efficient algorithms to compute  $e(P, Q)$  for all  $P, Q \in G_1$ .

### 3 Security Weaknesses of Several ECC-Based RFID Authentication Schemes

In this section, we review two ECC-based RFID authentication schemes and demonstrate active tracking attacks and other weaknesses of them.

We introduce the notations as follows, and we will omit the mod  $q$  operation to simplify the presentation when the context is clear.

$E(F_p), P, q$ :  $P \in E(F_p)$  is a generator point of a group over  $E(F_p)$  of order  $q$ .

$h()$ : cryptographic hash function.

$T, S$ :  $T$  and  $S$  respectively denote the tag and the server.

$ID_T, ID_S$ :  $ID_T$  and  $ID_S$  respectively denote the identity of the tag and that of the server.

$x_T, x_S, P_C, P_S$ :  $x_T, x_S \in \mathbb{Z}_q^*$  respectively denote the private key of  $T$  and that of  $S$ .  
 $P_T = x_T P$  and  $P_S = x_S P$  respectively denote their corresponding public keys.

$r_1, r_2, r_3, R_1, R_2, R_3$ :  $r_1, r_2, r_3 \in \mathbb{Z}_q^*$  respectively denote ephemeral private keys.  $R_1 = r_1 P, R_2 = r_2 P, R_3 = r_3 P$  denote their corresponding public keys.

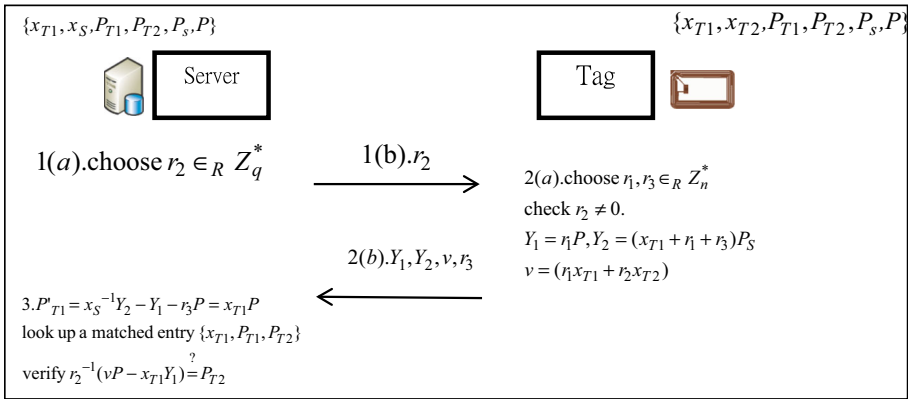
$\oplus, \parallel$ :  $\oplus$  denotes exclusive OR operation, and  $\parallel$  denotes concatenation. Here we abusively use the notation  $\oplus$  between two elliptic curve points to represent  $(x_1, y_1) \oplus (x_2, y_2) = (x_1 \oplus x_2, y_1 \oplus y_2)$ .

#### 3.1 Attacks on Zhang et al.’s Scheme [21]

##### 3.1.1 Zhang et al.’s Scheme

Initially, tag  $T$  owns two private keys  $x_{T1}, x_{T2}$  and two public keys  $P_{T1} = x_{T1} P, P_{T2} = x_{T2} P$ , and the server  $S$  owns its private key  $x_S$  and its public key  $P_S = x_S P$ . The server keeps  $\{x_{T1}, x_S, P_{T1}, P_{T2}, P_S, P\}$ , and tag  $T$  keeps  $\{x_{T1}, x_{T2}, P, P_S\}$ .

During authentication process,  $T$  and  $S$  perform the following steps. Figure 1 depicts the process.



