Research Article

An Elliptic Curve Cryptography-Based RFID Authentication Securing E-Health System

Chin-I Lee and Hung-Yu Chien

1Department of Information Management, Ling Tung University, No. 1, Ling Tung Road, Taichung 408, Taiwan
2Department of Information Management, National Chi Nan University, No. 1, University Road, Puli, Nantou 545, Taiwan

Correspondence should be addressed to Hung-Yu Chien; hychien@ncnu.edu.tw

Received 11 September 2015; Accepted 6 December 2015

Mobile healthcare (M-health) systems can monitor the patients’ conditions remotely and provide the patients and doctors with access to electronic medical records. In the development of M-health systems, Radio Frequency Identification (RFID) technology plays an important role for identifying and accessing patients and objects. Therefore, securely accessing these RFID tags and systems is critical to the success of M-health systems [1, 2].

In a RFID system, there are three types of roles: RFID tags, RFID readers, and a back-end server. Each tag has a unique number which is used to identify a RFID-tagged product. To obtain data from a tag, a reader first issues a query to the tag and then forwards the received information provided by the tag to a back-end server. The back-end server maintains a database of the information of tags and their labelled products. However, since a tag automatically responds to any readers’ queries via radio signal, the owner of the tagged product is even unaware of this action. If the tag transmits a fixed value in response to readers’ queries, it raises potential privacy threats to the labelled objects and the owner’s location.

Privacy protection in a RFID system is investigated in two respects. One is anonymity; the other is tracking attack resistance. The former is to provide confidentiality of tag’s identity such that an unauthorized observer cannot learn the identity of the tag. The latter is to provide unlinkability of any two RFID transmission sessions; that is, given any two RFID transactions, an attacker cannot tell whether the two transactions came from the same tag or not. Tracking attack could be classified into two categories: passive tracking attack and active tracking attack. The passive tracking attack is that an adversary tries to distinguish whether two RFID transactions came from the same tag by eavesdropping only, while the active tracking attack is that an adversary can actively participate in the transactions (like eavesdropping,)
interrupt, replay, and modification) to get the data to tell
whether two transactions came from the same tag. Both types
of tracking might be used to infer users’ location information
or even their personal profiles.

Due to the advances of hardware development, many
RFID schemes based on the public key techniques have been
proposed and implemented [3]. Compared with the other
cryptography mechanisms, the elliptic curve cryptography
(ECC) [4, 5] is more competitive since it could provide the
same security level with much smaller key size. Lee et al.
[6] proposed an ECC-based RFID authentication scheme.
Bringer et al. [7] and Deursen and Radomirovic [8] found
that Lee et al.’s scheme is vulnerable to the tracking attack
and the replay attack. Liao and Hsiao [9] proposed an ECC-
based RFID authentication scheme integrated with an ID
verifier transfer protocol; nevertheless, Peeters and Hermans
[10] showed Liao and Hsiao’s scheme cannot resist the server
impersonation attack. Tan [11] proposed ECC-based RFID
three-factor authentication. Arshad and Nikooghadam [12]
found that Tan’s scheme is not resistant to the replay attack
and the denial-of-service attack.

In 2014, He et al. [13] proposed an elliptic curve
cryptography- (ECC-) based RFID authentication protocol
which aimed at protecting tag’s anonymity and unlinkability
and improving the computational complexity. Compared
with the previous authentication schemes, He et al.’s scheme
has better performance in terms of security, computational
cost, and storage requirement. Unfortunately, we find that
their scheme fails to achieve the privacy protection if an
adversary launches active tracking attacks. We will show the
weaknesses and propose an improved scheme. The rest of this
paper is organized as follows. Section 2 gives the preliminary
sketch of the elliptic curve cryptography and bilinear pairing.
Section 3 reviews He et al.’s protocol and shows its security
weakness. In Section 4, we propose our new scheme, which is
followed by security analysis and performance evaluation
in Section 5. Finally, conclusions are given in Section 6.

2. Preliminaries

We briefly introduce the elliptic curve cryptography and the
bilinear pairing.

