RFID Authentication with Un-Traceability and Forward Secrecy in the Partial-Distributed-Server Model

Hung-Yu CHIEN†, Nonmember, Tzong-Chen WU††, Member, and Chien-Lung HSU†††, Nonmember

SUMMARY Secure authentication of low cost Radio Frequency Identification (RFID) tag with limited resources is a big challenge, especially when we simultaneously consider anonymity, un-traceability, and forward secrecy. The popularity of Internet of Things (IoT) further amplifies this challenge, as we should authenticate these mobile tags in the partially distributed-server environments. In this paper, we propose an RFID authentication scheme in the partially distributed-server environments. The proposed scheme owns excellent performance in terms of computational complexity and scalability as well as security properties.

key words: authentication, RFID, anonymity, distributed servers, error correction code

1. Introduction

The popularity of Radio Frequency Identification (RFID) tags arouses people’s concerns about its security which includes authentication and resistance to various attacks, and the potential infringement on the privacy of their carriers. To protect the privacy of the tagged objects and their carriers, anonymity, un-traceability and forward secrecy should be taken in account. Anonymity protects a tag’s identity from un-authorized access while un-traceability ensures that attackers have no way to link the communications from the same tag. The property of anonymity is covered by the achievement of the un-traceability property. A scheme that does not well protect its identity privacy from un-authorized access cannot provide un-traceability since attackers who derive the identities can trace the tags. On the contrary, a scheme that protects tag’s anonymity might be vulnerable to tracing attack, because an attacker might link two sessions to the same tag using the eavesdropped data. Forward secrecy further ensures that an attacker could not trace one tag’s past communications, even if we assume that the tag might be compromised in the future. To authenticate RFID tags with the security properties above is a big challenge, especially for those low-cost cryptographic tags like Mifare series [1]. In the rest of this paper, we assume low-cost cryptographic tags as our target; however, our scheme can be applied on other mobile devices.

Even though servers are much more computationally powerful in general and can support most cryptographic algorithms, we still should be careful in designing protocols to avoid unnecessarily heavy computations. Unfortunately, in some schemes like [2]–[5], the server must perform computations and matching operations for all tags in the database to identify an anonymous tag. It is very inefficient. Given the possible exponential growth of the number of tags, this poor performance could be a big burden.

Conventionally, design of RFID authentication protocols concentrates on the single-server model where one central backend server run by a single organization owns all the data to authenticate tags (servers which are owned by the same authority, share and synchronize the same data can be regarded as a single logical server). This single-server model fitted many RFID applications scenarios before, but it gradually falls far behind satisfying as Internet of Things (IoT) are becoming more and more popular now.

In IoT, a tag might be authenticated by several independent organizations (and their corresponding servers) during its lifetime, and the moving patterns of a tag might not be fully predictable. It means that any authorized server needs to authenticate these tags at any possible time without any pre-notification from other servers and not a single server can control the roaming patterns of tags. Figure 1 depicts the potential scenario of mobile tags roaming among independent servers; for example, Mifare card has been used as an access tokens for some transport system, as a cash card in stores, and as an access token for museum or exhibitions; Similarly, Suica card [7] in Japan serves as an access token and (a cash card) for many applications; however, anonymity and un-traceability are not considered in these applications now. This infringes user’s privacy and limits the potential of IoT. We call this scenario the distributed-server model which could be further divided into two cases: full-distributed-server model and partial-distributed-server model. The full-distributed-server model requires that all the servers are completely independent while the partial-distributed-server model calls for independent servers after initialization. The full-distributed-server model might be implemented using public-key-infrastructure (PKI), which is impractical for the current low-cost RFID scenarios. We, therefore, focus on the partial-distributed-server model in this paper.

In the partial-distributed-server model, the servers might co-operate and share some data of the tags in some
ways, but it is inconvenient or unreasonable for these servers to synchronize the data and keys during the lifetime of tags, or to notify other servers which tags will be authenticated in the IoT scenarios. This makes the partial-distributed-server model different from a logical server consisting of several servers. We notice that it is not trivial to extend an extant scheme in the single-server model to the partial-distributed-server model while keeping all the desirable security properties. We might naively duplicate and synchronize the database of one server to multiple servers without re-designing these protocols; however, this approach would compromise either forward secrecy property or feasibility of authentication between tag and server of existing schemes (details are listed in Table 3).

RFID authentication has been intensively studied by researchers like [2]–[6], [11]–[26]. The security features of extant schemes have been improved to some extent, but only few of them simultaneously considered anonymity, un-traceability and forward secrecy. More importantly, all of the previous studies considered only the single-server model.

