RAKA: Revocable Authenticated Key Agreement System for Mobile Social Networks

TZU-HSIN HO, SHIUHPYNG WINSTON SHIEH, AND CHIEN-CHAO TSENG
Department of Computer Science
National Chiao Tung University
Hsinchu 300, Taiwan

Mobile social networks (MSN) provide users with potential indirect relationships, such as friends of friends and similar transitive relationships. However, adversaries may be involved in the potential relationships, and therefore trustworthiness of the relationships must be ensured. That means 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 MSN. 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 MSN mobile devices using a single message broadcast. RAKA is proved to be resilient against malicious attacks.

Keywords: Authenticated key agreement, revocation, Mobile social networks (MSN), rekeying process, secure communication.

1. INTRODUCTION

Mobile social networks (MSN) enable social networking in mobile phones, bringing much excitement to smartphone and tablet markets. In addition to essential capability offered by Internet-based social networks, mobile social networks enable mobile users with some potential relationship to extend their online social networks through opportunistic meets in the physical vicinity. The 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 contents. By the potential relationship, two or more mobile users in the same vicinity can interact with each other to share specific information [12] [30].

Since users can interact with others through potential relationships in the MSN environment, two security problems may arise [23]. First, users may interact with malicious adversaries who disguise to establish potential relationships. To prevent this from happening, the relationship validation should be assured before the communication is started. Second, a user may only intend to selectively disclose his/her private information to specific users. To resolve the two security problems simultaneously, 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.

However, applying authenticated key agreement protocol to MSN environments

introduces some serious challenges to relationship revocation, and at least three issues 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. To revoke malicious adversaries from the potential relationship becomes a major challenge. Second, rekeying information delivery must be atomic. Users with wireless connections may experience unpredictable disconnection to the central key server due to 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, inconsistent with other users. Third, the communication and computation overhead of a rekeying process should be low due to the limited resource and capability of mobile devices. All these issues will be further addressed later in this paper.

Various security protocols have been proposed [10, 17, 20, 25]. In the route-based authentication [20] for data delivery, entity authentication is 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 to 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 by 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 [9] [2] have been proposed allowing users to revoke external access to data using revocation list. However, a user may only interact with others using potential relationships. In this case, the user may not know the identity of his communicating party, let alone know whether the party's identity is in the revocation list or not. Therefore, these methods are not suitable for MSN. Using authority for key update is another choice [4, 16, 18, 26, 31]. However, users need to contact the authority periodically [4] [26] to set expiration time for users' private keys. Consequently, revocation with immediate revoking capability has been provided [16]. This method may need multiple secure channels for rekeying. This is because the total number of secure channels for sending key update information may increase with respect to the number of the non-revoked users. To reduce the overhead of establishing multiple secure channels, a method was proposed [31] to revoke a user via 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 MSN. In the system, an implicitly authenticated key agreement protocol is proposed which keeps user privacy at the same time. For user revocation, the efficient rekeying process is designed for 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 computation overhead of constrained mobile devices, RAKA aggregates multiple successive key update operations into a single rekeying process. Furthermore, we provide 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 aserver. 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 our system.

2.1 Bilinear Pairing

Bilinear pairing has been widely used to build up various security schemes [5, 27]. Let $\mathbb{G}_1$ be a cyclic additive group with a generator P, $\mathbb{G}_2$ be a cyclic multiplicative group, and $\mathbb{G}_1$ and $\mathbb{G}_2$ have the same prime order p. Besides, let $\mathbb{Z}_p = \{0,1,\cdots,p-1\}$ and $\mathbb{Z}_p^* = \{1,2,\cdots,p-1\}$. A function $\hat{e}:\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is called an admissible bilinear map if it satisfies the following properties:

1. Bilinearity: $\forall R,Q \in \mathbb{G}_1$ and $\forall a,b \in \mathbb{Z}_p^*$
   $\hat{e}(aR,bQ) = \hat{e}(aR,Q)^b = \hat{e}(R,bQ)^a = \hat{e}(R,Q)^{ab}$.
2. Non-degeneracy: $\hat{e}(P,P) \neq 1$.
3. Computability: There exists an efficient algorithm to compute $\hat{e}(R,Q)$ for all $R,Q \in \mathbb{G}_1$.