**Fig. 1** Zheng et al.'s scheme

1.  $S \rightarrow T: r_2$   
The server chooses a random number  $r_2 \in_R Z_q^*$ , and sends it as a challenge to  $T$ .
2.  $T \rightarrow S: Y_1, Y_2, v, r_3$   
Upon receiving the challenge,  $T$  chooses two random numbers  $r_1, r_3 \in_R Z_q^*$ , validates whether  $r_2 \neq 0$ , and then computes  $Y_1 = r_1P$ ,  $Y_2 = (x_{T1} + r_1 + r_3)P_S$  and  $v = (r_1x_{T1} + r_2x_{T2}) \bmod q$ .
3.  $S$ :  
 $S$  first computes  $P^*_{T1} = x_S^{-1}Y_2 - Y_1 - r_3P = (x_{T1} + r_1 + r_3)P - r_1P - r_3P = x_{T1}P$  and uses  $P_{T1}'$  to look up a matched entry  $\{x_{T1}, P_{T1}, P_{T2}\}$ . It uses the data from the matched entry to verify whether the equation  $r_2^{-1}(vP - x_{T1}Y_1) \stackrel{?}{=} P_{T2}$ . If the equation holds, then it accepts the tag.

### 3.1.2 Security Weaknesses

Apparently, the scheme only provided unilateral authentication of tag to server. We now introduce an active-tracking attack as follows. Let Eve be the attacker. She sends the same challenge  $r_2$  to tags it encounters. If the same tag  $T$  receives the same challenge  $r_2$  twice, then it will respond with  $\{Y_1 = r_1P, Y_2 = (x_{T1} + r_1 + r_3)P_S, v = (r_1x_{T1} + r_2x_{T2}), r_3\}$  in one session and  $\{Y_1' = r_1'P, Y_2' = (x_{T1} + r_1' + r_3')P_S, v' = (r_1'x_{T1} + r_2x_{T2}), r_3'\}$  in the other session, where  $(r_1, r_3)$  and  $(r_1', r_3')$  are respectively the random numbers chosen by  $T$  in the two sessions.

Now Eve computes the values  $Y_1 - Y_1' = (r_1 - r_1')P$  and  $v - v' = (r_1x_{T1} + r_2x_{T2}) - (r_1'x_{T1} + r_2x_{T2}) = (r_1 - r_1')x_{T1}$ . Next, she checks whether the equation  $e(Y_1 - Y_1', P_{T1}) \stackrel{?}{=} e((v - v')P, P)$  holds to validate whether the two sessions came from the same tag  $T$ . The above equation should hold if the transcripts came from the same tag  $T$ , because  $e(Y_1 - Y_1', P_{T1}) = e((r_1 - r_1')P, x_{T1}P) = e((r_1 - r_1')P, P)^{x_{T1}} = e((r_1 - r_1')x_{T1}P, P) = e((v - v')P, P)$ . That is, the scheme falls victim to our active-tracking attack.

### 3.2 Attacks on Liao–Hsiao’s Scheme [27]

#### 3.2.1 Liao–Hsiao’s Scheme

Initially, tag  $T$  owns one private key  $x_T$  and one public key  $P_T = x_T P$ , and the server  $S$  owns its private key  $x_S$  and its public key  $P_S = x_S P$ . The server keeps  $\{x_T, x_S, P_T, P_S, P\}$ , and tag  $T$  keeps  $\{x_T, P_T, P, P_S\}$ .

During the authentication process,  $T$  and  $S$  perform the following steps. The process is also depicted in Fig. 2.

1.  $S \rightarrow T: R_2$   
The server chooses a random number  $r_2 \in_R Z_q^*$ , and sends  $R_2 = r_2 P$  as a challenge to  $T$ .
2.  $T \rightarrow S: R_1, Auth_T$   
Upon receiving the challenge,  $T$  chooses one random numbers  $r_1 \in_R Z_q^*$ , and computes  $R_1 = r_1 P$ ,  $TK_{T1} = r_1 R_2$ ,  $TK_{T2} = r_1 P_S$  and  $Auth_T = P_T + TK_{T1} + TK_{T2}$ .
3.  $S \rightarrow T: Auth_S$   
 $S$  first computes  $TK_{S1} = r_2 R_1$ ,  $TK_{S2} = x_S R_1$  and  $Auth_T - TK_{S1} - TK_{S2} = P_T$ . It uses  $P_T$  to look up a matched entry in its database. If a matched entry is found, then it accepts the tag and computes  $Auth_S = x_T R_1 + r_2 P_T$ . It sends  $Auth_S$  to the tag.
4.  $T$ :  
The tag checks whether  $Auth_S \stackrel{?}{=} r_1 P_T + x_T R_2$  holds. If it holds, then the tag accepts the server.