We, therefore, aim at proposing anonymous RFID authentication with un-traceability and forward secrecy in this partial-distributed-server model. Based on Rabin cryptosystem [8] and Error Correction Codes (ECC) [9], [10], this paper will propose a new RFID authentication protocol that satisfies the above-mentioned properties. The tags in our scheme only perform Pseudo Random Number Generator (PRNG) functions, squaring modulo and simple operations like XOR. The Combination of Rabin cryptosystem with ECC is one of the potential solutions. One another potential solution is those Rabin-based schemes with chosen-ciphertext-attack security like Hofheinz et al.’s scheme [39]. Compared to Hofheinz et al.’s scheme, one advantage of our scheme is that its computational complexity of tag is much lighter than two modular exponentiations in Hofheinz et al.’s scheme.

The merits of this paper include:

1. we highlight the security concerns in the partial-distributed-server model, which was previously neglected;

2. we propose a new RFID authentication scheme with anonymity, un-traceability and forward secrecy in the partial-distributed-server model;

3. the computation on tag is affordable on those low-cost cryptographic tags like Mifare series, and its excellent performance makes it attractive to IoT applications.

The remainder of this paper is organized as follows. Section 2 gives some preliminaries on error correction codes and Rabin cryptosystem. Section 3 proposes our scheme. The security analysis is given in Sect. 4, which is followed by the performance evaluation in Sect. 5. Finally, Sect. 6 states the conclusions and future work. In the rest of this paper, we interchange the terms organization and server, depending on which one fits the context better.

2. Preliminaries

2.1 Linear Block Codes

Linear block codes are commonly used to recover noises during transmission [9], [10]. A linear error correction code of length n, dimension k, and minimum distance d over \( GF(2) \) is denoted by \( C(n, k, d) \), and the codes can be defined by its \( k \)-by-\( n \) generator matrix \( G_{k \times n} \) over \( GF(2) \). To transmit a message vector \( m = (m_1, \ldots, m_k) \), a sender encodes \( m \) into \( c = m \cdot G = (c_1, \ldots, c_n) \). Assume there are noises during the transmission, and we denote it as \( \tilde{c} = c + e \), where \( e \) is the noise with \( |e| \leq t \) and \( t = (d - 1)/2 \). Then, upon receiving the transmission \( \tilde{c} \), a designated receiver who owns the parity matrix \( P \) (corresponding to the generator matrix \( G \) ) can identify the noise \( e \) and codeword \( c \), and recovers the message \( m \). In our proposed scheme, the sender will add artificial noises such that only the designated receiver who owns the secret parity matrix \( P \) can recover and verify the message \( m \).

2.2 Rabin Cryptosystem

The Rabin cryptosystem [8] is an asymmetric cryptographic technique, whose security has been proved to be as hard as integer factorization, and was proved in-distinguishability against chosen plaintext attack (IND-CPA) secure. However, it has the property that each output of the Rabin function can be generated by any of four possible inputs; for each output as a ciphertext, extra complexity is required on decryption to identify which of the four possible inputs is the true plaintext. The scheme consists of three algorithms - key generation, encryption and decryption as follows.

**Key Generation:** Choose two large distinct primes \( p \) and \( q \) with \( p = q = 3 \mod 4 \) to simplify the computation of square roots modulo \( p \) and \( q \). Let \( N = p \cdot q \). Then \( N \) is the public key, and \((p, q)\) is the private key.

**Encryption:** Let \( m \in [0, \ldots, N - 1] \) be the plaintext. To encrypt \( m \), the sender computes the ciphertext \( C = m^2 \mod N \).
Table 1 Notations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_i, IDT_i$</td>
<td>Tag $T_i$ with identity $IDT_i$</td>
</tr>
<tr>
<td>$R_i, IDR_i$</td>
<td>Reader $(R_i)$ with identity $IDR_i$</td>
</tr>
<tr>
<td>$S$</td>
<td>The backend server</td>
</tr>
<tr>
<td>$g()$</td>
<td>Pseudo-random function</td>
</tr>
<tr>
<td>$p, q, N$</td>
<td>$p$ and $q$ are two large primes, and $N = p \cdot q$</td>
</tr>
<tr>
<td>$\oplus$</td>
<td>Exclusive OR, string concatenation</td>
</tr>
<tr>
<td>$C(n, k, d)$</td>
<td>A linear error correction code of length $n$, dimension $k$, and minimum distance $d$</td>
</tr>
<tr>
<td>$G_{k,n}$</td>
<td>$k$-by-$n$ generator matrix $G_{k,n}$</td>
</tr>
<tr>
<td>$e$</td>
<td>The noise with $</td>
</tr>
<tr>
<td>$c_i$</td>
<td>A codeword belongs to $C(n, k, d)$</td>
</tr>
<tr>
<td>$K_i$</td>
<td>The assigned secret key to the tag $T_i$</td>
</tr>
<tr>
<td>$N_R$</td>
<td>Random challenge from reader (or the server)</td>
</tr>
<tr>
<td>$V_{TR}, V_{ST}$</td>
<td>Validation value from the tag; validation from the server to the tag</td>
</tr>
</tbody>
</table>

Decryption: To decrypt the ciphertext $C$, the receiver who knows the private keys $p$ and $q$ can apply the Chinese remainder theory to derive the four answers $\{m_1, m_2, m_3, m_4\}$. To uniquely identify the plaintext, one common technique is to add some pre-defined padding in the plaintext or requires the plaintext conform to some pre-defined format.