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 following:

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, the private key of a user is associated with a set of attributes. The data are encrypted associated with a set of attributes. A user can decrypt a ciphertext if and only if the number of the overlapping attributes between attributes with the user's private key

and with the ciphertext is over a threshold. Let d be the threshold of overlapping attribute, ω' be an attribute set associated with a ciphertext, and ω 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

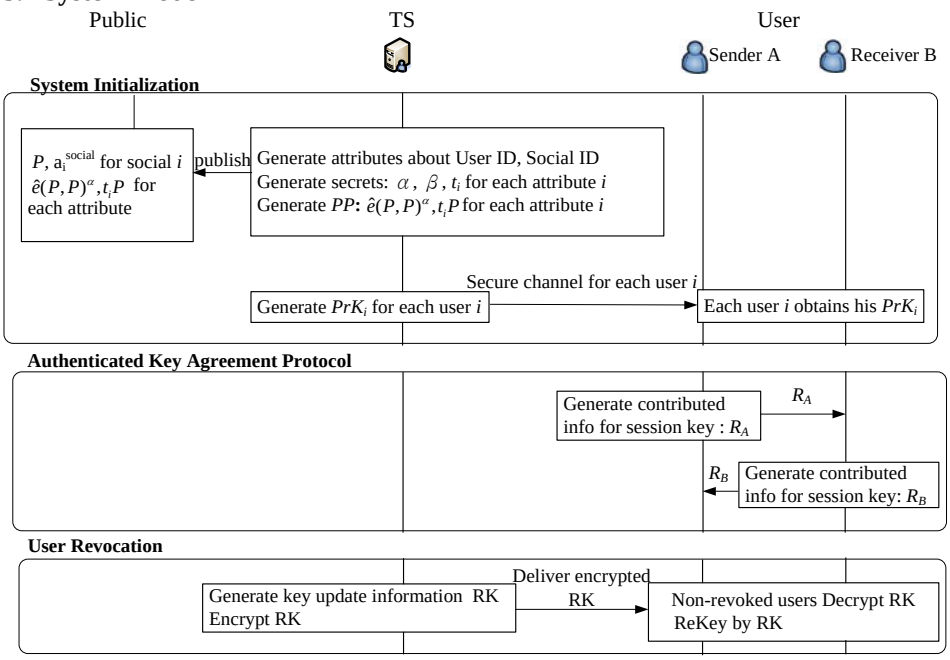

Fig 1. The revocable authenticated key agreement system (RAKA)

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

1. *Trusted Server (TS). TS* is the authority managing the attribute set, issuing and handling keys. It generates public parameters, and private keys for users. *TS* is also responsible for generating key update information when user revocation occurs. It is the only party 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 *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, *TS* sets the potential relationships of each user, decides the 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. *TS* also 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, *TS* generates key update information and protects it by encryption. Then, the encrypted key update information can be broadcast in a secure fashion. Only the non-revoked users can decrypt the key update information, and use it to update their private keys. The three major functions, including system initialization, authenticated key agreement protocol and user revocation will be described in the later 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 identities of involved users should be hidden from others.
3. *Resistance to passive attack*: A passive adversary who tries to eavesdrop the 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.
5. *Forward secrecy*: If an adversary breaches the two communicating parties' private keys, the past session keys established between the two users should remain secure.

3.3 System Initialization

TS first determines parameter *d* as threshold for decryption, and generates public parameters and private keys for users. Let $\mathbb{G}_1$ be subgroup of the basing elliptic curve group $\mathbb{E}$. Let the order of $\mathbb{G}_1$ be *p* and the order of $\mathbb{E}$ be $\#\mathbb{E}$. Hence, $h = \frac{\#\mathbb{E}}{p}$ is the cofactor of the group $\mathbb{G}_1$. Two public functions are used in the system, including

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

The details are introduced below.

1) Parameter Generation: In the system, each potential relationship is assigned a unique

attribute, called social ID, and each user is also assigned a unique attribute, called user ID. Here, the attribute of social *i* is denoted as a_i^{social}, and the user attribute of user *i* is denoted as a_i^{user}. In addition, *TS* chooses *d-1* attributes to form a dummy attribute set, denoted as $\mathcal{DA}$, which is a common attribute set known by all users. All dummy attributes can't be re-used to present other attributes. For each attribute *i*, *TS* chooses a random secret $t_i \in \mathbb{Z}_p$. *TS* also chooses two master secrets α, β randomly from $\mathbb{Z}_p$, where α is the master secret for encrypting key update information, and β is the master secret for generating session key. A public parameter set *PP* denoted as

$$PP = \left\langle \begin{array}{l} \hat{e}(P,P)^{\alpha}; \\ t_i P \text{ for all attribute } i \end{array} \right\rangle$$

is published to all users.