#### 3.2.2 Security Weaknesses

3.2.2.1 Active-Tracking Attack Using Two Sessions We now introduce an active-tracking attack using two sessions. Let Eve be the attacker. She chooses an integer  $r_2$  and sends the

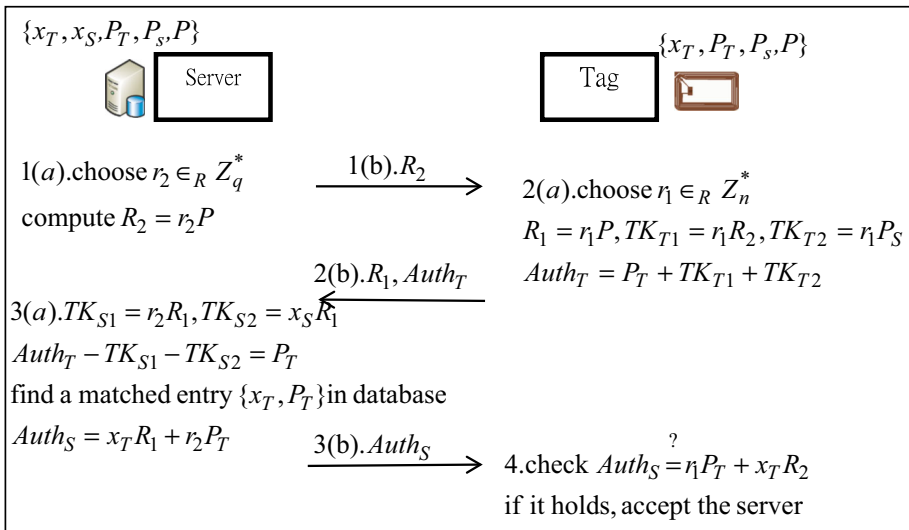


Fig. 2 Liao–Hsiao’s scheme

same challenge  $R_2 = r_2P$  to tags. If the same tag  $T$  receives the same challenge  $R_2$  twice, then it will respond with  $\{R_1, Auth_T = P_T + TK_{T1} + TK_{T2}\}$  in one session and  $\{R_1', Auth_{T'} = P_T + TK_{T1}' + TK_{T2}'\}$  in the other session, where  $R_1 = r_1P, R_1' = r_1'P, TK_{T1} = r_1R_2, TK_{T2} = r_1P_S, TK_{T1}' = r_1'R_2, TK_{T2}' = r_1'P_S$  and  $(r_1, r_1')$  are respectively the random numbers chosen by  $T$  in the two sessions.

Now Eve computes the values  $r_2R_1 = TK_{T1}$  and  $r_2R_1' = TK_{T1}'$ . Next she computes  $Auth_T - TK_{T1} = P_T + TK_{T2}, Auth_{T'} - TK_{T1}' = P_T + TK_{T2}'$  and  $P_T + TK_{T2} - (P_T + TK_{T2}') = TK_{T2} - TK_{T2}' = (r_1 - r_1')P_S$ . Finally, she checks whether the equation  $e((r_1 - r_1')P_S, P) \stackrel{?}{=} e(R_1 - R_1', P_S)$  holds to validate whether the two sessions came from the same tag  $T$ . The above equation should hold if the transcripts came from the same tag  $T$ , because  $e((r_1 - r_1')P_S, P) = e((r_1 - r_1')x_S P, P) = e((r_1 - r_1')P, x_S P) = e(R_1 - R_1', P_S)$ . That is, the scheme falls victim to our active-tracking attack.

**3.2.2.2 Passive-Tracking Attack Using Two Sessions** We further show our passive-tracking attack using two sessions, where Eve, instead of actively involving the communications, only passively eavesdrops on the communications. From the eavesdropped data, she gets  $\{Auth_S = x_T R_1 + r_2 P_T\}$  in one session and  $\{Auth_{S'} = x_T R_1' + r_2' P_T\}$  in the other. Now she computes  $Auth_S - Auth_{S'} = (x_T R_1 + r_2 P_T) - (x_T R_1' + r_2' P_T) = x_T(r_1 - r_1')P + (r_2 - r_2')P_T$ . Finally, she checks whether the equation  $e(Auth_S - Auth_{S'}, P) \stackrel{?}{=} e(R_1 - R_1', P_T) \cdot e(R_2 - R_2', P_T)$  holds to validate whether the two sessions came from the same tag  $T$ . The equation should hold if they came from the same tag, because  $e(Auth_S - Auth_{S'}, P) = e(x_T(r_1 - r_1')P + (r_2 - r_2')P_T, P) = e(x_T(r_1 - r_1')P, P) \cdot e((r_2 - r_2')P_T, P) = e((r_1 - r_1')P, x_T P) \cdot e((r_2 - r_2')P, x_T P) = e(R_1 - R_1', P_T) \cdot e(R_2 - R_2', P_T)$ . The scheme is vulnerable to the passive-tracking attack.