3. The Proposed Scheme in the Partial-Distributed-Server Model

Based on Rabin cryptosystem and ECC, this section proposes a new RFID authentication scheme with anonymity, un-traceability and forward secrecy in the partial-distributed-server model, and it meets the resource limitation of low-cost cryptographic tags like Mifare series tags.

3.1 Assumptions

We assume that tags can only support simple functions like PRNG, squaring modulo, and simple bitwise operations (like addition, XOR, AND, and OR). The system consists of several partially independent distributed servers, readers, and tags. Here independent servers mean that they might share public parameters and share some tags’ initial data; but they operate independently and would not synchronize their data after initialization. Generally, it is reasonable to assume that the channel between a server and its corresponding readers is secure, since readers are resource-abundant to implement those conventional and well-studied cryptographic algorithms; we, therefore, use the notation "reader (or R)" to denote that the reader on behalf of the server performs the authentication functions.

3.2 The Scheme

The main idea of our scheme is that the sender randomly adds a noise $e$ with $|e| = t$ to its pre-assigned codeword $c$ to have $m = c + e$, and then applies the encryption of Rabin scheme to have $C = m^2 \mod N$, where $N = p \cdot q$. Upon receiving $C$, the designated receiver who knows the secrets $p$ and $q$, can apply the Chinese remainder theory to get the four possible answers $\{m_1, m_2, m_3, m_4\}$, uses the secret parity matrix (corresponding to $G_{k,n}$) to identify the noise $e$ and to derive the corresponding codeword $c$ for each $m_i$, and then verifies which one satisfies the verification equation. However, this simple scheme does not own many security properties we need; so the proposed scheme is introduced as follows.

The scheme involves three kinds of entities; servers (denoted as $\{S_i\}$), readers (denoted as $\{R_i\}$), and tags (denoted as $\{T_i\}$). The scheme consists of two phases - initialization and authentication.

Initialization Phase: Initially, the system chooses a public PRNG function $g()$, two large primes $p$ and $q$, and a randomly chosen secret linear code $C(n, k, d)$ over $GF(2)$, which is specified by its generator matrix $G_{k,n}$. The system publicly publishes $g()$ and $N = p \cdot q$, but secretly shares the values $(p, q)$ and the linear codes specified by $G_{k,n}$ among the set of servers $\{S_i\}$. For each registered tag $T_i$, $1 \leq i \leq 2^2 - 1$, the system randomly chooses a key $K_i$ and an un-used non-zero codeword $c_i \in C(n, k, d)$ to tag $T_i$. The system assigns tag $T_i$ the attributes $(IDT_i, c_i, K_i, ind_{T_i} \leftarrow 1)$, where $IDT_i$ is the tag’s identity, $c_i$ is the assigned codeword, $K_i$ is the assigned secret key (which will be updated per successful authentication), and $ind_{T_i}$ is the index indicating the number of successful authentications for this tag (also the number of key updating). The servers $\{S_i\}$ keep $(IDT_i, c_i, K_i, pre, ind_{T_i,pre})$ for each tag $T_i$, where $K_i, pre \leftarrow K_i$ and $ind_{T_i,pre} \leftarrow ind_{T_i} = 1$ initially. The values $K_i, pre$, $ind_{T_i,pre}$, and $ind_{T_i}$ will be updated separately by $T_i$ and each individual server $S_i$ in later authentication sessions.

The authentication phase: The authentication phase of the protocol is described as follows (and in Fig. 2).

1. $R(S) \rightarrow T_i : N_R$
   The reader sends its Query message with a random challenge $N_R$ to tag $T_i$.

2. $T_i \rightarrow R(S) : M_i$
   2.a. $T_i$ generates a random error vector $e_i$ with Hamming weight $t$, and computes $\tilde{c}_i = c_i + e_i$ and $V_T = g(e_i \oplus g(N_R \oplus K_i))$, where $c_i$ is the codeword assigned to $T_i$ and $K_i$ is the assigned secret key. Here we abuse the notation $g(e_i \oplus g(N_R \oplus K_i))$, even if the length of $e_i$ is different from the output of $g()$, and a necessary string expansion or shrinking is applied when the lengths of two operands are different. $T_i$ further lets $m_i = \tilde{c}_i[\|ind_{T_i}\|V_T]$ and computes $M_i = m_i^2 \mod N$.
   2.b. $T_i$ sends $M_i$ to $R(S)$.