2) Private Key Generation: TS generates the 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 ω_i denote the attribute set $G_i \cup \{a_i^{user}\} \cup \mathcal{DA}$. For each user *i*, *TS* randomly chooses a personal polynomial $f_i(\cdot)$ of *d-1* degree and with $f_i(0) = \alpha$. The private key $PrK_i = \langle D1, D2 \rangle$ for user *i* is

$$D1 = \left\langle D1_k^i = \frac{f_i(k)}{t_k} P \text{ for } k \in \omega_i \right\rangle,$$

$$D2 = \left\langle D2_k^i = \frac{\beta - f_i(k)}{t_k} P \text{ for } k \in G_i \right\rangle.$$

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

3.4 Proposed Authenticated Key Agreement Protocol

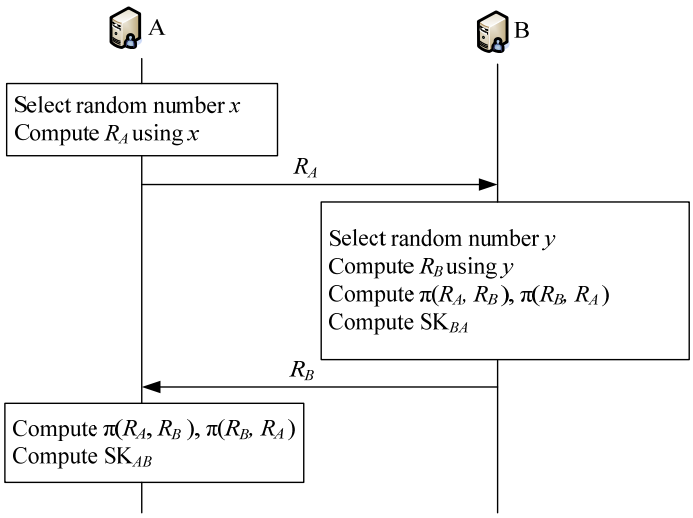

Fig. 2. Proposed authenticated key agreement protocol

Let user *A* and user *B* have certain potential relationship representative by the attribute a_j^{social}. For simplicity, notation $k = a_j^{social}$ will be used in the following sections. Then, user *A* owns the private key component corresponding to attribute k, denoted as $\left(D1_k^A, D2_k^A\right)$, and User *B* holds the private key component corresponding to attribute k, denoted as $\left(D1_k^B, D2_k^B\right)$. And let $T_k = t_k P$ be public parameter corresponding to attribute k. Only the 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 clearly 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 received R_A to compute $s_A = \pi(R_A, R_B), s_B = \pi(R_B, R_A)$, and the common session key sk_{BA} as

$$\begin{aligned} sk_{BA} &= \hat{e}\left((y+s_B)h(D1_k^B + D2_k^B), s_A T_k + R_A\right) \\ &= \hat{e}\left(P,P\right)^{(y+s_B)h\beta(x+s_A)} \end{aligned} \tag{1}$$

3. User *A* uses received R_B to compute $s_A = \pi(R_A, R_B), s_B = \pi(R_B, R_A)$, and the common session key sk_{AB} as

$$\begin{aligned} sk_{AB} &= \hat{e}\left((x+s_A)h(D1_k^A + D2_k^A), s_B T_k + R_B\right) \\ &= \hat{e}\left(P,P\right)^{(x+s_A)h\beta(y+s_B)} \end{aligned} \tag{2}$$

3.5 User Revocation

3.5.1 Basic Revocation Method

To revoke user from a potential relationship representative by the attribute k, we perform two basic operations: (i) update the secret, public parameter related to the attribute k; (ii) update the private key component corresponding to the attribute k of each non-revoked user. Operation (i) is performed by *TS*, and operation (ii) is delegated to users. *TS* generates a short key update information and broadcasts it. According to the key update information, non-revoked users can update their private keys by themselves. The detail of operations are as follows: *TS* chooses a random number $t_k^{'} \in \mathbb{Z}_p$ 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 *TS*, $RK_k = \left(t_k \cdot t_k^{'-1}\right)$.