2.1. Elliptic Curve Cryptography. Koblitz [4] and Miller [5]
introduced elliptic curves for cryptographic applications.
Since then, elliptic curve cryptography (ECC) has played an
important role in many cryptosystems. An elliptic curve \( E \)
defined over the equation \( y^2 = x^3 + ax + b \) over \( F(q) \),
where \( q \) is a large prime and \( F(q) \) is a finite field of order \( q \).
The main attraction of ECC is that ECC with 160-bit key can reach a
security level the same as that of 1024-bit RSA and thereby
significantly reduce the key size.
The security of He et al.’s protocol is based on the
complexity of the elliptic curve discrete logarithm problem
(ECDLP) [14].

Elliptic Curve Discrete Logarithm Problem (ECDLP). Given an
elliptic curve \( E \) over \( F(q) \) and two points \( P \) and \( Q \) on \( E \), the
elliptic curve discrete logarithm problem is to find an integer
\( x \in \mathbb{Z}_q^* \) such that \( xP = Q \).

2.2. The Bilinear Pairing. The bilinear pairing was initially
considered as a negative property on the design of elliptic
curve cryptosystems, because it reduces the discrete loga-

thesis problem on some elliptic curves (especially for super-
singular curves) to the discrete logarithm problem in a finite
field [15]. Such property diminishes the strength of super-
singular curves in practice [16]. However, followed by the
triptite key agreement protocol proposed by Joux [17] and
the identity-based encryption scheme proposed by Boneh
and Franklin [18], pairing becomes beneficial and favorable
to the design of cryptographic protocols or cryptosystems [19].

Let \( G_0 \) be an additive cyclic group (which is the elliptic
curve group \( E(F_q) \) here) and let \( G_1 \) be a multiplicative cyclic
group with the same prime order \( n \); that is, \( |G_1| = |G_2| = n \).
Bilinear pairing is defined by \( \hat{e} : G_1 \times G_1 \rightarrow G_2 \) which satisfies
the following properties:

1. Bilinear: for all \( P, Q \in G_1 \) and all \( u, v \in \mathbb{Z}_n^* \), we have
   \( \hat{e}(uP, vQ) = \hat{e}(uP, Q)^v = \hat{e}(P, uQ)^v = \hat{e}(P, Q)^{uv} \).
2. Nondegenerate: \( \hat{e}(P, P) \neq 1 \) for some \( P \in G_1 \).
3. Computable: given \( P, Q \in G_1 \), there is an efficient
   algorithm to compute \( \hat{e}(P, Q) \).

We find that He et al.’s protocol is vulnerable to active
tracking attack. We will utilize the bilinear pairing to facilitate
our active attacks in Section 4.

3. Weaknesses of He et al.’s Protocol

3.1. Review of He et al.’s Protocol. This section reviews He
et al.’s protocol [13]. The system consists of three kinds of
entities: readers, a back-end server, and a set of tags; but the
RFID reader is omitted from the protocol description since it
acts as an intermediate party that relays messages exchanged
between a tag and the server. It is assumed that the communi-

cation between the reader and back-end server is secure. The
proposed protocol comprises two phases: setup and authenti-
cation. Notations used in the protocol are defined as follows:

(i) \( n, q \): two large primes.
(ii) \( F(q) \): a finite field of order \( q \).
(iii) \( E \): an elliptic curve defined by the equation \( y^2 = x^3 + ax + b \) over \( F(q) \).
(iv) \( P \): a generator point for a group of order \( n \) over \( E \).
(v) \( x_i \): the private key of the server.
(vi) \( P_i \): the public key of the server \( P_i = x_iP \).
(vii) \( X_T \): the ID verifier of the tag.

Setup Phase. To set up the system, the back-end server
performs the following tasks:

(i) Define \( \text{params} = \{q, a, b, P, n\} \) as the elliptic curve
domain parameters.
Server \((x_s, P_s, X_T)\) & Tag\(_{X_T}(P, X_T)\) \\
\[ r_1 \in \mathbb{Z}_n^* \quad R_1 = r_1 P \quad m_1 = \{R_1\} \quad r_2 \in \mathbb{Z}_n^* \quad R_2 = r_2 P \quad m_2 = \{R_2, Auth_T\} \]

\[
TK_{s1} = x_s R_1 \\
TK_{s2} = x_s R_2 \\
X_T = (\text{Auth}_s \oplus TK_{s2}) - TK_{s1} \\
\text{Search } X_T \\
\text{Auth}_s = (X_T + 2TK_{s1}) \oplus (2TK_{s2}) \\
m_3 = \{\text{Auth}_s\} \\
\text{Check } \text{Auth}_s = (X_T + 2TK_{s1}) \oplus (2TK_{s2})
\]

**Figure 1:** The authentication phase of He et al.'s protocol.

(i) Choose a random number \(x_s \in \mathbb{Z}_n^*\) as the server’s private key, and compute \(P_s = x_s P\) as the server’s public key.

