Distributed Multi-authority Attribute-based Encryption Scheme for Friend Discovery in Mobile Social Networks

Wenbo Wang¹, Fang Qi¹, Xiaoqiang Wu², and Zhe Tang¹ *

¹ School of Information Science and Engineering, Central South University, Changsha, China, 410083
wb.wang@csu.edu.cn, qi.fangfang@gmail.com, tz@csu.edu.cn
² Institute of Software, Chinese Academy of Sciences, Beijing, China, 100190
xiaoqiang2014@iscas.ac.cn

Abstract
In recent years, the rapid expansion of the capability of portable devices, cloud servers and cellular network technologies is the wind beneath the wing of mobile social networks. Compared to traditional web-based online social networks, the mobile social networks can assist users to easily discover and make new social interaction with others. A challenging task is to protect the privacy of the users’ profiles and communications. Existing works are mainly based on traditional cryptographic methods, such as homomorphic and group signatures, which are very computationally costly. In this paper, we propose a novel distributed multi-authority attribute-based encryption scheme to efficiently achieve privacy-preserving without additional special signatures. In addition, the proposed scheme can achieve fine-grained and flexible access control. Detailed analysis demonstrates the effectiveness and practicability of our scheme.

Keywords: Multi-authority, Attribute-based Encryption, Privacy Preserving, Access Control, Profile Matching

1 Introduction

A boom in mobile hand-held devices greatly enriches the social networking application [1]. Many social networking services are available on the mobile devices (e.g., WeChat, QQ, MocoSpace, etc.). According to eMarketer [2], they estimate that the number of US smartphone users will reach 192.4 million by 2016 and 2.28 billion worldwide [3]. Friend discovery and communication are two important basic steps of social networks. When people take part in social networks, they usually begin by creating a profile, then interact with others. The personal profile usually contains a large amount information, such as hobbies, age, education degree, etc. Profile matching is a common and helpful method to make new friend with mutual interests or experience. Unfortunately, a series of unaddressed security and privacy problems dramatically impede its practicability and popularity [4].

*Corresponding author
In recent years, many private matching schemes have been proposed to solve this problem. Among these schemes, some protect user’s privacy based on trusted third party (TTP) 
[5, 6, 7, 8], the other is TTP-free [9, 10, 1]. Although, this kind of approaches can achieve profile matching without the support of TTP, they have some disadvantages. The reliance on public-key cryptosystem and homomorphic encryption [11, 12, 7, 8] requires multiple rounds of interaction which causes high communication and computation overhead. Moreover, matched and unmatched users are all involved in the expensive computation and learn the matching result. Li et al. [9] propose a private matching scheme based on the common interests, which is not fine-grained. Zhang et al. [8] present a fine-grained private matching scheme but fail in considering the priority related to every attribute and they employ the homomorphic encryption which is resource consuming on mobile devices. Qi et al. [10] employ an asymmetric-scalar-production based on kNN query, but the presentation of interests is too single to get an accurate result. Moreover, the widely used technique of group signature [13][14] always costs huge volume of computational resources on users’ hand-held devices, and the access control based on the key-policy attribute-based encryption [15] is not efficient enough. In addition, if any server or TTP is compromised, the confidentiality of the stored data may be compromised, too. Therefore, considering the powerful computational ability of the TTP and cloud server, the main point of our work is to design an efficient privacy-preserving and fine-grained friend discovery system based on the combination of TTP and cloud server.

In this paper, we propose an efficient distributed multi-authority attribute-based encryption scheme, which can achieve privacy preserving and fine-grained access control. By using ciphertext-policy attribute-based encryption (CP-ABE) [16], the encrypted information can kept confidential even if the storage server is not fully-trusted and users can design their own access policy. Hence, the fine-grained access control can be achieved efficiently. By employing the powerful storage and computational ability of cloud server, the storage and computation overhead of the client can be greatly reduced. The multi-authorities are designed to be distributed, which can significantly relieves the users’ trust on a single authority and is secure against collusion attack as well as chosen-plaintext attack. The main contributions are outlined as follows.

- A multi-authority attribute-based encryption scheme is proposed for fine-grained multi-level access control in cloud friend discovery system. Users can design their own access policy to find the potential friends, which is user friendly.

