RAKA: Revocable Authenticated Key Agreement System for Mobile Social Networks*

TZU-HSIN HO, Shiu-PynG Winston Shieh and ChiEn-Chao Tseng
Department of Computer Science
National Chiao Tung University
Hsinchu, 300 Taiwan

Mobile social networks (MSNs) provide users with potential indirect relationships such as friends of friends and similar transitive relationships. However, adversaries may be involved in potential relationships, the trustworthiness of the relationships must be ensured. Thus, the disclosure of private information must be well controlled, and users have to be authenticated. In this paper, we propose a revocable authenticated key agreement system (RAKA), which does not require intensive computation for authentication in MSNs. RAKA not only provides a simple rekeying process but also aggregates multiple successive private key update operations into a single rekeying process. The proposed revocation method incurs low communication overhead on mobile devices using a single message broadcast. RAKA is proved to be resilient against malicious attacks.

Keywords: authenticated key agreement, revocation, mobile social networks (MSNs), rekeying process, secure communication

1. INTRODUCTION

Mobile social networks (MSNs) enable social networking in mobile phones, bringing much excitement to smartphone and tablet markets. In addition to the essential capability offered by Internet-based social networks, MSNs enable mobile users with some potential relationship to extend their online social networks through opportunistic meets in the physical vicinity. Potential relationship can be a chain of friends-of-friends, leading a user to a potential friend he would like to meet. It also can be the users of same interest to share specific content. Through potential relationship, two or more mobile users in the same vicinity can interact with each other to share specific information [12, 30].

Because users can interact with others through potential relationships in MSN environments, two security problems may arise [23]. First, users may interact with malicious adversaries who disguise themselves as people willing to establish potential relationships. To prevent this from happening, relationship validation should be assured before the communication is started. Second, a user may intend to only selectively disclose his/her private information to specific users. To simultaneously resolve the two security problems, an authenticated key agreement protocol allows each participating party to contribute a partial secret to derive the shared session key for communication. To establish authenticated and secure communication between users with a potential relationship, the authenticated key agreement protocol is a promising fundamental approach.

Received November 24, 2014; revised March 10 & May 13, 2015; accepted July 13, 2015.
Communicated by Hung-Min Sun.
* This work was supported in part by Ministry of Science and Technology, Taiwan, under the grants NSC 101-2221-E-009-031-MY3 and NSC 100-2221-E-009-072-MY3.
However, applying authenticated key agreement protocol to MSN environments poses several serious challenges to relationship revocation, and at least three problems must be considered [21]. First, the revocation of relationship with adversaries must be assured. Since the authenticated key agreement protocol is provided between users with a potential relationship, a malicious user may disseminate false information using the potential relationship. Therefore, revoking malicious adversaries from a potential relationship becomes a major challenge. Second, rekeying information delivery must be atomic. Users with wireless connections may experience unpredictable disconnection to central key servers because of user mobility. In this case, users may not be able to obtain key update information. Consequently, a user may not have up-to-date key information, leading to inconsistency with other users. Third, the communication and computational overhead of a rekeying process should be low because of the limited resource and capability of mobile devices. All these issues will be further addressed in subsequent sections.

Various security protocols have been proposed [10, 17, 20, 25]. In the route-based authentication scheme for data delivery presented in [20], entity authentication was not addressed. Entity authentication is mentioned in other systems [10, 17], however, these systems are not concerned with the situation where users with potential relationships may not know the identities of others. Considering the potential relationship, a trust establishment method [22] is proposed which takes advantage of relationship as the key information. However, the relationship is only confined by time and space. Since potential relationships such as bike-interest, friend chains can be represented as attributes, the trustworthiness of potential relationship can be determined according to the representative attributes. Therefore, attribute-based authenticated key agreement protocols [1, 7, 11, 15, 19, 28] can support authentication and secure communication using potential relationship. However, complicated computation for session key is required in these studies.

For revocation, methods [2, 9] have been proposed allowing users to revoke external access to data using a revocation list. However, a user may only interact with other users by using potential relationships. In this case, the user may not know the identity of his communicating party, let alone know whether the identity of the party is in the revocation list. Therefore, these methods are not suitable for MSNs. Using authority for key update is another choice [4, 16, 18, 26, 31]. Nevertheless, users must periodically contact authorities [4, 26] to set the expiration time for users’ private keys. Consequently, revocation with immediate revoking capability has been provided [16]. In this method, multiple secure channels may be required for rekeying. This is because the total number of secure channels for sending key-update information may increase with respect to the number of non-revoked users. To reduce the overhead of establishing multiple secure channels, a method was proposed in [31] to revoke a user by broadcasting revocation messages.