(ii) Choose a random point \(X_T\) on \(E\) denoted as a tag’s ID verifier.

(iii) Choose a random number \(s_1\) and computes \(R_1 = s_1 P\), and sends \(m_1 = \{R_1\}\) as a challenge to the tag.

(iv) (params, \(P_s, X_T\)) is stored at both the tag and the server’s database.

(v) The server also keeps \(x_s\) secret.

**Authentication Phase.** To achieve mutual authentication, the server \((S)\) and the tag \((\text{Tag}_{X_T})\) do the following steps. The authentication phase is illustrated in Figure 1.

**Step 1** \((S \rightarrow \text{Tag}_{X_T} : m_1 = \{R_1\})\). \(S\) randomly chooses \(r_1 \in \mathbb{Z}_n^*\), computes \(R_1 = r_1 P\), and sends \(m_1 = \{R_1\}\) as a challenge to the tag.

**Step 2** \((\text{Tag}_{X_T} \rightarrow S : m_2 = \{R_2, \text{Auth}_T\})\). \(\text{Tag}_{X_T}\) randomly chooses \(r_2 \in \mathbb{Z}_n^*\) and computes \(R_2 = r_2 P, TK_{T1} = r_2 P_s, TK_{T2} = r_2 R_1,\) and \(\text{Auth}_T = (X_T + TK_{T1}) \oplus TK_{T2}\). Then, \(\text{Tag}_{X_T}\) sends back \(m_2 = \{R_2, \text{Auth}_T\}\) to \(S\). \(S\) can compute \(TK_{s2} = r_2 R_2\), which equals \(TK_{T2}\). Then, it obtains \(X_T + TK_{T1} \oplus TK_{T2} \oplus TK_{s2} = (X_T + TK_{T1})\). Now \(S\) performs the following steps to verify whether the two transactions came from the same tag:

1. It computes \((X_T + TK_{T1}) - (X_T + TK_{T1}) = TK_{T1} - TK_{T1} = (r_2 - r_2)P_s = (r_2 - r_2)P\).
2. It checks whether the equation \(\bar{e}(R_2 - R_2, P_s) = \bar{e}((X_T + TK_{T1}) - (X_T + TK_{T1}), P)\) holds.

If the transactions came from the same tag, the above verification equation should hold, because \(\bar{e}(X_T + TK_{T1}) - (X_T + TK_{T1}), P) = \bar{e}(r_2 - r_2, P_s, P) = \bar{e}(r_2 - r_2, P, P)\). That is, He et al.’s protocol cannot resist the active tracking attack.

**3.2. The Weaknesses.** We find that He et al.’s protocol is vulnerable to active tracking attack. We utilize the bilinear pairing to check whether the two transactions came from the same tag or not. We demonstrate our active attack as follows, where \(\text{Adv}\) denotes the notion that the adversary impersonates the server to get the responses for tracking. First of all, \(\text{Adv}\) randomly chooses \(r_1 \in \mathbb{Z}_n^*\), computes \(R_1 = r_1 P\), and sends message \(m_1 = \{R_1\}\) to probe the tags it encounters. In the following, we assume \(\text{Adv}\) encounters the same tag \(\text{Tag}_{X_T}\).

Upon receiving the query, \(\text{Tag}_{X_T}\) randomly chooses \(r_2 \in \mathbb{Z}_n^*\) and computes \(R_2 = r_2 P, TK_{T1} = r_2 P_s, TK_{T2} = r_2 R_1,\) and \(\text{Auth}_T = (X_T + TK_{T1}) \oplus TK_{T2}\). Then, \(\text{Tag}_{X_T}\) sends back \(m_2 = \{R_2, \text{Auth}_T\}\) to \(\text{Adv}\). \(\text{Adv}\) can compute \(TK_{s2} = r_2 R_2\), which equals \(TK_{T2}\). Then, \(\text{Adv}\) obtains \(X_T + TK_{T1} \oplus TK_{T2} \oplus TK_{s2} = (X_T + TK_{T1})\).