Moreover, to prevent key update information intercepted by revoked/unauthorized users during delivery, *TS* encrypts RK_k before sends it out. There are plenty of selectively broadcast schemes [6, 8, 14, 24] available such that all but the revoked/unauthorized users are able to decrypt RK_k. In our revocation solution, we use a list to selectively include non-revoked users to access key update information based on Fuzzy-IBE. Setting all non-revoked users into the list, *TS* assigns the set ω' with attributes a_i^{user} of each user i in the list, and encrypts the key update information RK_k with ω'. The encrypted RK_k, denoted as $E(RK_k)$, is computed as below

$$E\left(RK_k\right) = \left\langle \begin{array}{l} \omega' = \{a_i^{user}\text{: all user } i \text{ in the list}\}; \\ E1 = RK_k \left[\hat{e}\left(P,P\right)^{\alpha}\right]^r; \\ E2_j = r(t_j P) \text{ for all } j \in \omega' \cup \mathcal{DA} \end{array} \right\rangle, \text{ where } r \in_R \mathbb{Z}_p.$$

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

$$D\left(E\left(RK_k\right)\right) = \frac{E1}{\prod_{j\in\omega'\cup\mathcal{DA}} \hat{e}(D1_j, E2_j)^{\varepsilon_j}} = RK_k, \text{ where}$$

$\varepsilon_j = \prod_{j\in\omega'\cup\mathcal{DA};\, l\neq j} \dfrac{0-l}{j-l}$ is a Lagrangian coefficient that satisfies $\sum_{j\in\omega'\cup\mathcal{DA}} \varepsilon_j f_i(j) = f_i(0)$.

It indicates the 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:

$$D1_k^i = \left(t_k \cdot t_k^{'-1}\right) \cdot f_i(k) \cdot t_k^{-1} \cdot P = \frac{f_i(k)}{t_k^{'}} P,$$

$$D2_k^i = \left(t_k \cdot t_k^{'-1}\right) \cdot \left(\beta - f_i(k)\right) \cdot t_k^{-1} \cdot P = \frac{\beta - f_i(k)}{t_k^{'}} P.$$

3.5.2 Advanced Revocation Method

To reduce the computation overhead of users for updating private key repeatedly, we aggregate multiple successive key update operations into a single process. *TS* associates version number ver_i with encrypted key update information $E(RK_k)$, denoted as $E(RK_k)_i$, and broadcasts $E(RK_k)_i$ out. Each user can record all versions of encrypted key update information he received. The process for users updating key from version *i* to version *j* is performed by the following steps:

1. Decrypt the $E(RK_k)_i$. If this user is a revoked one, the decryption will fail. Otherwise, the user can decrypt $E(RK_k)_i$ to obtain RK_k and go 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 $j+1=i$, succeeds and then exits. Otherwise, go back to step 1.

Moreover, with the recorded encrypted key update information, the one can assist others who have older private key 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. The user who initiates the algorithm is called initiator. The other party is called 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 the initiator has higher version than that of the responder, he sends sufficient successive encrypted key update information to the responder. Then the responder can update his private key using function *ReKey* by himself. If the initiator has lower version, he obtains encrypted key update information from responder. And consequently, he can complete his private key update using function *ReKey*. Since key update information are well protected by encryption, only the non-revoked users can decrypt the key update information. It means that only non-revoked users can use key update information to update his private key to newly one.

Table 1. Algorithm *Aggregative_Key_Update*

```
// Initiator A requests version info to responder B
// Compare the version of A with the version of B
If A.ver > B.ver then
        send E(RK_k)_{B.ver+1} ~ E(RK_k)_{A.ver}  to B
else
        receive E(RK_k)_{A.ver+1} ~ E(RK_k)_{B.ver}  from B
        ReKey(k, A.ver+1,B. ver)
end if

Function ReKey(k, ver1, ver2)
for i=ver1 to ver2 do
        if decrypt E(RK_k)_i then
```

```
            D1_k ← RK_k · D1_k ; D2_k ← RK_k · D2_k
        else
            Caller is not a valid member
    end for
    end Function
```

4. SECURITY ANALYSIS

In this section, we will prove that the proposed authenticated key agreement protocol is scure.