In this paper, we design a revocable authenticated key agreement system (RAKA) to support secure communication between users with potential relationships in MSNs. In this system, an implicitly authenticated key agreement protocol is proposed which keeps user privacy at the same time. An efficient rekeying process is designed for user revocation in the MSN environment. To lower the communication overhead of transmitting key-update information, only a short key-update information is broadcast to users.
To reduce the computational overhead of constrained mobile devices, the RAKA aggregates multiple successive key update operations into a single rekeying process. Furthermore, we present an algorithm that the nearby parties in remote area can help with each other to update key even though parties are disconnected to the central key server. The proposed protocol is proved to be secure. As shown in the performance analysis, RAKA is efficient and incurs low communication overhead and computational complexity.

The rest of this paper is organized as follows. In Section 2, the bilinear pairing, Decisional Bilinear Diffie-Hellman assumption and Fuzzy-IBE cryptography used in our system are depicted. Section 3 describes the proposed revocable authenticated key agreement system. Security analysis and proof is given in Section 4. Section 5 evaluates the performance, and Section 6 concludes this work.

2. CRYPTOGRAPHIC PRELIMINARIES

In this section, we introduce cryptographic techniques that will be used in this paper. In Section 2.1, we describe bilinear pairing [5] that is used in the underlying cryptosystem of this work. In Section 2.2, Fuzzy-IBE [27] is introduced briefly, which is used to protect key-update information in the proposed system.

2.1 Bilinear Pairing

Bilinear pairing has been widely used to build various security schemes [5, 27]. Let \( G_1 \) be a cyclic additive group with a generator \( P \), \( G_2 \) be a cyclic multiplicative group, and \( G_1 \) and \( G_2 \) have the same prime order \( p \). In addition, let \( \mathbb{Z}_p = \{0, 1, \ldots, p-1\} \) and \( \mathbb{Z}_p^* = \{1, 2, \ldots, p-1\} \). A function \( \hat{e}: G_1 \times G_1 \rightarrow G_2 \) is called an admissible bilinear map if it satisfies the following properties:

1. Bilinearity: \( \forall R, Q \in G_1 \) and \( \forall a, b \in \mathbb{Z}_p^* \), \( \hat{e}(aR, bQ) = \hat{e}(aR) \cdot \hat{e}(bQ) = \hat{e}(R)^a \cdot \hat{e}(Q)^b \).
2. Non-degeneracy: \( \hat{e}(P, P) \neq 1 \).
3. Computability: An efficient algorithm exists to compute \( \hat{e}(R, Q) \) for all \( R, Q \in G_1 \).

An admissible bilinear map can be constructed by modified Weil pairing [5] or Tate pairing [13].

In order to prove the security of the proposed system, we assume that the Decisional Bilinear Diffie-Hellman (DBDH) is a hard problem. The assumption [27] is described as follows:

**Definition 1 (DBDH Assumption)** Suppose a challenger chooses \( a, b, c, z \in \mathbb{Z}_p \) at random. No polynomial-time adversary is able to distinguish the tuple \((A=aP, B=bP, C=cP, Z=e(P, P)^{abc})\) from \((A=aP, B=bP, C=cP, Z=e(P, P)^z)\) with more than a negligible advantage.

2.2 Fuzzy-IBE Cryptography

In Fuzzy-IBE cryptography, the private key of a user is associated with a set of at-
tributes. The data are encrypted associated with a set of attributes. A user can decrypt ciphertext if and only if the number of overlapping attributes between attributes with the user’s private key and those with the ciphertext exceeds a threshold. Let $d$ be the threshold of overlapping attributes, $\omega'$ be an attribute set associated with ciphertext, and $\omega$ be an attribute set associated with a private key. The ciphertext can be decrypted if and only if $|\omega' \cap \omega| \geq d$.

3. REVOCABLE AUTHENTICATED KEY AGREEMENT SYSTEM

The proposed revocable authenticated key agreement system (RAKA) is illustrated in this section.

3.1 System Model

Below, we sketch the designed authenticated key agreement system as depicted in Fig. 1. The system mainly performs the authenticated key agreement protocol to provide secure communication between users with potential relationship. Moreover, it provides a rekeying process to manage user revocation. The system consists of two entities:

1. Trusted Server. The trusted server (TS) is the authority managing the attribute set, issuing and handling keys. It generates public parameters and private keys for users. The TS is also responsible for generating key-update information when user revoca-
tion occurs. It is the only party that is fully trusted by all entities participating in the system.