**3.2.2.3 Impersonating a Tag** The scheme authenticates a tag by checking whether the tag can form a valid  $Auth_T = P_T + TK_{T1} + TK_{T2}$  value. But, we should notice that  $P_T$  is a public key, and  $TK_{T1} = r_1R_2, TK_{T2} = r_1P_S$  could be computed by an attacker using his chosen random number  $r_1$ . That is, an attacker can forge valid  $Auth_T$ . The scheme fails in authenticating a tag.

**3.2.2.4 Disclosing the Tag's identity Using One Active-Involved Session** Now we show how to disclose a tag's identity using one simple probing. Eve just chooses a random number  $r_2 \in \mathbb{R}Z_q^*$ , and sends  $R_2 = r_2P$  as a challenge to  $T$ .  $T$  will respond with  $R_1 = r_1P, Auth_T = P_T + TK_{T1} + TK_{T2}$ , where  $TK_{T1} = r_1R_2$  and  $TK_{T2} = r_1P_S$ . Next she computes  $Auth_T - r_2R_1 = P_T + TK_{T1} + TK_{T2} - r_2R_1 = P_T + TK_{T2}$ . Now she iteratively picks up one potential tag  $T'$  with public key  $P_{T'}$  from its database and checks whether the equation  $e(P_T + TK_{T2} - P_{T'}, P) \stackrel{?}{=} e(R_1, P_S)$  holds. The equation should hold, if  $P_{T'} = P_T$ , as  $e(P_T + TK_{T2} - P_{T'}, P) = e(TK_{T2}, P) = e(r_1P_S, P) = e(r_1P, x_S P) = e(R_1, P_S)$ . If the verification holds, then the attacker can identify the identity of the tag. The scheme fails in protecting tag's anonymity.

### 4 The Proposed Scheme

Now we propose a new scheme to improve the security properties. Initially, tag  $T$  owns one private key  $x_T$ , one public key  $P_T = x_T P$  and one secret key  $K_T = x_T P_S = x_T x_S P$  with the server; the server  $S$  owns its private key  $x_S$  and its public key  $P_S = x_S P$ . The server keeps  $\{x_S, P_T, P_S, P, K_T, h()\}$ , and tag  $T$  keeps  $\{x_T, P_T, P, P_S, K_T, h()\}$ . Please note the server in our scheme does not keep tag's secret keys.

$T$  and  $S$  perform the following steps during the authentication, and Fig. 3 depicts the process.

1.  $S \rightarrow T: R_2$   
The server chooses a random number  $r_2 \in \mathbb{Z}_q^*$ , and sends  $R_2 = r_2 P$  as a challenge to  $T$ .
2.  $T \rightarrow S: R_1, Auth_{T1}, Auth_{T2}$   
Upon receiving the challenge,  $T$  chooses one random number  $r_1 \in \mathbb{Z}_q^*$ , and computes  $R_1 = r_1 P$ ,  $TK_{T1} = r_1 R_2$ ,  $TK_{T2} = r_1 P_S$ ,  $Auth_{T1} = (P_T + TK_{T1}) \oplus h(TK_{T2})$ , and  $Auth_{T2} = h(TK_{T1} \oplus K_T)$ .
3.  $S \rightarrow T: Auth_S$   
 $S$  first computes  $TK_{S1} = r_2 R_1$ ,  $TK_{S2} = x_S R_1$  and  $(Auth_{T1} \oplus h(TK_{S2})) - TK_{S1} = P_T$ . It uses  $P_T$  to look up a matched entry in its database. If a matched entry is found, then it verifies the validity of  $Auth_{T2}$ . If the verification succeeds, then it accepts the tag and computes  $Auth_S = h(TK_{S1} \oplus TK_{S2})$ . It sends  $Auth_S$  to the tag.
4.  $T$ :  
The tag checks whether  $Auth_S = h(TK_{S1} \oplus TK_{S2})$  holds. If it holds, then the tag accepts the server.  
The final session key could be computed as  $sess = h(P_T, P_S, TK_{S1})$ .