Based on the Bellare and Rogaway model [3], we construct the following security model. In the model, adversary completely controls over network communication. Let oracle $\prod_{A,B}^{s}$ denote the behavior of the user *A* to carry out a protocol session *s* to communicate with the user *B*. For each $\prod_{A,B}^{s}$, a record, denoted *view*, is kept to track its input/output messages. Let the adversary $\mathcal{A}$ interact with the participants via oracle accesses. The oracle queries are listed in the following:

1. $Send\left(\prod_{A,B}^{s}, X\right)$. The query models active attacks, where the adversary can send a message X to $\prod_{A,B}^{s}$. 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. $\mathrm{Re}veal\left(\prod_{A,B}^{s}\right)$. The query models the exposure of the session key of protocol session s.
3. $Corrupt(A)$. The query models the exposure of the private key of user *A*.
4. $\pi(X,Y)$. The query generates a hash value from the two messages X and Y.
5. $Test\left(\prod_{A,B}^{s}\right)$. The query models the adversary's advantage of guessing session key. The adversary $\mathcal{A}$ can make a query to $\prod_{A,B}^{s}$ for distinguishing a real session key from a random key. The oracle $\prod_{A,B}^{s}$ gives $\mathcal{A}$ the real session key or a random key according to a random coin bit $b \in \{0,1\}$. $\mathcal{A}$ has to guess the value of b.

After adversary $\mathcal{A}$ has made all the queries it wants to ask, $\mathcal{A}$ executes *Test* to query a fresh oracle which has accepted and have not been queried *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. Adversary $\mathcal{A}$'s advantage for guess is defined as

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

where k is 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 $\prod_{A,B}^{s}$ and $\prod_{B,A}^{t}$ have matching conversations, then both oracles will always accept and hold the same session key which is distributed uniformly over the key space.
2. $Adv^{\mathcal{A}}(k)$ is negligible.

Theorem 1 *The proposed protocol is a secure authenticated key agreement protocol assuming π is an independent random oracle.*

Proof. Condition 1 holds as follows. We consider the session key generation between oracles $\prod_{A,B}^{s}$ and $\prod_{B,A}^{t}$ for the relationship *k*. Let user *A* hold the private key components $\left(D1_k^A, D2_k^A\right)$. Upon receiving R_B from *Send* query, $\prod_{A,B}^{s}$ chooses $x \in_R \mathbb{Z}_p$ and generates $R_A = xT_k$. 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 $\pi(R_B, R_A)$ query, respectly. $\prod_{A,B}^{s}$ produces the shared secret sk_{AB} as following.

$$\begin{aligned}
sk_{AB} &= \hat{e}\left(\left(x + s_A\right) h\left(D1_k^A + D2_k^A\right), s_B T_k + R_B\right) \\
&= \hat{e}\left((x + s_A) h\left(\frac{p_A(k)}{t_k} P + \frac{\beta - p_A(k)}{t_k} P \right), s_B T_k + y T_k \right) \\
&= \hat{e}\left((x + s_A) h\left(\frac{p_A(k)}{t_k} P + \frac{\beta - p_A(k)}{t_k} P \right), (s_B + y) t_k P \right) \\
&= \hat{e}\left(P, P\right)^{(x+s_A) h \frac{\beta}{t_k} (s_B + y) t_k} \\
&= \hat{e}\left(P, P\right)^{(x+s_A) h \beta (s_B + y)}
\end{aligned}$$

Let user *B* hold private key components $\left(D1_k^B, D2_k^B\right)$. Upon receiving R_A from *Send* query, $\prod_{B,A}^{t}$ chooses $y \in_R \mathbb{Z}_p$ and generates $R_B = yT_k$. 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 $\pi(R_B, R_A)$

query, respectly. $\prod_{B,A}^{t}$ produces the shared secret sk_{BA} as following.

$$\begin{aligned} sk_{BA} &= \hat{e}\left((y+s_B)h\left(D1_k^B + D2_k^B\right), s_A T_k + R_A\right) \\ &= \hat{e}\left((y+s_B)h\left(\frac{p_B(k)}{t_k}P + \frac{\beta - p_B(k)}{t_k}P\right), s_A T_k + xT_k\right) \\ &= \hat{e}\left((y+s_B)h\left(\frac{p_B(k)}{t_k}P + \frac{\beta - p_B(k)}{t_k}P\right), (s_A + x)t_k P\right) \\ &= \hat{e}\left(P,P\right)^{(y+s_B)h\frac{\beta}{t_k}(s_A+x)t_k} \\ &= \hat{e}\left(P,P\right)^{(x+s_A)h\beta(y+s_B)} \end{aligned}$$

Since sk_{AB} is identical to sk_{BA}, both parties generate the same session key under random oracle π. 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 $\prod_{A,B}^{s}$ and *y* is generated by $\prod_{B,A}^{t}$. The product of these elements is also a random element in $\mathbb{Z}_p$. Since the shared secret's exponent part is random, the shared secret is distributed uniformly over $\mathbb{G}_2$.