3. $R(S) \rightarrow T_i : V_{ST}$
   3.a. The server $S$ who knows $p$ and $q$ first applies the Chinese remainder theory to derive four answer $\{m_{1u}, m_{2u}, m_{3u}, m_{4u}\}$ from $M_i$, where $m_{iu} = \tilde{c}_{iu}[\|ind_{T_i}\|V_{Tu}$ for $u = 1, 2, 3, 4$ and one of them should be the right answer for $m_i$, if $V_{Tu}$ is valid. For each $\tilde{c}_{iu}$, the server uses the par-
Fig. 2 Anonymous RFID authentication with forward secrecy and un-traceability in the partial-distributed-servers model.

3.a. For $M_l$, decode 4 possible $\{c_l, ind_{T_l}, V_T\}$;
For each $c_l$, decode $(c_l, e_l)$ and identify the tag $T_l$ & $K_{l,pre}$ & $ind_{T_l,pre}$
check whether $ind_{T_l,pre} \leq ind_{T_l}; K_l \leftarrow g^{ind_{T_l,pre}}(K_{l,pre})$
check $V_T = g(e_l \oplus g(N_R \oplus K_l))$;
If succeeds, $V_{ST} = g(N_R \oplus g(e_l \oplus K_l))$,
update $K_{l,pre} \leftarrow K_{l,pre} \leftarrow ind_{T_l,pre}$

3.b. $V_{ST}$

4. verify $V_{ST} = g(N_R \oplus g(e_l \oplus K_l))$
update $K_l \leftarrow g(K_l), ind_{T_l} \leftarrow ind_{T_l} + 1$

Fig. 3 One snapshot of key chains of a tag and those distributed servers.

one is identified, the server identifies the corresponding tag $T_l$, its key $K_l$ and the current index $ind_{T_l,pre}$. $S$ then prepares the response $V_{ST} = g(N_R \oplus g(e_l \oplus K_l))$. It updates $K_{l,pre} \leftarrow K_l$ and $ind_{T_l,pre} \leftarrow ind_{T_l,pre}$.

3.b. The server $S$ sends the identity of the tag $V_{ST}$ and to $R$, which forwards $V_{ST}$ to $T_l$.

4. $T_l$;
$T_l$ checks whether the verification equation $V_{ST} = g(N_R \oplus g(e_l \oplus K_l))$ holds to authenticate the reader ($R$) and the server ($S$). If OK, it accepts the server and the reader, and updates $K_l = g(K_l)$ and $ind_{T_l} = ind_{T_l} + 1$.

Note that the rational of combining Rabin cryptosystem and error correction codes is to ensure anonymity and
un-traceability. If we send ID and challenges separately and encrypt the ID only, then the cipher-text of the encrypted ID will be the same for the same tag, and the attackers can trace this tag. There existed some publications like [22],[30] that apply Rabin cryptosystem to encrypt ID and challenges, but some security weaknesses of them had been reported [26].

Previously proposed schemes do not satisfy all the security properties we need in the partial-distributed-server model.

The authentication is based on the challenge-and-response technique, applying on the secret code-word, the challenges, and the loosely-synchronized key chain among tags and partial-distributed servers. Figure 3 shows one possible snapshot of the key chains of a tag and those partial-distributed servers, where one server’s index is one step behind that of the tag while other servers’ indexes are behind that of the tag in distinct steps, depending on how long they did not authenticate the tag. The ECC decoding and the Rabin decryption of $M_f$ are performed only by the server. The tags perform only pseudo random function, additions, XOR, and squaring modulo.

4. Security Analysis and Privacy Property

The security of the proposed scheme is based on three mechanisms- Rabin cryptosystem, the secret linear codes decoding and the challenge-response technique using secret loosely-synchronized key chains. The ECC decoding part can be viewed as a secret-key-based authentication scheme, and the output of ECC encoding is further encrypted by the Rabin encryption. Rabin cryptosystem has been proved to be as secure as the integer factoring problem. Therefore, those chosen cipher text attacks [27] on the McEliece-like public key cryptosystems (which are ECC-based public key schemes), and the chosen plaintext attacks [28] on the ECC-based symmetric encryption schemes could not be directly applied on the proposed scheme. However, those message-resend attacks and related-message attacks [29] on the McEliece-like public key cryptosystems and on Chien-Laih’s ECC-based authentication [11] should be examined, as the same tag would be probed many times by either genuine readers or attackers. The security properties are examined as follows.