- User’s identity and personal profile are encrypted under the access policy specified by the user himself and outsourced to the cloud server, the client is lightweight.

- The distributed multi-authority model in friend discovery cloud computing system also reduces the risk of a single central authority being compromised for potential privacy leakage.

- Formal security proof and simulation evaluation demonstrate that our scheme is secure against chosen-plaintext attack and collusion attack in the standard model.

The remainder of this paper is organized as follows. Preliminaries are introduced in Section 2. The system architecture and models are presented in Section 3. We propose our scheme in Section 4, followed by the formal security proof and performance evaluations respectively in Section 5 and 6. Finally, we conclude our work.
2 System Architecture and Models

2.1 System Architecture

The architecture of the friend discovery system mainly contains the following components: the personal profile which is outsourced in the encrypted form into the cloud by the initiator; the cloud server that stores huge volumes of users’ personal profiles and performs the efficient attribute matching process to realize the multi-level fine-grained access control for privacy preserving; the responders who will attend the profile matching and may be the potential friend. Moreover, in our system, there are also $D$ central authorities ($CA_1, CA_2, \ldots, CA_D$) and $K$ attribute authorities ($AA_1, \ldots, AA_K$). Each responder has a global identifier $gid \in GID$, where the $GID$ is the identity set of all registered users. Responders get the keys concerning their unique $gid$ from $CA_i$ ($i \in 1, 2, \ldots, D$). Each attribute authority $AA_k$ ($k \in 1, 2, \ldots, K$) manages a set of attributes $U_k (U_i \cap U_j = \emptyset \land U = \bigcup_{k=1}^{K} U_k)$ ($i, j \in \{1, 2, \ldots, K\} \land i \neq j$). Each authorized responder with attribute set $AS_{gid}$ will obtain their attribute secret keys from the corresponding $AA_k$s. We assume that all the authorities are run by different organizations and governed by the government. The multiple authority setting greatly relieves the users’ trust on a single $CA$ or $AA$, so it is unlikely for all the authorities to collude (or to be compromised) to derive the secret keys. Figure 1 illustrates the architecture of the cloud friend discovery system.

![Figure 1: Architecture of Cloud Friend Discovery System](image)

2.2 Security Model

The formal security model of our proposed scheme is defined by the following game runs between a challenger $C$ and an adversary $A$.

**Key query phase 1:** The adversary $A$ tries to query the following random oracles.

$O^{CAKeyGen(gid,i)}$: $A$ queries with $gid$ and $i^*$, where $gid$ is the global identity. It returns the corresponding responder-identity-key ($rpsk^0_{gid,i}$, $rpsk^1_{gid,i}$) and $rppk_{gid,i}$.

$O^{AAKeyGen(att,rpapk_{gid,i},k)}$: $A$ queries with $rpapk_{gid,i}$, $att$ and $k$, where $att$ is the attribute in $U_k$. If the submitted $pcpk_{gid,i}$ is illegal, it returns $\otimes$; otherwise, it returns $rask^0_{gid,i}$. 


Challenge phase: $A$ submits two equal length message $m_0$, $m_1$ and access policy $A^*$. The challenger flips a random coin $b \in \{0, 1\}$ and encrypts $m_b$ under $A^*$. The ciphertext $CT^*$ is given to $A$.

Key query phase 2: $A$ is once again to repeat the steps in Key query phase 1.

Guess: the adversary $A$ outputs a guess $b'$ of $b$.

The advantage that an adversary $A$ wins this game is $Adv(\lambda) = |Pr[b' = b] - \frac{1}{2}|$. The proposed scheme is secure if for any polynomial time, the advantage $Adv(\lambda)$ is negligible.

2.3 Adversary Model and Design Goal

In the profile matching process, there usually exists two main adversary models: In the honest-but-curious (HBC) model [17], an attacker honestly follows the protocol but tries to get more information from the received message than allowed. In this paper, we suppose all the authorities and users are honest-but-curious. In malicious model [18], an attacker tries to learn more information using background knowledge beyond his/her received message or by deliberately deviating from the protocol.