Condition 2 holds under the DBDH assumption. By way of contradiction, suppose that $Adv^{\mathcal{A}}(k)$ is non-negligible. We construct an adversary $\mathcal{C}$ from $\mathcal{A}$ to solve DBDH problem with non-negligible advantage. $\mathcal{C}$ takes as input $< P, uP, vP, wP, \hat{e}(P,P)^{\tilde{a}} >$, where $\tilde{a}$ can be *uvw* or a random number *z* in $\mathbb{Z}_p$. $\mathcal{C}$ tries to distinguish $\hat{e}(P,P)^{uvw}$ from $\hat{e}(P,P)^{z}$. $\mathcal{C}$ chooses *A* and *B* from a set of entities and sessions s and t randomly. $\mathcal{C}$ generates all entities' private keys at random excluding user *A*'s. $\mathcal{C}$ selects $c_1, c_2 \in_R \mathbb{Z}_p$ and creates the private key $D1_k^A = c_1 uP, D2_k^A = c_2 vP$ for user *A*. $\mathcal{C}$ answers all $\mathcal{A}$'s queries itself. $\mathcal{C}$ answers all *Corrupt* queries as specified by the protocol execpt the *Corrupt* queries from $\mathcal{A}$ asking user *A* or *B*. If these queries occure, $\mathcal{C}$ aborts. $\mathcal{C}$ also answers all *Send* queries including queries of $\prod_{A,B}^{s}$ and $\prod_{B,A}^{t}$ as specified by the protocol. When $\mathcal{A}$ asks the *Send* query about the first flow of $\prod_{B,A}^{t}$, $\mathcal{C}$ answers $R_B = c_4 c_3 wP$, where $c_3, c_4 \in_R \mathbb{Z}_p$. $\pi(R_A, R_B)$ and $\pi(R_B, R_A)$ can be assumed to take c_5 and c_6 uniformly from $\mathbb{Z}_p$, respectively. Then, $\mathcal{C}$ sets $\hat{e}(P,P)^{\tilde{d}} = \hat{e}(P,P)^{(x+c_5)h(c_1u+c_2v)w(c_6c_3+c_4c_3)}$, where *x* is a random number in $\mathbb{Z}_p$, and sends $\hat{e}(P,P)^{\tilde{d}}$ to $\mathcal{A}$. If $\mathcal{A}$ determines $\hat{e}(P,P)^{\tilde{d}}$

to be a session key, then $\mathcal{C}$ guesses $\hat{e}(P,P)^{\tilde{a}} >$ coming from $\hat{e}(P,P)^{uvw}$. Otherwise, $\mathcal{C}$ guesses $\hat{e}(P,P)^{\tilde{a}} >$ coming from $\hat{e}(P,P)^{z}$. Thus, algorithm $\mathcal{C}$ can distinguish $\hat{e}(P,P)^{uvw}$ from $\hat{e}(P,P)^{z}$ with non-negligible advantage, which is a contradiction to DBDH Assumption.

5. PERFORMANCE ANALYSIS

In this section, the proposed protocol will be compared with some existing protocols, including Wa09 [28], SHDFM [1] and Yo10 [19], in terms of the communication overhead and the computational complexity. Table 2 gives the comparison results.

Table 2. Performance comparisions of the unerlying cryptographies

	Wa09 [28]	SHDFM [1]	Yo10 [19]	Proposed protocol
Communication overhead	$\mathcal{O}(n\,\|m_{G_1}\|+\|m_{G_2}\|)$	$\mathcal{O}(\|m_{G_1}\|)$	$\mathcal{O}(n\,\|m_{G_1}\|)$	$\mathcal{O}(\|m_{G_1}\|)$
Computational complexity	$\mathcal{O}(nT_p)$	$\mathcal{O}(T_p)$	$\mathcal{O}(nT_p)$	$\mathcal{O}(T_p)$

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 to generate a session key. We used the pairing-based cryptography (PBC) library with 160-bit prime order p on Android 4.2 Jelly Bean platform with an 1.6 GHz Dual Core CPU. The obtained result is 0.223 seconds. Let n be the number of attributes averagely held by each user and d be the threshold required to decrypt a 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 pre-computation before the protocol session. In brief, the proposed protocol remains the better choice.