2. Users. Any user can interact with other users with potential relationships in the same vicinity without access to the TS. They can be either senders or receivers. As a sender, the user initiates sending his contributed information for establishing a common session key to the other party. As a receiver, the user responds with his contributed information for common session key to the sender.

In the initialization phase, the TS sets the potential relationships of each user, determines system-wise parameters, and generates secrets and public parameters. The secrets are only known by TS, and the public parameters are published to all users. Furthermore, the TS generates a private key for each user, which is transmitted to the user via a secure channel. When two users engage in an interaction, the sender initiates his contributed information to the receiver, and the receiver responds with his own contributed information. If sender has certain attribute the same as the receiver has, both of them can authenticate implicitly and establish a shared session key. For revocation, the TS generates key-update information and protects it by encryption. The encrypted key-update information can then be securely broadcast. Only non-revoked users can decrypt the key-update information and use it to update their private keys. The three major functions of the proposed system, including system initialization, the authenticated key agreement protocol, and user revocation, are described in the subsequent subsections.

3.2 Security Requirements

Adversaries may eavesdrop the communication or launch active attacks. To resist malicious attacks, the system should meet the following security requirements:

1. Implicit key authentication: Each user should ensure that the other user is the exact party in possession of the session key.
2. User privacy: The identity of the involved users should be hidden from others.
3. Resistance to passive attack: A passive adversary attempting to eavesdrop on a public communication channel should not obtain any useful session information.
4. Resistance to known session keys attack: An adversary should learn nothing about a fresh session key even if he obtains some session keys of other sessions.

3.3 System Initialization

The TS first determines the parameter $d$ as the threshold for decryption and then generates public parameters and private keys for users. Let $G_1$ be the subgroup of the basing elliptic curve group $E$. Let the order of $G_1$ be $p$ and that of $E$ be $\#E$. Hence, $h = \frac{\#E}{p}$ is the cofactor of the group $G_1$. Two public functions are used in the system, including

1. $\pi: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{Z}_p^*$ is a hash function.
2. $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is an admissible bilinear map.
The details are introduced below.

(1) **Parameter generation:** In the proposed system, each potential relationship is assigned a unique attribute, called social ID, and each user is assigned a unique attribute, called user ID. Here, the attribute of social $i$ is denoted as $a^i_{\text{social}}$, and the user attribute of user $i$ is denoted as $a^i_{\text{user}}$. In addition, the TS chooses $d - 1$ attributes to form a dummy attribute set, denoted as $\mathcal{D}_A$, which is a common attribute set known by all users. All dummy attributes cannot be reused to present other attributes. For each attribute $i$, the TS chooses a random secret $t_i \in \mathbb{Z}_p$. The TS also chooses two master secrets $\alpha, \beta$ randomly from $\mathbb{Z}_p$, where $\alpha$ is the master secret for encrypting key-update information, and $\beta$ is the master secret for generating session keys. A public parameter set $PP$ denoted as

$$PP = \left\{ \hat{e}(P, P)^\alpha; \left\{ t_i P \text{ for all attribute } i \right\} \right\}$$

is published to all users.

(2) **Private key generation:** The TS generates a private key for each user. Because each user can have several potential relationships, each user $i$ holds an attribute set of social IDs to represent all his relationships, denoted as $G_i$. Let $\omega$ denote the attribute set $G_i \cup \{a^i_{\text{user}}\} \cup \mathcal{D}_A$. For each user $i$, the TS randomly chooses a personal polynomial $f_i(\cdot)$ of $d-1$ degree and with $f_i(0)=\alpha$. The private key $PrK_i=(D_1, D_2)$ for user $i$ is expressed as follows:

$$D_1 = \left\{ D^i_1 = \frac{f_i(k)}{t_k} P \text{ for } k \in \omega \right\},$$

$$D_2 = \left\{ D^i_2 = \frac{\beta - f_i(k)}{t_k} P \text{ for } k \in G_i \right\}.$$

The private key is transmitted to the user via a secure channel or preloaded to the user.

### 3.4 Proposed Authenticated Key Agreement Protocol

Let user $A$ and user $B$ have a certain potential relationship represented by the attribute $a^i_{\text{social}}$. For simplicity, the notation $k = a^i_{\text{social}}$ is used in the following sections. User $A$ owns the private key component corresponding to the attribute $k$, denoted as $(D^1, D^2)$, and user $B$ holds the private key component corresponding to the attribute $k$, denoted as $(D^3, D^4)$. And let $T_k = t_k P$ be the public parameter corresponding to the attribute $k$. Only users with the same attribute can use their own corresponding private keys and the public information provided by the other party to establish a common session key successfully. When participating users $A$ and $B$ want to establish common session key, the protocol shown in Fig. 2 proceeds as follows:
1. User A chooses \( x \) randomly from \( \mathbb{Z}_p^* \), computes \( R_A = xT_k \), and sends \( R_A \) to user B.