### 5 Security Analysis and Performance Evaluation

We analyze the security properties of our scheme in Sect. 5.1, and then evaluate the performance in Sect. 5.2.

#### 5.1 Security Analysis

We analyze the security properties of the proposed scheme as follows.

*Mutual authentication* The authentication of a tag depends on the validity of  $Auth_{T2}$ . To generate a valid  $Auth_{T2} = h(TK_{T1} \oplus K_T)$ , it needs the knowledge of the secret key  $K_T$  with the fresh, random challenge  $TK_{T1}$ . It ensures only a genuine tag can generate the value. The authentication of the server depends on the validity of  $Auth_S = h(TK_{S1} \oplus TK_{S2})$ , where  $TK_{T1} = r_1 R_2 = r_2 R_1$  is the ephemeral Diffie–Hellman key depending on tag's and server's challenges, and  $TK_{T2} = r_1 P_S = x_S R_1$  requires the server to demonstrate its knowledge of  $x_S$ . This ensures the authenticity of the server.

*Anonymity of the tag* Among the transmitted data, only the value  $Auth_{T1} = (P_T + TK_{T1}) \oplus h(TK_{T2})$  involves tag-specific public key  $P_T$ .  $Auth_{T1}$  can be viewed as an encryption using the two keys  $TK_{T1}$  and  $TK_{T2}$ , where the computation  $TK_{T1} = r_1 R_2 = r_2 R_1$  needs either tag's random secret or the server's random secret, and the computation of

$TK_{T2} = r_1P_S$  needs either the knowledge of tag’s secret challenge  $r_1$  or the server’s secret key. This ensures that only the genuine server could derive the tag’s public key  $P_T$ .

**Resistance to tracking** To track a tag either passively or actively, one needs to link two sessions to the same source or to differentiate one session from others. In our protocol,  $R_1, R_2$  are random and fresh in each session, and the values  $Auth_{T2} = h(TK_{T1} \oplus K_T), Auth_S = h(TK_{S1} \oplus TK_{S2})$  are hashing of secret key and the ephemeral Diffie–Hellman (D–H) key  $TK_{T1}$  and  $TK_{S2}$ ; therefore, an outsider who has no knowledge of  $(K_T, TK_{S2})$  cannot infer any clue that whether these values came from any two sessions. The value  $Auth_{T1} = (P_T + TK_{T1}) \oplus h(TK_{T2})$  encrypts a tag’s public key  $P_T$  using the key  $h(TK_{T2})$ , where the computation of  $TK_{T2} = r_1P_S$  needs either the knowledge of tag’s secret challenge  $r_1$  or the server’s secret key: it ensures that only the server or the sender itself is capable of calculating the value. This ensures the protection of the transmission  $P_T$  and the possible linking of any two sessions.

**Forward secrecy** The session key is defined as  $sess = h(P_T, P_S, TK_{S1})$ , where  $TK_{T1} = r_1R_2 = r_2R_1$  is an ephemeral D–H key. So even assume that the long-term private keys of the tag and the server are compromised some day later, the previous session keys of our scheme are still secure. This ensures the forward secrecy property.

### 5.2 Performance Evaluation

We make a comparison of the performance of ECC-based schemes in Table 1. First, we specify what kind of authentication a scheme tried to provide: unilateral or mutual. Among those schemes in Table 1, only our scheme and Liao–Hsiao’s scheme aimed at providing mutual authentication. Next, we concern the security properties: vulnerability to passive tracing attack or active tracing attack, disclosing tag’s identity, and impersonation of tag. From the table, we can see that only our scheme can resist all the attacks while others are vulnerable to some of the threats.