Theorem 1: The proposed scheme achieves mutual authentication between genuine tags and readers (and the servers) respectively.

Proof: The security of the mutual authentication is based on the challenge-response applied on challenges ($N_R, e_i$) and the secret key $K_f$. In order to be authenticated, one tag should generate valid response $V_T = g(e_i \oplus g(N_R \oplus K_f))$ using the secret value $K_f$ and the current challenge $N_R$. On the other side, the server should correctly decode the transmission $M_t$ to identify the tag, its current error vector $e_i$ and retrieve the key $K_f$ to generate the valid response $V_{ST} = g(N_R \oplus g(e_i \oplus K_f))$; It requires that the server should know the secret parameters $p$ and $q$, the secret linear codes $G_{sec}$ and the tag’s secret key $K_f$. So only the genuine tag and the genuine server who owns the secrets could generate valid responses and authenticate each other.

Theorem 2: The proposed scheme can resist replay attack, modification attack and other possible impersonation attacks.

Proof: (1) Resistance to replay attack. In each session, the server will issue a random fresh challenge $N_R$ and the tag will choose a fresh random error vector $e_i$ as challenge, and expect the communicating party can reply with correct responses $V_T$ and $V_{ST}$, respectively. Therefore, a replay data which is based on obsolete challenges ($N_R, e_i$) could cheat neither the server nor the tag. (2) Resistance to modification attack and other possible impersonation attacks. For any impersonation attacks (including modification attack) to be successful, attackers should generate correct responses $V_T$ or $V_{ST}$, of which the computation is based on the current challenges ($N_R, e_i$) and the current secret key $K_f$. Attackers who have no knowledge of $K_f$ have no way to generate valid responses. This proves the theorem.

Theorem 3: The proposed scheme can resist de-synchronization attack (or called Denial of Service attack).

Proof: The authentication of the proposed scheme depends on tag’s and servers’ synchronization of the key chain. An attacker might try to modify the transmissions to desynchronize the key chain shared between a tag and the servers. In our scheme, a server keeps the key $K_{f,pre}$, which is the matched key it had in the last successful authentication and this key might lag behind the key the tag keeps now. However, a server learns the tag’s current key chain index $ind_{f,t}$ by deciphering $M_t$; after that, it can synchronize its key with the tag by computing $K_f = g^{ind_{f,t}-ind_{f,pre}}$. This mechanism ensures that each genuine server could synchronize its key with the tag, and ensures authentication between the tag and the servers. This proves this theorem.

Theorem 4: The proposed scheme achieves forward secrecy.

Proof: Low-cost tags are simple devices and might be compromised by attackers who capture a tag. Forward secrecy requires that, even if we assume a tag might be compromised in the future, an attacker who compromises the tag still could not trace the past communications from the same tag. In our scheme, attackers who compromises a tag could derive the identity, the index and the current secret key of the compromised tag; however, to verify whether one previous transmission transcript ($M_t, V_T, V_{ST}$) came from the same tag, it is required that attackers should derive the corresponding noise $e_i$ and the corresponding key $K_f$ used in that session; fortunately, it is infeasible to derive the previous secret key from the compromised current key, because the key chain is securely updated as $K_f = g(K_f)$; and, it is infeasible to derive the random noise without having the secret generator matrix. This proves the theorem.
Theorem 5: The proposed scheme can resist message-resend attack and related-message attack, and, therefore, protects anonymity and un-traceability.

Proof: In some ECC-based schemes like [11], [29], attackers may try impersonating readers and repeatedly probing the same tag several times to trace the tag. If the distance of the two responses $M_t$s from two distinct sessions is less than $2t$, where $t = \lfloor (d-1)/2 \rfloor$, then the attackers infer that the two responses may come from the same tag, since the tag adds a random noise with weight equaling $t$ each time. However, this attack is not applicable on our scheme, because the code-related value $m_t = \tilde{c}_t \parallel \text{ind}_{T_t} \parallel V_t$, where $\tilde{c}_t = c_t + e_t$, is encrypted (squaring modulo) as $M_t = m_t^2 \mod N$ using Rabin cryptosystem. The randomness of added artificial error vector is $\log_2(\frac{\pi M_l}{\sqrt{\text{sti}}}c_l)$. By choosing proper parameters (as two examples in Sect. 5.2), the added error vectors and inclusion of the secret value $\text{ind}_{T_t}$ provides enough randomness as secret input to the PRNG function. The squaring operation further makes the difference between two outputs $M_t$s randomly distributes, even if they come from the same tag. This proves the un-traceability. The property of anonymity is covered by the achievement of the un-traceability property. A scheme that does not well protect its identity privacy from un-authorized access cannot provide un-traceability since attackers who derive the identities can trace the tags. In our scheme, the only identity-related value is the code word $c_t$. A tag concatenates its code word with the index value $\text{ind}_{T_t}$ and the random value $V_t$, and then encrypts the concatenated value using Rabin encryption. This ensures the anonymity and un-traceability of our scheme.