2. User B chooses \( y \) randomly from \( \mathbb{Z}_p^* \), computes \( R_B = yT_k \), and sends \( R_B \) to user A.

Subsequently, user B uses the received \( R_A \) to compute
\[
s_{BA} = \pi(R_A, R_B), \quad s_B = \pi(R_B, R_A),
\]
and the common session key
\[
\hat{k}_{BA} = k_B + k_A.
\]

3. User A uses the received \( R_B \) to compute
\[
s_{AB} = \pi(R_A, R_B), \quad s_A = \pi(R_B, R_A),
\]
and the common session key
\[
\hat{k}_{AB} = k_A + k_B.
\]

3.5 User Revocation

3.5.1 Basic revocation method

To revoke a user from a potential relationship represented by the attribute \( k \), two basic operations are conducted: (i) update the secret, public parameter related to the attribute \( k \); and (ii) update the private key component corresponding to the attribute \( k \) of each non-revoked user. Operation (i) is performed by the TS, and operation (ii) is delegated to users. The TS generates a short key-update information and broadcasts it. According to the key-update information, non-revoked users can themselves update their private keys. The operations are detailed as follows: The TS chooses a random number
Tzu-Hsin Ho, Shiu-Hpyng Winston Shieh and Chien-Chao Tseng

738

As a new secret for \( k \) and updates the public parameter \( T_k = t_k'P \) accordingly. The key-update information \( RK_k \) is given by the TS, \( RK_k = (t_k \cdot t_k'^{-1}) \).

To prevent revoked/unauthorized users from intercepting key-update information during delivery, the TS encrypts \( RK_k \) before broadcasting it. There are plenty of selectively broadcast schemes \([6, 8, 14, 24]\) available such that all but the revoked/unauthorized users can decrypt \( RK_k \). In our revocation solution, we use a list to selectively include non-revoked users to access key-update information based on FuzzyIBE. Setting all non-revoked users into the list, TS assigns the set \( \omega' \) with attributes \( a_i' \) of each user \( i \) in the list and encrypts the key-update information \( RK_k \) with \( \omega' \). The encrypted \( RK_k \), denoted as \( E(RK_k) \), is computed as follows:

\[
E(RK_k) = \left\{ \omega' = \{a_i': \text{all user } i \text{ in the list}\}; \right.
\]

\[
E_1 = RK_k \left[ (f(P, P)')^r \right];
\]

\[
E_2_j = r(t_jP) \text{ for all } j \in \omega' \cup DA\}
\]

Upon receiving the encrypted key-update information \( E(RK_k) \), the receiver checks if he owns the user ID in the received attribute \( \omega' \). If Yes, the receiver can perform the decryption successfully in the following

\[
D(E(RK_k)) = \frac{E_1}{\prod_{j \in \omega' \cup DA} E(D1_j, E2_j)} = RK_k \quad \text{where}
\]

\[
a_j = \prod_{j \in \omega' \cup DA} \epsilon(D1_j, E2_j)^{-1} \quad \text{is a Lagrangian coefficient that satisfies } \sum_{j \in \omega' \cup DA} a_j \epsilon_j = f_j(0).
\]

This indicates that revoked users cannot decrypt the key-update information. Hence, only each non-revoked user \( i \) can correctly decrypt \( E(RK_k) \), and updates his private key components using \( RK_k \) as follows:

\[
D1_i = \left( t_k \cdot t_k'^{-1} \right) \cdot f_i(k) \cdot t_k^{-1} \cdot P = \frac{f_i(k)}{t_k} P,
\]

\[
D2_i = \left( t_k \cdot t_k'^{-1} \right) \cdot (\beta - f_i(k)) \cdot t_k^{-1} \cdot P = \frac{\beta - f_i(k)}{t_k} P.
\]

3.5.2 Advanced revocation method

To reduce the computational overhead of users for updating private key repeatedly, we aggregate multiple successive key update operations into a single process. The TS associates the version number \( ver_i \) with the encrypted key-update information \( E(RK_k) \), denoted as \( E(RK_k)_i \) and broadcasts \( E(RK_k)_i \). Each user can record all versions of the received encrypted key-update information. The process for users updating key from version \( i \) to version \( j \) is described as follows:
1. Decrypt $E(RK_k)$. If this user is a revoked user, the decryption fails. Otherwise, the user can decrypt $E(RK_k)$ to obtain $RK_k$ and proceed to step 2.