For revocation, trusted server needs to encrypt key update information with computational complexity $\mathcal{O}(dT_e+T_p)$. Each non-revoked user needs to decrypt key update information with computational complexity $\mathcal{O}(dT_p)$. To update the new private key, each non-revoked user requires only computational complexity $\mathcal{O}(T_e)$.

5.2 Communication Overhead

Let m_{G_1} be an element of $\mathbb{G}_1$, and $|m_{G_1}|$ be the size of an element of $\mathbb{G}_1$. Let

m_{G_2} be an element of $\mathbb{G}_2$, 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 $\mathcal{O}(|m_{G_1}|+|m_{G_2}|)$ and $\mathcal{O}(|m_{G_1}|)$, respectively. Both Wa09 and Yo10 consider the access structure of each user to achieve authentication. In the proposed protocol and SHDFM, the size of exchanged messages is $\mathcal{O}(|m_{G_1}|)$, which is not affected by the number of attributes a user owns. As shown, the proposed protocol needs lower communication cost than other schemes. For the prosposed revocation, trusted server sends only encrypted key update information with communication overhead $\mathcal{O}(d\,|m_{G_1}|+|m_{G_2}|)$.

5.3 Feature Comparison

Table 3. Feature comparsions of the attribute-based authentication prtocotocols

	ABS[7]	Wa09 [28]	SHDFM [1]	Yo10 [19]	EPABKEM [15]	Proposed protocol
Key Auth	o	o	o	o	o	o
Key Exchange	x	o	o	o	o	o
User Privacy	o	o	x	o	o	o
User Revocation	x	Δ	x	x	x	o

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

In Table 3, we compare RAKA with other attribute-based authentication protocols with respect to the requirements in key authentication, key exchange, user privacy, and user revocation. All protocols provide key authentication between parties, but none of the existing schemes offers user revocation. In brief, RAKA provides all requirements including user revocation to fit into the mobile social environment.

6. CONCLUSION

In this paper, we design a revocable authenticated key agreement system to establish secure communication between users with potential relationship in the mobile social networks. To address the problems in regard to user revocation in MSN environment, we propose a user revocation to send short key update information by a single broadcast to enable rekeying process with low communication overhead. Moreover, we aggregate multiple private key update operations into a rekeying process to reduce computation overhead. And we facilitate the key update of user with user's assistance in mobile social environments. The security of the proposed protocol is analyzed, and we also provide performance analysis in terms of communication overhead and computational complexity to show the efficiency of the proposed protocol.

REFERENCES

1. Giuseppe Ateniese, Jonathan Kirsch, and Marina Blanton, "Secret handshakes with

dynamic and fuzzy matching," in *NDSS*, pages –1–1, 2007.
2. Nuttapong Attrapadung and Hideki Imai, "Conjunctive broadcast and attribute-based encryption," in *International Conference Palo Alto on Pairing-Based Cryptography*, pages 248–265, 2009.
3. Mihir Bellare and Phillip Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology*, volume 773 of LNCS, pages 232–249. Springer Berlin Heidelberg, 1994.
4. John Bethencourt, Amit Sahai, and Brent Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.
5. Dan Boneh and Matt Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology: Proceedings of Crypto*, volume 2139 of LNCS, pages 213–229. Springer-Verlag, 2001.
6. Dan Boneh, Craig Gentry, and Brent Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer Berlin Heidelberg, 2005.
7. Dan Cao, Baokang Zhao, Xiaofeng Wang, Jinshu Su, and Yijiao Chen, "Authenticating with attributes in online social networks," in *International Conference on Network-Based Information Systems*, volume 6487, pages 607–611, 2011.
8. Ling Cheung, Joseph A. Cooley, Roger Khazan, and Calvin Newport, "Collusion-resistant group key management using attribute-based encryption," 2007.
9. Ling Cheung and Calvin Newport, "Provably secure ciphertext policy abe," in *ACM conference on Computer and communications security*, pages 456–465, 2007.
10. Wei Dong, Vacha Dave, Lili Qiu, and Yin Zhang, "Secure friend discovery in mobile social networks," in *IEEE Infocom*, pages 1647–1655, 2011.
11. Liming Fang and Jinyue Xia, "Full security: Fuzzy identity based encryption," *IACR Cryptology ePrint Archive*, 2008.
12. Jialu Fan, Jiming Chen, Yuan Du, Wei Gao, Jie Wu, and Youxian Sun, "Geocommunity-based broadcasting for data dissemination in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, 24(4):734–743, April 2013.
13. Steven D. Galbraith, Keith Harrison, and David Soldera, "Implementing the tate pairing," in *Algorithmic Number Theory Symposium*, volume 2369, pages 324–337, Mar 2002.
14. MichaelT. Goodrich, JonathanZ. Sun, and Roberto Tamassia, "Efficient tree-based revocation in groups of low-state devices," in *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 511–527. Springer Berlin Heidelberg, 2004.
15. M. Choudary Gorantla, Colin Boyd, and Juan Manuel Gonz a' lez Nieto, "Attribute-based authenticated key exchange," in *Proceedings of Australasian Conference on Information Security and Privacy*, ACISP, pages 300–317, 2010.
16. Luan Ibraimi, Milan Petkovic, Svetla Nikova, Pieter Hartel, and Willem Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Information Security Applications*, pages 309–323. Springer-Verlag, 2009.
17. Anna-Kaisa Pietil ä inen, Earl Oliver, and Jason LeBrun, "Mobiclique: Middleware