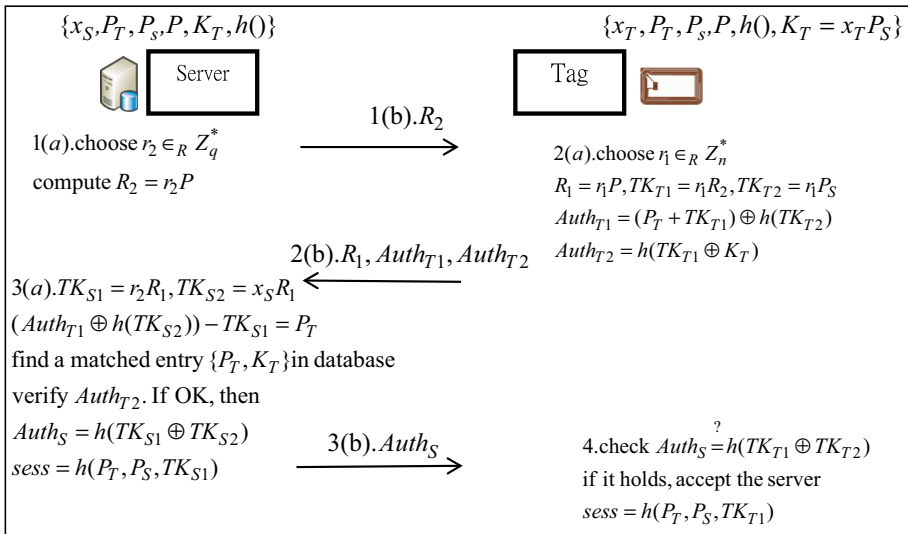


Fig. 3 New scheme



**Table 1** Summary of comparison among ECC-based RFID authentication

	Lee et al. [20]	Zhang [21]	Tuyuls-Batina [22]	Batina et al. [23]	Liao-Hsiao [27]	Our
Unilateral or mutual authentication	Unilateral	Unilateral	Unilateral	Unilateral	Mutual	Mutual
Vulnerability of Passive tracing	No	No	Yes	Yes	Yes	No
Vulnerability of active tracing	Yes	Yes	Yes	Yes	Yes	No
Disclosing tag's identity	Yes	Yes	Yes	Yes	Yes	No
Impersonation of tag	Yes	No	No	No	Yes	No
Server's cost for finding a matched tag <sup>a</sup>	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$
Tag's computation	$2T_{EM} + 2T_{mdl,q}$	$2T_{EM} + 2T_{mdl,q}$	$1T_{EM} + 1T_{mdl,q}$	$2T_{EM} + 1T_{EA} + 2T_{mdl,q}$	$5T_{EM} + 3T_{EA}$	$2T_{EM} + 1T_{EA} + 3T_h$

<sup>a</sup>  $O(1)$  means that the scheme just needs to perform few numbers of computations to identify a tag while  $O(n)$  means that the number of computations needed to identify a tag is proportional to the number of tags in the database, where  $n$  is the number of tags in the database

Now we discuss the computational cost for a server to identify a tag. Some schemes [20, 21, 27] and our scheme only need to perform few calculations to identify a tag and the number of calculations is independent of the number of potential tags. Here we use  $O(1)$  to denote this notation. While the complexity of computation for identifying a tag in other schemes like [22, 23] is proportional to the number of potential tags. Here we use  $O(n)$  to denote the notation, where  $n$  is the number of tags in the database. Both our scheme and Liao–Hsiao’s mutual authentication scheme are  $O(1)$  in this context.

Finally, we evaluate the computational complexity of tag. Here we only count those computations un-negligible but neglect those light computations like exclusive OR and simple field addition.  $T_{EM}$  denotes the time complexity of one elliptic curve point multiplication,  $T_{EA}$  denotes that for one elliptic curve point addition,  $T_h$  denotes that for one hash operation,  $T_{mul,q}$  denotes that for one multiplication in Field  $q$ . Our scheme needs  $2T_{EM} + 1T_{EA} + 3T_h$ , and Liao–Hsiao’s mutual authentication scheme needs  $5T_{EM} + 3T_{EA}$ . Apparently, our scheme demands much lighter computation than its mutual ECC-based counterpart [27].