The main goal as well as the great challenge of our scheme is to conduct efficient matching against the chosen-plaintext attack and collusion attack.

3 Proposed Scheme

In this section, we will propose a piecewise multi-authority CP-ABE scheme. It mainly consists of the following phases: system initialization, key generation, information encryption, profile matching and decryption.

3.1 System Initialization

GlobalInit: On input $1^\lambda$, where $\lambda$ is the security parameter, this algorithm outputs the global public parameter $GPRA$. $G$ is a bilinear cyclic group with the order $N = p_1p_2p_3$, where $p_1, p_2, p_3$ are distinct big prime numbers. $G_{p_1}$ is the subgroup of $G$ with order $p_1$, $g$ is the generator of $G_{p_1}$ and $X_3$ is the generator of $G_{p_3}$. Randomly choose $h \in_r G_{p_1}$. Finally, the global public parameter is published as $GPRA = \{N, g, h, X_3, \Sigma_{sig}\}$, where $\Sigma_{sig} = \{KeyGen, Sign, Verify\}$ is the secure signature scheme against chosen-plaintext attack.

CASSetup: On input $GPRA$, this algorithm outputs $CA_i$’s public parameter $CAPAR_i$, public key $CAPK_i$ and master key $CAMSK_i$. First of all, each $CA_i$ runs the algorithm KeyGen in the $\Sigma_{sig}$ to generate a pair of secret key and public key $<sk_i, pk_i>$. $CA_i$ randomly chooses $\alpha_i, a_i, \in_r Z_N$ to generate master secret key $CAMSK_i = (\alpha_i, a_i, sk_i)$ and empty table $T_i$, then publishes the public parameter $CAPAR_i = (e(g, g)^{\alpha_i}, g^{a_i})$ and public key $CAPK_i = pk_i$.

AASetup: This algorithm takes $GPRA$, $AA_k$’s index $k$ and the attribute universe $U_k$ belonging to $AA_k$ as input, and outputs master secret key $AAMSK_k$, public parameter $AAPRA_k$ and public key $AAPK_k$. For each $att \in U_k$, $AA_k$ randomly selects $s_{att} \in_r Z_N$ and $v_{k,i} \in_r Z_N$, then computes $T_{att} = g^{s_{att}}$ and $V_{k,i} = g^{v_{k,i}}$. Finally, $AA_i$ sets its master secret key $AAMSK_k = (v_{k,i}, \{s_{att} | att \in U_k\})$, and publishes the public parameter $AAPAR_k = \{T_{att} | att \in U_k\}$ and public key $AAPK_k = V_{k,i}$.

3.2 Key Generation

In this phase, a responder submits his/her information to request the public and secret keys.
3.2.1 CA key Generation

In this step, responder registers his/her gid to CA for requesting the responder-identity keys and finally the rppk<sub>gid,i</sub> is published. The detailed procedure is shown in Algorithm 1.

**Algorithm 1: Responder-identity-Keys Generation**

- **Input:** responder’s identifier gid
- **Output:** responder’s signature σ<sub>gid,i</sub>, public key rppk<sub>gid,i</sub>, a pair of secret keys <rpsk<sub>0</sub>gi<i>d,i</sub>, rpsk<sub>1</sub>gi<i>d,i</sub>>