2. Compute $D1_k \leftarrow RK_k \cdot D1_k$; $D2_k \leftarrow RK_k \cdot D2_k$.

Compute $i = i + 1$. If $j + 1 = i$, then the user succeeds and exits. Otherwise, the user returns to step 1.

Moreover, with the recorded encrypted key update information, a user can assist others who have older private keys to update their private keys. The algorithm is depicted in Table 1. Any user can perform the algorithm to assist or be assisted by other parties. A user initiating the algorithm is called an initiator. The other party is called a responder. In the algorithm, the initiator requests the current version of encrypted key-update information from the responder. After obtaining response, the initiator compares the version he owns with the version the responder has. If initiator’s version is higher than that of the responder, the initiator sends sufficient successive encrypted key-update information to the responder. The responder can then update his private key by using the function $ReKey$. If the initiator’s version is lower than that of the receiver, the initiator obtains encrypted key-update information from the responder. Consequently, the initiator can complete his or her private key update by using the function $ReKey$. Because key-update information is adequately protected by encryption, only non-revoked users can decrypt it. Thus, only non-revoked users can use key-update information to update their private keys.

### Table 1. Algorithm Aggregative Key Update.

<table>
<thead>
<tr>
<th>Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>// Initiator $A$ requests version info to responder $B$</td>
<td></td>
</tr>
<tr>
<td>// Compare the version of $A$ with the version of $B$</td>
<td></td>
</tr>
<tr>
<td>If $A.\text{ver} &gt; B.\text{ver}$ then</td>
<td></td>
</tr>
<tr>
<td>send $E(RK_k)<em>{A.\text{ver}+1} \sim E(RK_k)</em>{B.\text{ver}}$ to $B$</td>
<td></td>
</tr>
<tr>
<td>else</td>
<td></td>
</tr>
<tr>
<td>receive $E(RK_k)<em>{A.\text{ver}+1} \sim E(RK_k)</em>{B.\text{ver}}$ from $B$</td>
<td></td>
</tr>
<tr>
<td>$ReKey(k, A.\text{ver}+1, B.\text{ver})$</td>
<td></td>
</tr>
<tr>
<td>end if</td>
<td></td>
</tr>
<tr>
<td>Function $ReKey(k, \text{ver1}, \text{ver2})$</td>
<td></td>
</tr>
<tr>
<td>for $i=\text{ver1}$ to $\text{ver2}$ do</td>
<td></td>
</tr>
<tr>
<td>if decrypt $E(RK_k)$, then</td>
<td></td>
</tr>
<tr>
<td>$D1_k \leftarrow RK_k \cdot D1_k$; $D2_k \leftarrow RK_k \cdot D2_k$</td>
<td></td>
</tr>
<tr>
<td>else</td>
<td></td>
</tr>
<tr>
<td>Caller is not a valid member</td>
<td></td>
</tr>
<tr>
<td>end for</td>
<td></td>
</tr>
<tr>
<td>end Function</td>
<td></td>
</tr>
</tbody>
</table>

4. SECURITY ANALYSIS

In this section, we will prove that the proposed authenticated key agreement protocol is secure.
On the basis of the Bellare and Rogaway model [3], we construct the following security model. In the model, adversary completely controls the network communication. Let oracle $\Pi'_{A,B}$ denote the behavior of user $A$ when conducting a protocol session $s$ for communicating with user $B$. For each $\Pi'_{A,B}$, a record, denoted by view, is kept to track its input/output messages. Let the adversary $\mathcal{A}$ interact with the participants through oracle access. The oracle queries are listed as follows:

1) $\text{Send}(\Pi'_{A,B}, X)$. The query models active attacks, where the adversary can send a message $X$ to $\Pi'_{A,B}$. The adversary gets the response message according to the sending message $X$. The adversary can ask $A$ to initiate a session with $B$ by setting $X$ as an empty string.

2) $\text{Reveal}(\Pi'_{A,B})$. The query models the exposure of the session key of protocol session $S$.

3) $\text{Corrupt}(A)$. The query models the exposure of the private key of user $A$.

4) $\pi(X, Y)$. The query generates a hash value from messages $X$ and $Y$.