To further assess the computational performance, we evaluate the computational cost under the practical setting from NSA [30] and the algebra equations of elliptic curve operations [31]. The security of ECC with 160-bit key is equivalent to that of RSA with 1024-bit key or D–H algorithm with 1024-bit key. Under the above figures,  $T_{mul,p}$  (the time complexity of a field multiplication in  $Z_p$ , where  $p$  is 1024-bit) is 41 times  $T_{mul,q}$  (the time complexity of field multiplication in  $Z_q$ , where  $q$  is 160-bit),  $T_{EM} \sim 29T_{mul,p}$ , and  $T_{EM} \sim 241 T_{EA}$ , where  $\sim$  means “roughly equal”. To simplify the comparison and get an insight of the computational performance, we can focus on the number of ECC point multiplication, point addition, modular exponentiation and modular multiplication only because the other operations are not computationally significant. In this simplification, the tag in our scheme needs  $2T_{EM} + T_{EA} \sim 58.12T_{mul,p}$ , the tag in [27] needs  $5T_{EM} + 3T_{EA} \sim 145.36T_{mul,p}$ . Based on these figures, we can get an insight that the tag in our scheme only takes roughly 39 % computational complexity of Liao–Hsiao’s ECC-based scheme [27].

In summary, our scheme owns better performance than other schemes in terms of security, server’s computational performance, and tag’s computational performance.

## 6 Conclusion

In this paper, we have shown the security weaknesses of Zhang et al.’s scheme and Liao–Hsiao’s scheme, and we highlight that active-tracking attack is one powerful attack that compromises all previous ECC-based scheme. We have proposed a new scheme to conquer the security weaknesses. Compared to Liao–Hsiao’s mutual authentication scheme, our scheme not only improves the security but also needs only 39 % tag’s computational complexity.

**Acknowledgments** This project is partially supported by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant No. MOST 105-2221-E-260-014.

## References

1. EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz, Version 1.2.0. EPCglobal Inc., October 2008. [www.gs1.org](http://www.gs1.org).