1. randomly select c<sub>i</sub> ∈ Z<sub>N</sub>, r<sub>gid,i</sub> ∈ Z<sub>N</sub>, R<sub>gid,i</sub>, R'<sub>gid,i</sub>, R''<sub>gid,i</sub> ∈ G<sub>p3</sub>;
2. compute: rpsk<sub>j</sub>gi<i>d,i</sub> = c<sub>i</sub>, L<sub>gid,i</sub> = g<sup>r<sub>gid,i</sub></sup>R<sub>gid,i</sub>, L'<sub>gid,i</sub> = (g<sup>a<sub>i</sub></sup>)<sup>r<sub>gid,i</sub></sup>R'<sub>gid,i</sub>, μ<sub>0</sub>i = α<sub>u,i</sub>, μ<sub>1</sub>i = α<sub>i</sub> − α<sub>u,i</sub>;
3. for j from 0 to 1 do
4. compute rpsk<sub>j</sub>gi<i>d,i</sub> = g<sup>μ<sub>j</sub>i</sup>a<sub>i</sub> + c<sub>i</sub>h<sup>r<sub>gid,i</sub></sup>R<sub>gid,i</sub>;
5. end
6. for k from 1 to K do
7. randomly choose R<sub>gid,k,i</sub> from G<sub>p3</sub>;
8. compute Γ<sub>gid,k,i</sub> = V<sub>k,i</sub>[(a<sub>i</sub> + c<sub>i</sub></sup)r<sub>gid,i</sub>R<sub>gid,k,i</sub>];
9. end
10. generate σ<sub>gid,i</sub> = Sig<sub>sk</sub>(gid||L<sub>gid,i</sub>||L'<sub>gid,i</sub>||∪<sub>k=1</sub>K Γ<sub>gid,k,i</sub>, σ<sub>gid,i</sub>) and rppk<sub>gid,i</sub> = (gid, L<sub>gid,i</sub>, L'<sub>gid,i</sub>, {Γ<sub>gid,k,i</sub>, σ<sub>gid,i</sub>});
11. add (c<sub>i</sub>, gid) to T<sub>i</sub>;
12. return σ<sub>gid,i</sub>, rppk<sub>gid,i</sub>, <rpsk<sub>0</sub>gi<i>d,i</sub>, rpsk<sub>1</sub>gi<i>d,i</sub>>

3.2.2 AA key Generation

When a responder submits his/her keys to AA<sub>k</sub> for the secret key concerning some attribute att ∈ U<sub>k</sub> in his/her attribute set AS<sub>gid</sub>. The authorities first will verify the identity of the responder according to the formula:

\[
VALID \leftarrow \begin{cases} 
\varepsilon(g, Γ_{gid,k,i}) = \varepsilon(V_{k,i}, L'_{gid,i}, L''_{gid,i}) \\
Verif(\sigma_{gid,i}, AS_{gid}, L'_{gid,i}, ∪_{k=1}^K Γ_{gid,k,i}, σ_{gid,i}) 
\end{cases}
\]

If it fails to pass one of the verification, AA<sub>k</sub> outputs ⊘ which means that the responder is invalid and the system will end the whole procedure.

If the verification is correct, AA<sub>k</sub> will run the Algorithm 2 to generate rask<sub>gid,i</sub>. After running the algorithm, AA<sub>k</sub> transmits AS<sub>gid</sub> to the cloud server to find a matcher.

3.3 Encryption

This algorithm is performed on the initiator’s hand-held device. Suppose the initiator’s real identity is ID, the personal profile is m<sub>profile</sub>, the symmetric identity encryption key is K<sub>id</sub>, the personal profile encryption key is K<sub>profile</sub>, the access policy is A = (A, ρ), the secure symmetric encryptions are E<sub>K<sub>id</sub></sub>(·) and E<sub>K<sub>profile</sub></sub>(·). The access policy is defined by a LSSS matrix (A, ρ), where A is a l × n matrix and ρ will map each row A<sub>x</sub> in A to get an attribute ρ(x). ρ is
Algorithm 2: Attribute Key Generation

Input: responder’s identifier $gid$, attribute $att$

Output: the attribute secret key $rask_{gid,i}$

1. randomly select $R_{att,gid}^i \in G_{p_i}$;
2. for $i$ from 1 to $D$
   3. for $\forall att \in U_k \cap AS_{gid}$
      4. compute $pask_{att,gid,i} = (\Gamma_{gid,k,i})^{s_{att}/v_{k,i}}R_{att,gid}^i = T_{att}^{(a_i+c_i)r_{gid,i}}R_{gid,k,i}^{s_{att,i}/v_{k,i}}R_{att,gid}^i$;
      5. set $R_{att,gid,i} = R_{att,gid}^i$;
      6. $pask_{att,gid,i}$ is denoted as $T_{att}^{(a_i+c_i)r_{gid,i}}R_{att,gid}^i$;
   7. end
8. end
9. generate $rask_{gid,i} = \{rask_{att,gid,i} | att \in AS_{gid}\}$;
10. return $rask_{gid,i}$

required that when mapping different rows, the attribute must not be the same. The detailed encryption procedure is shown in Algorithm 3.