5) $\text{Test}(\Pi'_{A,B})$. The query models the advantage of the adversary in guessing the session key. The adversary $\mathcal{A}$ can make a query to $\Pi'_{A,B}$ for distinguishing a real session key from a random key. The oracle $\Pi'_{A,B}$ sends the real session key or a random key to $\mathcal{A}$ according to a random coin bit $b \in \{0,1\}$. $\mathcal{A}$ must guess the value of $b$.

After the adversary $\mathcal{A}$ has made all queries it wants to ask, $\mathcal{A}$ executes $\text{Test}$ to query a fresh oracle which has accepted and have not been queried $\text{Reveal}$, to get this oracle’s or the matching oracle’s session key. Finally, $\mathcal{A}$ outputs the bit $b' \in \{0,1\}$ as its coin guess. The advantage of the adversary $\mathcal{A}$ in guessing the session key is defined as

$$\text{Adv}^{\mathcal{A}}(k) = \left| Pr[b = b'] - \frac{1}{2} \right|$$

where $k$ is the security parameter.

**Definition 1:** A protocol is a secure authenticated key agreement protocol if the following conditions are satisfied for any adversary:

1. If two uncorrupted oracles $\Pi'_{A,B}$ and $\Pi'_i$ have matching conversations, then both oracles always accept and hold the same session key, which is distributed uniformly over the key space.
2. $\text{Adv}^{\mathcal{A}}(k)$ is negligible.

**Theorem 1:** The proposed protocol is a secure authenticated key agreement protocol assuming that $\pi$ is an independent random oracle.

**Proof:** Condition 1 holds as follows. We consider the session key generation between oracles $\Pi'_{A,B}$ and $\Pi'_i$ for the relationship $k$. Let user $A$ hold the private key components $(D_1^A, D_2^A)$. Upon receiving $R_B$ from the $\text{Send}$ query, $\Pi'_{A,B}$ chooses $x \in R \subset Z_p$ and generates $R_A = xT_A$. Then, two hash values $s_A$ and $s_B$ from $R_A$ and $R_B$ were generated by $\pi(R_A, R_B)$ query and $\pi(R_A, R_B)$ query, respectively. Next, $\Pi'_{A,B}$ produces the shared secret
sk_{AB} as follows.

\[ sk_{AB} = \hat{e}\left((x+s_A) h\left(D_1^A + D_2^A\right), s_B T_k + R_B\right) \]
\[ = \hat{e}\left((x+s_A) h\left(\frac{p_A(k)}{t_k} P + \frac{\beta - p_A(k)}{t_k} P\right), s_B T_k + y T_k\right) \]
\[ = \hat{e}\left((x+s_A) h\left(\frac{p_A(k)}{t_k} P + \frac{\beta - p_A(k)}{t_k} P\right), (s_B + y) t_k P\right) \]
\[ = \hat{e}\left(P, P\right)^{(x+s_A) \frac{\beta}{t_k} (s_B + y)} \]
\[ = \hat{e}\left(P, P\right)^{(x+s_A) \beta (s_B + y)} \]

Let user B hold private key components \((D_1^B, D_2^B)\). Upon receiving \(R_A\) from the \(Send\) query, \(\Pi_{B,A}\) chooses \(y \in \mathbb{Z}_p\) and generates \(R_B = y T_k\). Subsequently, the \(\pi(R_A, R_B)\) and \(\pi(R_B, R_A)\) queries generate two hash values, \(s_A\) and \(s_B\), from \(R_A\) and \(R_B\), respectively. Next, \(\Pi_{B,A}\) produces the shared secret \(sk_{BA}\) as follows:

\[ sk_{BA} = \hat{e}\left((y+s_B) h\left(D_1^B + D_2^B\right), s_A T_k + R_A\right) \]
\[ = \hat{e}\left((y+s_B) h\left(\frac{p_B(k)}{t_k} P + \frac{\beta - p_B(k)}{t_k} P\right), s_A T_k + x T_k\right) \]
\[ = \hat{e}\left((y+s_B) h\left(\frac{p_B(k)}{t_k} P + \frac{\beta - p_B(k)}{t_k} P\right), (s_A + x) t_k P\right) \]
\[ = \hat{e}\left(P, P\right)^{(y+s_B) \frac{\beta}{t_k} (s_A + x)} \]
\[ = \hat{e}\left(P, P\right)^{(y+s_B) \beta (s_A + x)} \]

Because \(sk_{AB}\) is identical to \(sk_{BA}\), both parties generate the same session key under the random oracle \(\pi\). If both parties faithfully transmit messages to each other, both oracles will get the same session key. Moreover, the session key is distributed according to random elements \(x\) and \(y\) from \(\mathbb{Z}_p\), where \(x\) is generated by \(\Pi_{A,B}\) and \(y\) is generated by \(\Pi_{B,A}\). The product of these elements is also a random element in \(\mathbb{Z}_p\). Because the exponent part of the shared secret is random, the shared secret is distributed uniformly over \(\mathbb{Z}_p^2\).