2. Avoine's RFID Security & Privacy Lounge. <http://www.avoine.net/rfid/>.
3. Avoine, G., Dysli, E., & Oechslin, P. (2005). Reducing time complexity in RFID systems. In *The 12th annual workshop on selected areas in cryptography (SAC)*.
4. Juels, A., & Weis, S. A. (2005). Authenticating pervasive devices with human protocols. In *Advances in cryptography—Crypto '05, LNCS* (Vol. 3126, pp. 293–308). Berlin: Springer.
5. Duc, D. N., Park, J., Lee, H., & Kim, K. (2006). Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning. In *The 2006 symposium on cryptography and information security*.
6. Juels, A. (2005). Strengthening EPC tag against cloning. In *Proceedings of WiSe '05*.
7. Yang, J., Park, J., Lee, H., Ren, K., & Kim, K. (2005). Mutual authentication protocol for low-cost RFID. In *Handout of the ECRYPT Workshop on RFID and Lightweight Crypto*.
8. Hopper, N. J., & Blum, M. (2001). Secure human identification protocols. In *Proceedings of in advances in cryptography—ASIACRYPT 2001, LNCS* (Vol. 2248, pp. 52–66).
9. Piramuthu, S. (2006). HB and related lightweight authentication protocols for secure RFID tag/reader authentication. In *COLLECTeR Europe Conference*.
10. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2006). LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In *Proceedings of 2nd Workshop on RFID Security*.
11. Li, T., & Wang, G. (2007). Security analysis of two ultra-lightweight RFID authentication protocols. In *IFIP SEC 2007*.
12. Li, T., & Deng, R. H. (2007). Vulnerability analysis of EMAP—An efficient RFID mutual authentication protocol. In *The second international conference on availability, reliability and security (AREs 2007)*, 2007 Vienna.
13. Chien, H. Y. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transactions on Dependable and Secure Computing*, 4(4), 337–340.
14. Karthikeyan, S., & Nesterenko, M. (2005). RFID security without extensive cryptography. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, Alexandria, VA, USA, pp. 63–67, Nov., 2005.
15. Molnar, D., & Wagner, D. (2004). Privacy and security in library RFID: Issues, practices, and architectures. In *Proceedings of conference on computer and communications security—CCS'04*, Washington, DC, USA, pp. 210–219, Oct., 2004.
16. Ohkubo, M., Suzuki, K., & Kinoshita, S. (2003). *Cryptographic approach to 'Privacy-Friendly' tags*. Presented at the RFID Privacy Workshop (MIT, Cambridge, MA, Nov. 15 2003); [rfidprivacy.ex.com/2003/agenda.php](http://rfidprivacy.ex.com/2003/agenda.php).
17. Rhee, K., Kwak, J., Kim, S., & Won, D. (2005). Challenge-response based RFID authentication protocol for distributed database environment. In *Proceedings of international conference on security in pervasive computing—SPC*, Berlin, Germany, LNCS (Vol. 3450, pp. 70–84).
18. Chien, H. Y., & Laih, C. S. (2009). ECC-based lightweight authentication protocol with untraceability for low-cost RFID. *Journal of Parallel and Distributed Computing*, 69, 848–853.
19. Chien, H. Y. (2013). Combining Rabin cryptosystem and error correction codes to facilitate anonymous authentication with un-traceability for low-end devices. *Computer Networks*, 57(14), 2705–2717.
20. Lee, Y. K., Batina, L., & Verbauwhede, I. (2008). EC-RAC (ECDLP based randomized access control): Provably secure RFID authentication protocol. In *IEEE International Conference on RFID*, pp. 97–104.
21. Zhang, X. L., Li, L. S., Wu, Y., & Zhang, Q. (2011). An ECDLP-based randomized key RFID authentication protocol. In *2011 international conference on network computing and information security*.
22. Tuyls, P., & Batina, L. (2006). RFID-tags for anti-counterfeiting. *Lecture Notes in Computer Science*, 3860, 115–131.
23. Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., Verbauwhede, I. (2007). Public-key cryptography for RFID-tags. In *Fifth IEEE international conference on pervasive computing and communications workshops, 2007*, pp. 217–222.
24. Deursen, T., Radomirović, S. (2008). Attacks on RFID protocols. In *Cryptology ePrint Archive: listing for 2008* (2008/310), 2008.
25. Bringer, J., Chabanne, H., & Icart, T. (2008). Cryptanalysis of EC-RAC, a RFID identification protocol. In *International conference on cryptology and network security—CANS'08, Lecture Notes in Computer Science*. Berlin: Springer.
26. Godor, G., & Imre, S. (2011). Elliptic curve cryptography based authentication protocol for low-cost RFID tags. In *2011 IEEE international conference on RFID-technologies and applications*.
27. Liao, Y. P., & Hsiao, C. M. (2014). A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol. *Ad Hoc Networks*, 18, 133–146.

28. Jurisic, A., & Menezes, A. J. (1997). *Elliptic curves and cryptography*. Certicom Whitepaper.
29. Boneh, D., & Franklin, M. (2001). Identity-based encryption from the Weil pairing. In *Proceedings of Crypto'01*, Santa Barbara, California, USA, 19–23 August, *LNCS* (Vol. 2139, pp. 213–229). Berlin: Springer.
30. National Security Agency, the US, *The case for elliptic curve cryptography*. [https://www.nsa.gov/business/programs/elliptic\\_curve.shtml](https://www.nsa.gov/business/programs/elliptic_curve.shtml). Accessed December 25, 2014.
31. Jurisic, A., & Menezes A. J. (1997). *Elliptic curves and cryptography*. Certicom Whitepaper.



**Hung-Yu Chien** received the B.S. degree in Computer Science from NCTU, Taiwan, 1988, the M.S. degree in Computer and Information Engineering from NTU, Taiwan, 1990, and the doctoral degree in applied mathematics at NCHU 2002. He was an assistant researcher at TL, MOTC, Taiwan, during 1992–1995, the director of Computer Center at Nan-Kei College, an associate professor of Chaoyang University of Technology during 200309–200609, a professor of National Chi Nan University since 199808, and the department head of the Information Management department 2008.8–2011.7. He won the best paper award of the 2007 IFIP ICEUC and of the 2010 JWIS. He won the most cited paper award from NCNU 2012 and 2015. He won the outstanding faculty research awards of Taiwan National Science Council since 2011. He is one of the members of the editorial team of several international journals. He was a member of the Chinese Association for Information Security, an IEEE member, and an ACM member. His research interests include cryptography, networking, network security, ontology and Internet-of-Things.