for mobile social networking," in *ACM workshop on Online social networks*, pages 49–54, 2009.
18. Sonia Jahid, Prateek Mittal, and Nikita Borisov, "Easier: encryption-based access control in social networks with efficient revocation," in *ACM Symposium on Information, Computer and Communications Security*, pages 411–415, 2011.
19. Yoneyama Kazuki, "Strongly secure two-pass attribute-based authenticated key exchange," in *Pairing*, volume 6487, pages 147–166. Springer, 2010.
20. Xiaohui Liang, Xu Li, Tom H. Luan, Rongxing Lu, Xiaodong Lin, and Xuemin (Sherman) Shen, "Morality-driven data forwarding with privacy preservation in mobile social networks," *IEEE Transactions on Vehicular Technology*, PP(99), 2012.
21. Xiaohui Liang, Rongxing Lu, Xiaodong Lin, and Xuemin (Sherman) Shen, "An efficient and secure user revocation scheme in mobile social networks," in *IEEE Globecom*, pages 1–5, 2011.
22. Justin Manweiler, Ryan Scudellari, and Landon P. Cox, "Smile: Encounter-based trust for mobile social services," in *ACM conference on Computer and communications security*, pages 246–255, 2009.
23. A. Mohaien, D.F. Kune, E.Y. Vasserman, Myungsun Kim, and Yongdae Kim, "Secure encounter-based mobile social networks: Requirements, designs, and tradeoffs," *IEEE Transactions on Dependable and Secure Computing*, 10(6):380–393, Nov 2013.
24. Dalit Naor, Moni Naor, and Jeff Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer Berlin Heidelberg, 2001.
25. Omer Nawaz, Christian Gehrmann, and Markus Fiedler, "Secure mobile social networks using usim in a closed environment," in *International Conference for Internet Technology And Secured Transactions*, pages 439–446, Dec 2012.
26. Matthew Pirretti, Patrick Traynor, Patrick Traynor, and Brent Waters, "Secure attribute-based systems," in *ACM Conference on Computer and Communications Security*, pages 99–112, 2006.
27. Amit Sahai and Brent Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology: Proceedings of Eurocrypt 2005*, volume 3494 of LNCS, pages 457–473. Springer-Verlag, 2005.
28. Hao Wang, Qiuliang Xu, and Xiu Fu, "Revocable attribute-based key agreement protocol without random oracles," *Journal of Networks*, 4(8):787–794, 2009.
29. Yongge Wang, "Efficient identity-based and authenticated key agreement protocol," Cryptology ePrint Archive, Report 2005/108, 2005. http://eprint.iacr.org/.
30. Zhimin Yang, Boying Zhang, Jiangpeng Dai, Adam C. Champion, Dong Xuan, and Du Li, "E-smalltalker: A distributed mobile system for social networking in physical proximity," in *International Conference on Distributed Computing Systems (ICDCS)*, pages 468–477, 2010.
31. Shucheng Yu, Kui Ren, and Wenjing Lou, "Fdac: Toward fine-grained distributed data access control in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, 22(4):673–686, 2011.