Algorithm 3: Encryption

Input: $ID, m_{profile}, K_{id}, K_{profile}, GRPA, AAPAR_k, CAPAR_i, E_{K_{id}}(\cdot), E_{K_{profile}}(\cdot)$

Output: ciphertext: $C_{A,\rho}, CT_{id}, CT_{profile}$

1. choose a random vector $\vec{v} = (s,v_2,...,v_n) \in Z_N^n$;
2. for $x$ from 1 to $l$
   3. select a random number $r_x$, where $r_x \in Z_N$;
   4. compute $C_x = h^{A_x\cdot\vec{v}^T-r_x}^{\rho(x)}$;
5. end
6. compute $C' = g^s$ and $C'' = g^{a_{is}}$;
7. for $sth$ in $\{id, profile\}$
   8. compute $CT_{K_{sth}} = K_{sth} \prod_{i=1}^{d} e(g,g)^{\alpha_{is}}$
9. end
10. compute $CT_{id} = E_{K_{id}}(ID)$ and $CT_{profile} = E_{K_{profile}}(m_{profile})$;
11. define $C_{A,\rho} = \begin{cases} CT_{K_{id}} = K_{id} \prod_{i=1}^{d} e(g,g)^{\alpha_{is}}, \\ CT_{K_{profile}} = K_{profile} \prod_{i=1}^{d} e(g,g)^{\alpha_{is}}, \\ \{C_x = h^{A_x\cdot\vec{v}^T-r_x}^{\rho(x)}C''_x = g'^{x}, \\ C' = g^s, \\ C'' = g^{a_{is}} \}, \end{cases}, x \in \{1,2,...,l\}$;
12. return $C_{A,\rho}, CT_{id}, CT_{profile}$
3.4 Profile Matching and Decryption

First, the cloud server will help the responder find a matcher. If the responder’s attribute set $A_{\Sigma}$ satisfies the access policy $\mathbb{A} = (A, \rho)$, which means there exists constants $\omega_x \in \Sigma_N$ and $\sum_{p(x) \in A_{\Sigma}} \omega_x A_x = (1, 0, ..., 0)$, then the cloud server transmits $C_{A_{\rho}}$, $CT_{id}$, $CT_{profile}$ to the responder. When receiving the ciphertexts, the responder runs Algorithm 4 to decrypt.

Algorithm 4: Decryption

\begin{algorithm}
\begin{algorithmic}
\State Input: $rask_{gid,i}, \mu_i^0, \mu_i^1, C_{A_{\rho}}, CT_{id}, CT_{profile}$, $< rpsk_{gid,i}^0, rpsk_{gid,i}^1 >$, $D_{K_{profile}}()$
\State Output: initiator’s identity $ID$ and personal profile $m_{profile}$
\State compute $e((C')^{\rho_{sk_{gid,i}}^0}, C'_{\mu_i^0})$ \quad $\rightarrow e((g,g)^{\alpha_u-i}, e(g,g)^{(\alpha_u-a_u,i)}$;
\State $\quad e((C')^{\rho_{sk_{gid,i}}^1}, C'_{\mu_i^1})$ \quad $\rightarrow e((g,g)^{\alpha_u}, e(g,g)^{(\alpha_u-a_u,i)})$
\For{$sth \in \{id, profile\}$} \begin{algorithmic}
\State compute $K_{sth} = \frac{CT_{K_{sth}}}{\prod_{i=1}^{n}(e(g,g)^{\alpha_u}, e(g,g)^{(\alpha_u-a_u,i)})}$
\EndFor
\State compute $ID = D_{K_{id}}(CT_{id})$, $m_{profile} = D_{K_{profile}}(CT_{profile})$, where $D_{K_{id}}()$ and $D_{K_{profile}}()$ are corresponding decryption algorithms of $K_{id}$ and $K_{profile}$; \State return $ID$, $m_{profile}$
\end{algorithmic}
\end{algorithm}

4 Security Analysis

In this section, we give security proof of our proposed scheme to achieve multi-authority privacy-preserving friend discovery system. Suppose there exists an adversary $\mathbb{A}$ and a challenger $\mathbb{C}$.