5. Performance Analysis

We evaluate the performance of the proposed scheme and compare its performance with its counterparts. A general comparison of security properties, partial-distributed-server model support and scalability is discussed in Sect. 5.1. We then further examine the performance of several forward-secrecy schemes in the partial-distributed-server model in Sect. 5.2. Finally, Sect. 5.3 focuses on comparing the computational performance of several ECC-based schemes.

5.1 A General Comparison of Security Properties, Partial-Distributed-Server Model Support and Scalability

In order to protect tag’s identity privacy, we note that some previous schemes like [2]–[5], [13], [16], [30] did not transmit tag-identity-related data in plaintext but they required the server performing some computations for all tags in the database to identify an anonymous tag; the cost of this process is very costly when the number of tags is large. We note this property as exhaustive computation and matching in Table 2. On the contrary, to identify an anonymous tag, the servers in our scheme only perform one Rabin-cryptosystem decryption (which involves two square root computations (exponential modulo computations) and one Chinese remainder theorem computation), 2 ECC-decoding operations on average, and 5 pseudo random function computations. Compared to those schemes [2]–[5], [13], [16], [30] which require exhaustive computation and matching, our scheme is much more efficient, when the number of tags is large.

In our scheme, it is noted that only the servers are equipped with the decoding algorithms of ECC-decoding and the Chinese remainder theorem function. One ECC decoding operation involves one matrix multiplication: which is quite efficient for server; or one can implement ECC decoding by syndrome decoding, which is minimum distance decoding using a reduced lookup table (the time complexity is $O(k^2(n-k))$ and the space complexity is $p^{n-k}$ for a code $C(n,k,d)$ over $F_2$ (it is $2^{n-k}$ for $F_2$ in our scheme) [10]. The large space requirement on server is a burden; luckily, as technologies advance, storage of that size is feasible. The functions required on tag are only squaring modulo, PRNG, and simple bit-wise operations like XOR. The cost of computations on tags is low.

Since all the previous schemes were designed for the single-server model, there is no obvious way to compare our scheme with them; however, the single-server model is just a special case of the partial-distributed-server model; we, therefore, give a general comparison of our scheme applied in the single-server model with some previous schemes which aimed at providing anonymity, un-traceability and forward secrecy in Table 2. In short summary, our scheme is the only one that simultaneously provides the merits: (1) it supports the partial-distributed-server model; (2) it resists all possible attacks; (3) it achieves forward secrecy; and (4) no exhaustive computation and matching is required to identify an anonymous tag.

5.2 Discussion of Forward-Secrecy Schemes in the Partial-Distributed-Server Model

To further examine the performance of some previous schemes in the partial-distributed-server model, our comparison focuses on those schemes that provided forward secrecy, because forward-secrecy-preserving protocols are usually stateful protocols and it is challenging to preserve forward secrecy, state synchronization and the feasibility of authentication in the partial-distributed-server model. One possible approach to extend these single-server schemes to their partial-distributed-server versions is naively duplicating the single server and its database to multiple servers with individual duplicate databases; this naive approach might underestimate the potential of their other possible extensions, but it provides a basic insight of the performance of these schemes.

In Table 3, we apply this naive extension approach on Chen et al.’s scheme [22] and Ohkubo et al.’s scheme [4]. Apparently this naive extension of [22] could no longer provide correct authentication between tags and servers, because their servers would quickly lose synchronization of
their states with the tags after initialization. In [22], a server keeps two stateful values synchronized with that of each tag, one value is the possible next value of the state and the other is the last successful state; so if a server has been successively authenticating a tag, then all the other servers lose their synchronization.

Interestingly, if we apply the same naive extension on Ohkubo et al.’s scheme, the extension still keeps its accuracy of authentication, anonymity, un-traceability, DOS attack resistance, and forward secrecy; it is because the authentication in Ohkubo et al.’s scheme depends on two independent hashing chains so that any server could exhaustively trace down all the chains of all tags to identify an anonymous tag; but, we should notice that two main weaknesses of this approach- the poor computational performance of exhaustive search and the probability of successful replay attacks amplified in the partial-distributed-server environments. In the partial-distributed-server model, an eavesdropped session from one server could be used to launch replay attacks on any another server; and a server needs more efforts to re-synchronize and search a tag in the out-of-synchronized key chains of all possible tags in partial-distributed-server environments. From this table, we can see that it is more challenging to design an RFID authentication protocol with desirable properties in the partial-distributed-server model, and our scheme achieves excellent performance in terms of security, efficiency, server’s maintenance, cost and scalability.