When \(\text{Tag}_{X_T}\) is probed again, it randomly chooses \(\bar{r}_2 \in \mathbb{Z}_n^*\) and computes \(\bar{R}_2 = \bar{r}_2 P, TK_{T1} = \bar{r}_2 P_s, TK_{T2} = \bar{r}_2 R_1,\) and \(\bar{\text{Auth}}_T = (X_T + TK_{T1}) \oplus TK_{T2}\). Then, \(\text{Tag}_{X_T}\) responds with \(m_3 = \{\bar{R}_2, \bar{\text{Auth}}_T\}\). \(\text{Adv}\) computes \(TK_{s2} = \bar{r}_2 R_2\), which equals \(TK_{T2}\). Then, it obtains \(X_T + TK_{T1} \oplus TK_{T2} \oplus TK_{s2} = (X_T + TK_{T1})\). Now \(\text{Adv}\) performs the following steps to verify whether the two transactions came from the same tag:

(1) \(\text{Adv}\) computes \((X_T + TK_{T1}) - (X_T + TK_{T1}) = TK_{T1} - TK_{T1} = (r_2 - \bar{r}_2)P_s = (r_2 - \bar{r}_2)P\).

(2) \(\text{Adv}\) checks whether the equation \(\bar{e}(R_2 - \bar{R}_2, P_s) = \bar{e}((X_T + TK_{T1}) - (X_T + TK_{T1}), P)\) holds.

If the transactions came from the same tag, the above verification equation should hold, because \(\bar{e}(X_T + TK_{T1}) - (X_T + TK_{T1}), P) = \bar{e}(r_2 - \bar{r}_2, P_s, P) = \bar{e}(r_2 - \bar{r}_2, P, P)\). That is, He et al.’s protocol cannot resist the active tracking attack.

**4. The Proposed Scheme**

We propose a new ECC-based scheme, which owns excellent performance in terms of security, computational complexity,
and communicational cost. Our scheme can resist all security threats including active tracking attack. Regarding computational complexity, we reduce the number of elliptic curve scalar multiplications, which is the most computationally expensive operation in ECC cryptography. For embedded systems like RFID and wireless sensor network, the communication operations consume the highest amount of energy of all the operations; therefore, reducing the message length is critical for saving the energy of these devices. The proposed scheme consists of two phases: setup and authentication. Since the setup phase is the same as that in He et al’s protocol, it is omitted here. The authentication phase is described as follows.

**Authentication Phase.** To achieve mutual authentication, the server (S) and the tag (Tag_{X_T}) do the following steps. The authentication phase is illustrated in Figure 2.

**Step 1** (S \rightarrow Tag_{X_T} : m_1 = \{R_1\}). S randomly chooses r_1 \in \mathbb{Z}_n^*, computes R_1 = r_1P, and sends m_1 = \{R_1\} as a challenge to the tag.

**Step 2** (Tag_{X_T} \rightarrow S : m_2 = \{R_2, \text{Auth}_T\}). Tag_{X_T} randomly chooses r_2 \in \mathbb{Z}_n^*, computes R_2 = r_2P, TK_{T_1} = r_2R_1, TK_{T_2} = r_2R_2, and \text{Auth}_T = (X_T + TK_{T_2}) \oplus H(R_1 + TK_{T_1}). Then, Tag_{X_T} sends back m_2 = \{R_2, \text{Auth}_T\} to S.

**Step 3** (S \rightarrow Tag_{X_T} : m_3 = \{\text{Auth}_T\}). S computes TK_{T_1} = x_SR_1, TK_{T_2} = r_2R_1, and X_T = (\text{Auth}_T \oplus H(R_1 + TK_{T_1})) - TK_{T_2}. Then S checks the server’s database for X_T. If it is not found, the server S rejects the tag; otherwise, the tag Tag_{X_T} is authenticated and thereafter S computes \text{Auth}_S = (X_T + 2TK_{T_2}) \oplus 2H(R_1 + TK_{T_1}) and sends back m_3 = \{\text{Auth}_S\} to Tag_{X_T}.

**Step 4**. Upon receiving the server’s response, Tag_{X_T} checks if (X_T + 2TK_{T_2}) \oplus 2H(R_1 + TK_{T_1}) = \text{Auth}_S. If it succeeds, the server S is authenticated; otherwise, the tag stops the procedure.