**Definition 1.** Our proposed scheme can achieve privacy-preserving if it is secure in the security game in Section 2.2.

**Lemma 1.** Our proposed scheme achieves privacy against adversaries.

**Proof.** Suppose the adversary $\mathbb{A}$ can break our proposed scheme with advantage $Adv_{\mathbb{A}}$, then the challenger $\mathbb{C}$ can break the underlying multi-authority CP-ABE scheme with the advantage $Adv_{\mathbb{C}}$ which equals to $Adv_{\mathbb{A}}$.

**Setup:** the multi-authority CP-ABE scheme gives $\mathbb{C}$ the public parameters $GPK = \{N, g, h, X_3, \Sigma_{\text{sig}}\}$, $CPK_i = e(g,g)^{\alpha_i}$, $CAPK_i = \text{VerifyKey}_i$, $APK_k = \{T_{att} | att \in U_k\}$, $ACP_{k_i} = V_{k,i}$, $\mathbb{C}$ randomly selects $a_i \in \Sigma_N$ and gives $\mathbb{A}$ the following public parameters $GPRA = \{N, g, h, X_3, \Sigma_{\text{sig}}\}$, $CAPAR_i = (e(g,g)^{\alpha_i}, g^{\alpha_i})$, $CAPK_i = pk_i$, $AAPAR_k = \{T_{att} | att \in U_k\}$, $AAPC_{k_i} = V_{k,i}$ and $T_i = \emptyset$. Then, specifies the target uncorrupted $CA$ with index $i*$ and a set of corrupted $A_3$. $\mathbb{C}$ inputs $i*$ and gets $CMSK_k = \{a_i, \text{SignKey}_i\}$, $AMSK_k = \{v_{k,i}, s_{att} | att \in U_k\}$. Then $\mathbb{C}$ gives $CMSK_i = (a_i, a_i, \text{SignKey}_i)$ and $AMSK_k = \{v_{k,i}, s_{att} | att \in U_k\}$.

**Key query phase 1.** (1) When $\mathbb{A}$ submits $gid$ and $i*$ to the random oracle $O^{C\text{AKeyGen}}$ and $\mathbb{C}$ submits $(gid, i*)$ to the multi-authority CP-ABE scheme obtaining $ucsk_{gid,i^*}^{0,M_{A}} = \ldots$
\[ g^{a_{u,i}r} h_{g,i}^{r_{g,i}} R_{g,i}, \text{ucsk}^{1,MA}_{g,i} = g^{a_{u,i}r - a_{u,i}r} h_{g,i}^{r_{g,i}} R_{g,i}, L^{MA}_{g,i} = g^{r_{g,i}r'} \text{ and } \Gamma^{MA}_{g,i} = V_{g,i}^{r_{g,i}r'} R_{g,i}. \]

\[ C \text{ randomly selects } c_i \in \mathbb{Z}_N, t_{g,i} \in \mathbb{Z}_N, R' \in \mathbb{G}_p \text{ and sets } r_{g,i} = \frac{\Gamma^{MA}_{g,i}}{(a_{r,c_i}),} \]

\[ \text{where } c_i \text{ is noted that the above operations are with the restriction that } \text{progress and practicability. We assume that both of the initiator and the responder have mobile devices} \]

\[ \text{and sends them to } A: \]

\[ \left\{ \begin{array}{l}
\text{pcsk}^{0,MA}_{g,i} = (\text{ucsk}^{0,MA}_{g,i})^{-1} \\
\text{pcsk}^{1,MA}_{g,i} = (\text{ucsk}^{1,MA}_{g,i})^{-1} \\
R_{g,i} = V_{g,i}^{(a_{r,c_i})} R_{g,i} \\
L_{g,i} = (g^{a_{r,c_i}})(R'_{g,i})^{-1} \end{array} \right. \]

\[ \text{Then } C \text{ adds } (c_i, gid) \text{ to } T_i. \]