Condition 2 holds under the DBDH assumption. By way of contradiction, suppose that \(Adv^c(k)\) is non-negligible. We construct an adversary \(C\) from \(A\) to solve the DBDH problem with a non-negligible advantage. \(C\) takes \(<P, uP, vP, wP, \hat{e}(P, P)>\) as the input, where \(\hat{a}\) can be \(uvw\) or a random number \(z\) in \(\mathbb{Z}_p\). Next, \(C\) attempts to distinguish \(\hat{e}(P, P)\) from \(\hat{e}(P, P)\). \(C\) chooses \(A\) and \(B\) from a set of entities and sessions \(s\) and \(t\) randomly. \(C\) generates all entities’ private keys at random excluding that of user \(A\). \(C\) selects \(c_1, c_2 \in \mathbb{Z}_p\) and creates the private key \(D_1^c = c_1(uP), D_2^c = c_2(vP)\) for user \(A\). \(C\)
answers all A’s queries itself. C answers all the Corrupt queries as specified by the protocol, except for the Corrupt queries from A asking user A or B. If these queries occur, C aborts. C also answers all the Send queries including queries of $\Pi_{A, B}$ and $\Pi_{B, A}$ as specified by the protocol. When A asks the Send query regarding the first flow of $\Pi_{B, A}$, C answers $R_B = c_3wP$, where $c_3, c_4 \in \mathbb{Z}_p$. Here, $\pi(R_B, R_B)$ and $\pi(R_A, R_A)$ can be assumed to take $c_3$ and $c_4$ uniformly from $\mathbb{Z}_p$, respectively. Subsequently, C sets $\hat{e}(P, P)^d = e(P, P)^{x(c_3+c_4+c_5+c_6)}$, where $x$ is a random number in $\mathbb{Z}_p$, and sends $\hat{e}(P, P)^d$ to A. If A determines $\hat{e}(P, P)^d$ to be a session key, then C guesses $\hat{e}(P, P)^{uvw}$ coming from $\hat{e}(P, P)^{zw}$. Otherwise, C guesses $\hat{e}(P, P)^{zw}$ coming from $\hat{e}(P, P)^{zw}$. Thus, algorithm C can distinguish $\hat{e}(P, P)^{zw}$ from $\hat{e}(P, P)^{zw}$ with a non-negligible advantage, which contradicts the DBDH assumption.

5. PERFORMANCE ANALYSIS

This section presents a comparison of the proposed protocol with existing protocols, including Wa09 [28], SHDFM [1], and Yo10 [19] in terms of the communication overhead and computational complexity. Table 2 gives the comparison results.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication overhead</td>
<td>$O(n</td>
<td>m_{G_1}</td>
<td>+</td>
<td>m_{G_2}</td>
</tr>
<tr>
<td>Computational complexity</td>
<td>$O(nT_p)$</td>
<td>$O(T_p)$</td>
<td>$O(nT_p)$</td>
<td>$O(T_p)$</td>
</tr>
</tbody>
</table>

5.1 Computational Complexity

The dominant cryptographic operation in the system is bilinear pairing. Let $T_p$ denote the time required to compute a pairing operation and $T_e$ denote the time required to compute a point multiplication in an elliptic curve. We conducted experiments to measure the average time required for generating a session key. We used the pairing-based cryptography (PBC) library with a 160-bit prime order $p$ on the Android 4.2 Jelly Bean platform with a 1.6-GHz dual-core CPU. The obtained result is 0.232 seconds. Let $n$ be the number of attributes averagely held by each user and $d$ be the threshold required to decrypt ciphertext. Here, we simplify $n$ to be 1 to give a comparison with other attribute-based protocols. While key agreement protocol is performed by Wa09 and Yo10 with an attribute, one pairing operation is computed. The computation of the proposed protocol as well as SHDFM is not affected by the number of the attributes each user has. Besides, the proposed protocol could be sped up by letting each user do some precomputations before the protocol session. In brief, the proposed protocol remains a favorable choice.

For revocation, the TS needs to encrypt key-update information with a computational complexity $O(dT_p+T_p)$. Each non-revoked user needs to decrypt key-update information with a computational complexity $O(dT_p)$. To update the new private key, each
non-revoked user requires a computational complexity of only $O(T)$. 