![Figure 2: The authentication phase of the proposed protocol.](image)

5. **Security Analysis and Performance Evaluation**

5.1. Security Analysis. We analyze the security of the proposed scheme as follows.

**Mutual Authentication.** The authentication of the tag is dependent on tag’s ability to prove its knowledge of the secret X_T. In our scheme, the server receives the message m_2 = \{R_2, \text{Auth}_T\}, where R_2 = r_2P and \text{Auth}_T = (X_T + TK_{T_2}) \oplus H(R_1 + TK_{T_1}). The server will use its private key x_s to compute TK_{si} = x_SR_1 and TK_{si} = r_2R_2, and to extract X_T = \text{Auth}_T \oplus H(R_1 + TK_{T_1}) - TK_{T_2}. Then, the server checks whether X_T is stored in the database. Only the genuine tag that owns the secret X_T can generate valid Auth_T.

The authentication of the server is dependent on server’s ability to extract X_T and generate valid Auth_s. Only the genuine server that owns the secret x_s, can correctly extract X_T from Auth_T and then compute valid Auth_s = (X_T + 2TK_{T_2}) \oplus 2H(R_1 + TK_{T_1}). Without knowledge of the server’s secret key x_s, the adversary cannot obtain TK_{si} = x_SR_1. The tag checks the validity of Auth_s. If it is valid, then the server is authenticated.

**Anonymity.** In our scheme, m_1 = \{R_1\}, m_2 = \{R_2, \text{Auth}_T\}, and m_3 = \{\text{Auth}_S\} are transmitted, where the tag-identity-related messages are Auth_T = (X_T + TK_{T_2}) \oplus H(R_1 + TK_{T_1}) and Auth_s = (X_T + 2TK_{T_2}) \oplus 2H(R_1 + TK_{T_1}) which are random due to two random and fresh numbers r_1 and r_2 in each session. Therefore, the adversary can learn nothing about the identity of the tag from the transmission. The randomness and freshness of the two random numbers ensure the anonymity of the proposed scheme.

**Tracking Attack Resistance.** The essence of the active tracking resistance of the proposed scheme is that each calculation of Auth_T = (X_T + TK_{T_2}) \oplus H(R_1 + TK_{T_1}) involves the confusion value H(R_1 + TK_{T_1}), where the computation of TK_{T_1} needs either tag’s secret r_2 or the server’s private key x_s; therefore,
|                         | Arshad and Nikooghadam [12] | Liao and Hsiao [9] | He et al. [13] | Ours                  |
|-------------------------|-----------------------------|--------------------|----------------|-----------------------|
| The server's computational cost | $2T_{EM} + T_M + T_{INV} + 8T_H = 490.58T_{EA} + 8T_H$ | $5T_{EM} + 3T_{EA} = 1208T_{EA}$ | $5T_{EM} + 3T_{EA} = 1208T_{EA}$ | $4T_{EM} + 4T_{EA} + 2T_H = 968T_{EA} + 2T_H$ |
| The tag's computational cost | $2T_{EM} + T_M + 7T_H = 490.58T_{EA} + 7T_H$ | $5T_{EM} + 3T_{EA} = 1208T_{EA}$ | $5T_{EM} + 3T_{EA} = 1208T_{EA}$ | $4T_{EM} + 4T_{EA} + 2T_H = 968T_{EA} + 2T_H$ |
| Number of rounds/steps  | 3                           | 3                  | 3              | 3                     |
| Total length of transmitted message | $8|x| + 2L_{ECC}$                     | $4|x| + L_{ECC}$                  | $4|x| + L_{ECC}$                  | $4|x| + L_{ECC}$                  |
| The tag's transmission length | $4|x| + L_{ECC}$                          | $2|x| + L_{ECC}$                 | $2|x| + L_{ECC}$                 | $2|x| + L_{ECC}$                 |
| The server's storage cost  | $(n + 1)|x| + nL_{ECC}$                  | $(n + 1)|x| + nL_{ECC}$                  | $(n + 1)|x| + (n + 1)L_{ECC}$          | $(n + 1)|x| + (n + 1)L_{ECC}$          |
| The tag's storage cost    | $5|x| + L_{ECC}$                        | $2L_{ECC}$                        | $2L_{ECC}$                        | $2L_{ECC}$                        |
| Security weaknesses       | Server impersonation           | Active tracking             | No              | No                    |

Table 1: Performance comparison.
an active tracker has no way to derive any verifiable data from the transmissions. We can verify this by launching the same active attack on our proposed protocol as follows, where Adv denotes the notion that the adversary impersonates the server to get the responses for tracking.