(2) The adversary \( A \) submits \((pcpk_{g,d}, k, att)\) to \( O^{AK}KeyGen \) to obtain attribute key, \( C \) first verifies:

\[ \text{VALID} \leftarrow \left\{ \begin{array}{l}
\text{e}(g, \Gamma_{g,d,k,i}) = \text{e}(V_k, L'_{g,d,i} \text{pcsk}_{g,i}^{r_{g,i}r'}) \\
\text{VERIFY}(\text{pk}_k, \text{pk}_i, \text{pk}_{k,i}, \text{pk}_{g,i}, \text{pk}_{g,d}, \text{pk}_{g,i}) \\
\Gamma_{g,d,k,i} = (g^{a_{r,c_i}})(R'_{g,i})^{-1} \end{array} \right. \]

If the verification is passed, \( C \) randomly chooses \( R_{att,gid} \in \mathbb{G}_p \) and computes

\[ \text{pask}_{att,gid,i} = (\Gamma_{g,d,k,i})^{s_{att}/v_{k,i}} R'_{att,gid} = T_{att}^{(a_{r,c_i})} R'_{att,gid,i} \]

\[ \text{where } R_{att,gid,i} = r_{g,i}^{(a_{r,c_i})} R'_{att,gid}. \]

Finally, \( C \) transmits \( \text{pask}_{att,gid,i} \) to \( A \).

**Challenge phase.** The adversary \( A \) gives \( C \) the access policy \( A^* = (A^*, \rho) \) and two messages \( m_0, m_1 \) with the same length. Then \( C \) submits \((A^*, m_0, m_1)\) to the multi-authority CP-ABE scheme and gets the following ciphertexts:

\[ \left\{ \begin{array}{l}
CT_{K_{eth}} = m_b \prod_{i=1}^d \text{e}(g, g)^{a_{r,s}} (sth \in \{id, profile\}) \\
C' = g^{a_{r,s}} \\
C'' = g^{a_{r,x}} \\
\{C_x = h^{A^*\rho} T_{\rho(x)}^{-r_x}, C'' = g^{r_x} \} (x \in \{1, 2, \ldots, l\}) \end{array} \right. \]

It is noted that the above operations are with the restriction that \( AS_{g,d,A} \) cannot satisfy the access policy \( A^* \).

**Key query phase 2.** \( A \) is once again to repeat the operations in Key query phase 1.

**Guess.** The adversary \( A \) outputs a guess \( b' \) of \( b \), and \( C \) submits \( b' \) to the multi-authority CP-ABE scheme.

From the above analysis, it is obviously that the distribution of parameters, keys and ciphertexts are the same as the real scheme, there we can get \( \text{Adv}_C = \text{Adv}_A \).

5 Performance Analysis

In this section, we evaluate the proposed scheme with several existing works in terms of efficiency and practicability. We assume that both of the initiator and the responder have mobile devices with a 2.3 GHz CPU, e.g., Nexus 5 announced in 2013. This smart phone supports both
Bluetooth 4.0 and dual frequency Wi-Fi. We use Eclipse to implement the simulation code and it was written in Java. We perform the efficiency simulation and comparisons between the [19], [20] and our proposed scheme. The size of users’ attribute sets is fixed in 30 and \( n \) denotes the number of participated responders.

Figure 2(a) and Figure 2(b) illustrate the computational cost among [19], [20] and our scheme respectively on the initiator’s and responder’s ends. It is obvious that in [19] [20] the computational cost increase as the number of responders grows since it is required for the initiator to generate one group signature for each responder. Figure 2(c) shows communication overhead comparison among [19], [20] and our proposed scheme. It is apparent that the communication cost of [19] and [20] sharply grows as the number of responders increases from 50 to 500.

![Computational Cost Comparison](image1)

![Communication Cost Comparison](image2)

Figure 2: Computation and Communication Comparison

6 Conclusion

In this paper, a distributed multi-authority attribute-based encryption friend discovery scheme is proposed to achieve multi-level privacy and users can easily achieve fine-grained access control. The detailed security analysis demonstrates that the scheme can resist chosen-plaintext attack as well as collusion attack in the standard model and performs well in terms of storage, computational and communication cost. In our future work, we will improve the scheme by involving the functions of ciphertexts updating and revocation.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 61103035 and Grant No. 31470028, and the Fundamental Research Funds for the Central Universities of Central South University (2016zzts337).

References