5.3 Computational Performance Comparison among ECC-Based Schemes

Now we make a simple comparison of computations among ECC-based schemes in Table 4, where $T_P$ denotes that for one PRNG, $T_{sq}$ denotes that for one squaring modulo, $T_{sqr}$ denotes that for one square root solving, and $T_{ECC}$ denotes that for one ECC decoding operation. From Table 4, Chien-Laih’s scheme can support $O(k)$ number of tags while our scheme can support $O(2^k)$ number of tags, where $k$ is the dimension of the codes. Chien-Laih’s scheme did not provide forward secrecy while ours do. Our scheme additionally depends on Rabin cryptosystem, compared to Chien-Laih’s scheme; therefore, our scheme apparently demands one more squaring on tag, one more the Chinese remainder theorem function and one more ECC decoding on the average on server. Since both the square root computation and the ECC decoding are simple for server, the additional computational cost on server is negligible. However, the improvement of the number of supported tags from $O(k)$ to $O(2^k)$ is very significant. Furthermore, our scheme supports...
forward secrecy while Chien-Laih’s scheme does not.

Regarding the communication, we consider both the number of steps and the length of the messages. To achieve mutual authentication, the optimal number of steps is three, and all the three ECC-based schemes require three steps only. Let $L_N$ denote the bit length of the public key of Rabin cryptosystem, $L_{ch}$ denote the bit length of a challenge, $L_{PRNG}$ denote that of a PRNG output, and $L_m$ denote that of a code word. The message length of both our scheme and the scheme [26] is $L_N + L_{ch} + L_{PRNG}$, while that of the scheme [11] is $2L_m + L_{ch} + 5L_{PRNG}$. We take one possible set of parameter setting as an example: $L_m = 256$, $L_{ch} = L_{PRNG} = 80$, and $L_N = 1024$. Then both our scheme and the scheme [26] require 1184 bits totally, while the scheme [11] requires 996 bits. We can see that the difference of message length is not significant.

Consider the implementation requirement of key space and gate count of our scheme on low-cost cryptographic tags. If we select the linear code as $C(n = 128, k = 58, d = 21)$, the key length is 80 bits and the Rabin modulus being 1024 bits, then our scheme could support $2^{58}$ tags. The space requirement per tag of our scheme is $128 + 2 \times 80 + 1024 = 1312$ bits, and the effective bit length of randomness of artificial error vectors is 47.7. Another case is that if we select the linear code as $C(n = 256, k = 120, d = 35)$, and the key length is 80, then the effective bit length of the randomness of artificial error vectors is 86.9, the key space is 1440 bits and the number of supported tags is $2^{130}$. Now we examine the storage space feasibility of our scheme on some popular RFID tags on the market like Gen2 [31], ISO 15693 [33] and Mifare S50 [32]. The memory of the popular Gen 2 tag on the market is 512 bits, ISO15693 could support 256 bytes ~ 8K bytes, and Mifare S50 and S70 respectively support 1K and 4K bytes memory. This implies that the storage space on the current ISO 1569, Mifare S50 and S70 could support the space requirement of our scheme, but Gen2 does not. However, new Gen2 standard called Gen2V2 has been ratified in December 2013 [34], and it could support much larger space for practical implementation.

We further examine the feasibility of the computational requirement of our scheme on some low-cost cryptographic tags. In our scheme, the most computationally expensive operation on tag is modular squaring operation. According to [35],[36], a 512-bit modular multiplier could be implemented using around 12,500 gates, and a 1024-bit modular multiplier is estimated around 24,000 gates. A very low cost tag like Gen2 was estimated to have 7,500~15,000 gates and 2,500~5,000 gates of them could be used for security algorithms. Based on these figures, our scheme with $N=512$ or $N=1024$ bits are still infeasible for a very low cost tag like Gen2. However, we note that the encryption of Rabin cryptosystem is modulo squaring which is a special case of modulo multiplication; It is believed that a dedicated design of modulo squaring would require much fewer gates than modulo multiplication. As the manufacturing technology advances and the large competing market emerges, the prices of some cryptographic tags like Mifare classic (which supports PRNG, 3-way handshake authentication and Mifare classic security [37]), Mifare DesFire (which additionally supports DES/3DES) and even Mifare SmartMx with RSA colud be dropped to less than 10 cents on Alibaba market [38]. Our proposed scheme is quite attractive to these low cost cryptographic RFID tags now.