First of all, Adv randomly chooses \( r_1 \in \mathbb{Z}_p^* \), computes \( R_1 = r_1 P \), and sends message \( m_1 = \{R_1\} \) to probe the tags it encounters. In the following, we assume Adv encounters the same tag \( \text{Tag}_{X_1} \).

Upon receiving the query, \( \text{Tag}_{X_1} \) randomly chooses \( r_2 \in \mathbb{Z}_p^* \) and computes \( R_2 = r_2 P, \ TK_{T1} = r_2 P_T, \ TK_{T2} = r_2 R_1, \) and \( \text{Auth}_T = (X_T + TK_{T2}) \oplus H(R_1 + TK_{T1}) \). Then, \( \text{Tag}_{X_1} \) sends back \( m_2 = \{R_2, \text{Auth}_T\} \) to S. Since Adv cannot compute \( TK_{T1} = r_2 P_T \), Adv obtains nothing except \( \text{Auth}_T \). When \( \text{Tag}_{X_1} \) is probed again, it randomly chooses \( r_2 \in \mathbb{Z}_p^* \) and computes \( R_2 = r_2 P, \ TK_{T1} = r_2 P_T, \ TK_{T2} = r_2 R_1, \) and \( \text{Auth}_T = (X_T + TK_{T2}) \oplus H(R_1 + TK_{T1}) \). Then, \( \text{Tag}_{X_1} \) responds with \( m_2 = \{R_2, \text{Auth}_T\} \). Adv cannot compute \( TK_{T1} = r_2 P_T \), and Adv obtains nothing except \( \text{Auth}_T \). Adv cannot verify whether the two transactions came from the same tag. That is, our proposed protocol can resist the active tracking attack.

**Tag Masquerade Attack Resistance.** To impersonate a tag, the adversary must be able to generate a valid message \( m_1 = \{\text{Auth}_T\} \), where \( \text{Auth}_T = (X_T + TK_{T2}) \oplus H(R_1 + TK_{T1}) \). However, it is difficult to generate such a message without knowing the identity of the tag \( X_T \).

**Server Spoofing Attack Resistance.** To impersonate the server, the adversary must be able to generate a valid message \( m_1 = \{\text{Auth}_T\} \), where \( \text{Auth}_T = (X_T + TK_{T2}) \oplus H(R_1 + TK_{T1}) \). It is easy for the adversary to generate \( R_1 \), but it is difficult to generate \( \text{Auth}_T \) without knowledge of the server’s secret key \( x_T \) and the tag’s identity \( X_T \).

### 5.2. Performance Evaluation

We compare the proposed scheme with He et al.’s protocol [13] and some related schemes [9, 12] in terms of computational cost, communication cost, and storage cost. Let \( T_{EA} \) denote the cost of point addition over an elliptic curve \( E \), let \( T_{EM} \) denote the cost of scalar multiplication over an elliptic curve \( E \), let \( T_M \) denote the cost of modular multiplication over the underlying field \( F(q) \), let \( L_{ECC} \) denote the bit length of one elliptic curve point, let \( |x| \) denote the size of integer \( x \), and let \( n \) denote the number of tags in the system. To evaluate the complexity, we adopt the practical figures from [20]. In [20], it lists the timing for computing \( kP \) and \( q^x \) mod \( p \), where \( E \) is an elliptic curve defined over \( F(q) \), \( q = 2^{160} \), \( P \) is a point whose order is 160-bit prime over \( E \), \( k \) is a random 160-bit integer, and \( p \) is a 1024-bit prime. Therefore, we can conclude that \( T_M \approx (41/5)T_{EA} \approx 8T_{EA} \), \( T_{EM} \approx (29/0.12)T_{EA} \approx 241T_{EA} \), and \( T_{IN} \approx (3 \times 8/41)T_{EA} \approx 0.58T_{EA} \) [20].

Note that the cost of executing an exclusive-or operation (XOR) is negligible when compared with other operations stated above. Since the parameters \( q, a, b, P, n \) are stored in both the server and the tag, the storage cost of parameters is omitted in the following comparison.