5.2 Communication Overhead

Let $m_{G_1}$ be an element of $G_1$ and $|m_{G_1}|$ be the size of an element of $G_1$. Let $m_{G_2}$ be an element of $G_2$ and $|m_{G_2}|$ be the size of $m_{G_2}$. While session key establishment is performed by Wa09 and Yo10 with an attribute, the size of exchanged messages is $O(|m_{G_1}| + |m_{G_2}|)$ and $O(|m_{G_1}|)$, respectively. Both Wa09 and Yo10 evaluate the access structure of each user to achieve authentication. In the proposed protocol and SHDFM, the size of exchanged messages is $O(|m_{G_1}|)$, which is not affected by the number of attributes a user owns. These observations thus indicate that the proposed protocol requires a lower communication cost than other schemes do. For the proposed revocation, the TS sends only encrypted key-update information with communication overhead $O(d |m_{G_1}| + |m_{G_2}|)$. 

5.3 Feature Comparison

In Table 3, we compare RAKA with other attribute-based authentication protocols in terms of key authentication, key exchange, user privacy, and user revocation requirements. All protocols provide key authentication requirements between parties, but none of the existing schemes offers user revocation solutions. In brief, the RAKA satisfies all requirements, including user revocation, and is thus suitable for MSN environments.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Key Authentication</th>
<th>Key Exchange</th>
<th>User Privacy</th>
<th>User Revocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS [7]</td>
<td>o</td>
<td>x</td>
<td>o</td>
<td>x</td>
</tr>
<tr>
<td>Wa09 [28]</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>SHDFM [1]</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Yo10 [19]</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>EPABKEM [15]</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
<tr>
<td>Proposed protocol</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

Table 3. Feature comparisons of attribute-based authentication protocols.

o: supported; x: not supported; Δ: partially supported

6. CONCLUSION

We propose a RAKA system for enabling secure communication between users with potential relationship in MSNs. To address problems pertaining to user revocation in MSN environments, we propose a user revocation to send short key-update information by a single broadcast to enable rekeying process with low communication overhead. Moreover, in the proposed system, multiple private key-update operations are aggregated into a single rekeying process to reduce computational overhead. We also facilitate updating the keys of users with user assistance in MSN environments. The security of the proposed protocol is analyzed, and its efficiency was demonstrated by conducting a performance analysis in terms of communication overhead and computational complexity.
REFERENCES


Tzu-Hsin Ho (何姿欣) received her M.S. and Ph.D. degrees in Computer Science from the National Chiao-Tung University, Hsinchu, Taiwan, in 1996 and 2015, respectively. Her research interests include cryptography, network security, and wireless communication.

Shiuhpyng Winston Shieh (謝績平) received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Maryland, College Park, respectively. He is a Distinguished Professor and the Director of Taiwan Information Security Center at National Chiao Tung University (NCTU). He has served as the Chair of the Department of Computer Science, NCTU, and President of Chinese Cryptology and Information Security Association (CCISA). Being actively involved in IEEE, he has served as EIC of IEEE Reliability, EIC of RS Newsletter, Reliability Society VP Tech, Editor of IEEE Transactions on Reliability and IEEE Transactions on Dependable and Secure Computing. Dr. Shieh has also served as ACM SIGSAC Awards Committee member, Associate Editor of ACM Transactions on Information and System Security, Journal of Computer Security, Journal of Information Science and Engineering, Journal of Computers, and the guest editor of IEEE Internet Computing, respectively. He has been on the organizing committees of many conferences, such as the founding Steering Committee Chair and Program Chair of ACM Symposium on Information, Computer and Communications Security (AsiaCCS), Program Chair of IEEE Conference on Security and Reliability. Along with Virgil Gligor of Carnegie Mellon University, he invented the first US patent in the intrusion detection, and has published around 200 technical papers, patents, and books. He is an IEEE Fellow and ACM Distinguished Scientist. His research interests include reliability and security hybrid mechanisms, malware behavior analysis, network and system security. Contact him at ssp@cs.nctu.edu.tw.

Chien-Chao Tseng (曾建超) received his B.S. degree in Industrial Engineering from National Tsing-Hua University, Hsinchu, Taiwan, in 1981; M.S. and Ph.D. degrees in Computer Science from the Southern Methodist University, Dallas, Texas, USA, in 1986 and 1989, respectively. He is currently a Professor in the Department of Computer Science at National Chiao-Tung University, Hsinchu, Taiwan. His research interests include wireless internet, heterogeneous networks, and mobile computing.