### Table 4
Computational performance comparison among ECC-based schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chien’s scheme [26]</td>
<td>$5T_p + T_{sq}$</td>
<td>$5T_p + T_{sq} + 2T_{rec}$</td>
<td>$O(2^k)$</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Chien-Laih scheme [11]</td>
<td>$4T_p$</td>
<td>$4T_p + 1.5T_{sq}$</td>
<td>$O(k)$</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Our scheme</td>
<td>$5T_p + T_{sq}$</td>
<td>$5T_p + T_{sq} + 2T_{rec}$</td>
<td>$O(2^k)$</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

| $C_1$: Comput./tag $^1$ | $C_2$: Comput./server         | $C_3$: Number of tags          | $C_4$: Forward secrecy | $C_5$: Exhaustive Computation and matching | $C_6$: Support partial-distributed-server |

### Table 5
Communication performance comparison among ECC-based schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Number of steps</th>
<th>Total message length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chien [26]</td>
<td>3</td>
<td>$L_N + L_{ch} + L_{PRNG}$</td>
</tr>
<tr>
<td>Chien-Laih [11]</td>
<td>3</td>
<td>$2L_N + L_{ch} + 5L_{PRNG}$</td>
</tr>
<tr>
<td>Our scheme</td>
<td>3</td>
<td>$L_N + L_{ch} + L_{PRNG}$</td>
</tr>
</tbody>
</table>

6. Conclusions and Future Work

This paper has proposed the first RFID authentication scheme with anonymity, forward secrecy and un-traceability in the partial-distributed-server model. The proposed scheme not only requires moderate-lightweight computations on tag but also eliminates the exhaustive computation and matching computations on servers to identify an anonymous tag. The proposed scheme has good scalability performance: it could support the number of tags $O(2^k)$.

Four interesting topics of future work are described as follows. The first is to explore other possible extensions (non-naïve extension) of previous schemes to study whether they are suitable in the partial-distributed-server model. The second is to explore the possibility of simultaneously sup-
port RFID authentications and ownership transfer function in the partial-distributed-server model. That is, it would be desirable if we could authenticate RFIDs with the above mentioned properties in the partial-distributed-server model, while we also allow servers to transfer ownership of a tag from one server to another. Another issue is that, in our scheme and other previous schemes, if one server is compromised, then the whole system is compromised. Since our scheme is targeted at partial-distributed-server model (multiple independent servers), the third interesting open question is that whether we can keep the security properties even if some of the servers are compromised. The fourth one is to study formal model that well captures the security and privacy in the partial-distributed-server model.

Acknowledgments

This project is partially supported by the National Science Council, Taiwan, R.O.C., under grant no. NSC 101-2218-E-260-001.

References

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, and an associate professor of ChaoYang University of Technology during 200309–200609. Now he is a member of the Chinese Association for Information Security, an IEEE member and a professor of National Chi Nan University, and was the department head of the Information management department 2008.8–2011.7. His research interests include cryptography, networking, network security and RFID security.

Tzong-Chen Wu received B.S. degree in Information Engineering from National Taiwan University in 1983, M.S. degree in Applied Mathematics from National Chung Hsing University in 1989, and Ph.D. degree in Computer Science and Information Engineering from National Chiao Tung University in 1992. Dr. Wu joined the Department of Information Management, National Taiwan University of Science and Technology (NTUST) in 1992, and he served as the professor from February 1997 to March 2014. From March 2014 till now, he serves as the distinguished professor in the Department of Information Management, NTUST. Professor Wu is the members of IEEE, ACM, IEICE and the Chinese Cryptology and Information Security Association (CCISA). He served as the president of CCISA from June 2003 to May 2009. His current research interests include information security, mobile security, cryptographic protocols and related topics.

Chien-Lung Hsu received a B.S. degree in business administration, an M.S. degree in information management, and a Ph.D. degree in information management from the National Taiwan University of Science and Technology, Taiwan in 1995, 1997, and 2002, respectively. He was an Assistant Professor and an Associate Professor in the Department of Information Management, Chang Gung University (CGU), Taiwan from 2004 to 2007 and from 2007 to 2011, respectively. Currently, he is a Professor in the Department of Information Management, Chang Gung University since 2011. He is also the director of Chinese Cryptology Information Security Association (CCISA, Taiwan), the chair of Education Promotion Committee of CCISA, the director of Program of RFID Applications in Logistics Supply Chain Management of CGU, the director of Program of Information Security with Medical Applications of CGU, the director of Division of Instructional Support of Computer Center of CGU, the researcher of Healthy Aging Research Center (HARC) of CGU, the researcher of Elder Industry Development and Research Center (EIDRC) of CGU. His current research includes cryptography, information security, wireless sensor network, mobile commerce, digital forensics, vehicular system security, healthcare system and user acceptance, smart home system, and etc.