The performance comparison is summarized in Table 1. Since He et al.’s protocol [13], Liao and Hsiao’s scheme [9], Arshad and Nikooghadam’s scheme [12], and our proposed scheme rely on the ECDLP, the elliptic curve scalar multiplication is the most time-consuming operation in the elliptic curve cryptosystem. Although our proposed scheme has the same computational and storage costs as He et al.’s protocol, our proposed scheme owns better computational performance by eliminating one elliptic curve scalar multiplication operation. Our proposed scheme is more efficient than Liao and Hsiao’s scheme because our proposed scheme requires less cost in terms of computation, communication, and storage. Table 1 shows that Arshad and Nikooghadam’s scheme requires less computational cost than our proposed scheme. However, it has been studied that communication consumes more energy than computation in embedded wireless communication systems like RFID and wireless sensor network [21, 22].

Studies in the past have shown that 3000 instructions could be executed for the same energy usage as sending a bit 100 m by radio [23]; therefore, many studies in these fields devoted lots of efforts to reducing the communication complexity [24, 25]. It is important to optimize communication and minimize energy consumption. In our proposed scheme, the tag communication requires only 50% of that of Arshad and Nikooghadam’s scheme, while our scheme achieves the same security properties with slightly more computations.

### 6. Conclusions

Mobile healthcare systems are becoming more and more popular. Lack of protecting patient and data privacy may hinder the utility of mobile healthcare system. In this paper, we have shown the weakness of He et al.’s protocol. The protocol cannot meet privacy protection requirement since it is vulnerable to active tracking attack. We have proposed a new scheme which not only conquers the security weaknesses but also improves the computational performance.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgment

This project is partially supported by the National Science Council, Taiwan, under Grant no. MOST 103-2221-E-260-022.

### References

[1] A.-R. Sadeghi, I. Visconti, and C. Wachsmann, “User privacy in transport systems based on RFID e-tickets,” in Proceedings of the 1st International Workshop on Privacy in Location-Based Applications, pp. 102–122, Malaga, Spain, October 2008.

[2] Y.-C. Yen, N.-W. Lo, and T.-C. Wu, “Two RFID-based solutions for secure inpatient medication administration,” Journal of Medical Systems, vol. 36, no. 5, pp. 2769–2778, 2012.
[3] Y. Chen, J.-S. Chou, and H.-M. Sun, “A novel mutual authentication scheme based on quadratic residues for RFID systems,” Computer Networks, vol. 52, no. 12, pp. 2373–2380, 2008.

[4] N. Kobitz, “Elliptic curve cryptosystems,” Mathematics of Computation, vol. 48, no. 177, pp. 203–209, 1987.

[5] V. Miller, “Use of elliptic curves in cryptography,” in Advances in Cryptology—CRYPTO ’85 Proceedings, vol. 218 of Lecture Notes in Computer Science, pp. 417–426, Springer, Berlin, Germany, 1985.

[6] Y. Lee, L. Batina, and I. Verbauwhede, “EC-RAC (ECDLP based randomized access control): provably secure RFID authentication protocol,” in IEEE International Conference on RFID, pp. 97–104, Las Vegas, New, USA, April 2008.

[7] J. Bringer, H. Chabanne, and T. Icart, “Crytpanalysis of EC-RAC, a RFID identification protocol,” in Cryptology and Network Security: 7th International Conference, CANS 2008, Hong Kong, China, December 2–4, 2008. Proceedings, vol. 5339 of Lecture Notes in Computer Science, pp. 149–161, Springer, Berlin, Germany, 2008.

[8] T. Deursen and S. Radomirovic, “Attacks on RFID protocols (version 1.1),” Tech. Rep., University of Luxembourg, 2009.

[9] Y.-P. Liao and C.-M. Hsiao, “A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol,” Ad Hoc Networks, vol. 18, pp. 133–146, 2014.

[10] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” in Advances in Cryptology—ASIACRYPT 2001, vol. 2248 of Lecture Notes in Computer Science, pp. 514–532, Springer, Berlin, Germany, 2001.

[20] A. Joux, “A one round protocol for tripartite Diffie-Hellman,” in Proceedings of the 4th Algorithmic Number Theory Symposium (ANTS ’00), pp. 385–394, Leiden, The Netherlands, July 2000.

[21] F. Zhao and L. J. Guibas, Wireless Sensor Networks: An Information Processing Approach, Elsevier-Morgan Kaufmann, San Francisco, Calif, USA, 2004.
