Procedia Computer Science
Volume 80, 2016, Pages 617–626
ICCS 2016. The International Conference on Computational Science

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

Wenbo Wang[1], Fang Qi[1], Xiaoqiang Wu[2], and Zhe Tang[1] *

[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

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.350

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 crytosytem 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 computationanl as well as storage 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 destributed 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 the 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 computaion overhead of the client can be greatly reduced. The multi-authorities are designed to be destributed, 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 destributed 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,..., CA_D) and K attribute authorities (AA_1,..., 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 ($i \in 1, 2, ..., D$). Each attribute authority AA_k ($k \in 1, 2, ..., K$) manages a set of attributes $U_k(U_i \cap U_j = \emptyset \wedge U = \cup_{k=1}^{K} U_k)$ $(i, j \in \{1, 2, ..., K\} \wedge i \neq j)$. Each authorized responder with attribute set $\mathcal{AS}_{gid}$ will obtain their attribute secret keys from the corresponding AA_ks. 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.

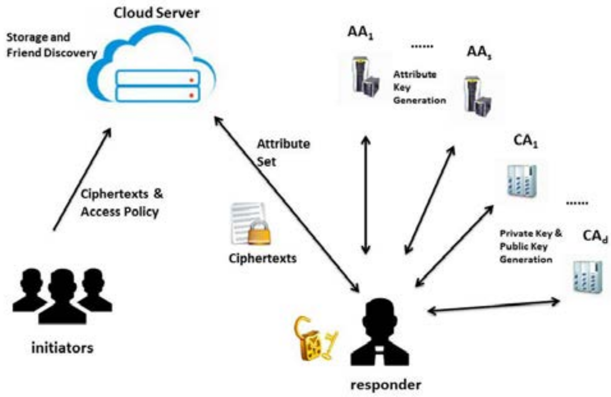


Figure 1: Architecture of Cloud Friend Discovery System

2.2 Security Model

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

Key query phase 1: The adversary $\mathcal{A}$ tries to query the following random oracles.

$O^{CAKeyGen(gid,i)}$: $\mathcal{A}$ 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 $rppk_{gid,i}$.

$O^{AAKeyGen(att,rppk_{gid,i},k)}$: $\mathcal{A}$ queries with $rppk_{gid,i}$, 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: $\mathcal{A}$ submits two equal length message m_0, m_1 and access policy $\mathbb{A}^*$. The challenger flips a random coin $b \in \{0, 1\}$ and encrypts m_b under $\mathbb{A}^*$. The ciphertext CT^* is given to $\mathcal{A}$.

Key query phase 2: $\mathcal{A}$ is once again to repeat the steps in **Key query phase 1**.

Guess: the adversary $\mathcal{A}$ outputs a guess b' of b.

The advantage that an adversary $\mathcal{A}$ 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^λ, where λ is the security parameter, this algorithm outputs the global public parameter $GPRA$. $\mathbb{G}$ 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. $\mathbb{G}_{p_i}$ is the subgroup of $\mathbb{G}$ with order p_i, g is the generator of $\mathbb{G}_{p_1}$ and X_3 is the generator of $\mathbb{G}_{p_3}$. 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 $\Sigma_{sig} = \{KeyGen, \mathrm{Sign}, Verify\}$ is the secure signature scheme against chosen-plaintext attack.

CASetup: 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 Σ_{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_i for requesting the responder-identity keys and finally the $rppk_{gid,i}$ 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 $\sigma_{gid,i}$, public key $rppk_{gid,i}$, a pair of secret keys $< rpsk^0_{gid,i}, rpsk^1_{gid,i} >$

randomly select $c_i \in \mathbb{Z}_N^*$, $r_{gid,i} \in \mathbb{Z}_N$, $R_{gid,i}, R'_{gid,i}, R''_{gid,i} \in \mathbb{G}_{p_3}$;
compute: $rpsk'_{gid,i} = c_i$, $L_{gid,i} = g_{gid,i}^{r_{gid,i}} R'_{gid,i}$, $L'_{gid,i} = (g^{a_i})^{r_{gid,i}} R''_{gid,i}$, $\mu_i^0 = \alpha_{u,i}$, $\mu_i^1 = \alpha_i - \alpha_{u,i}$;
for *j from 0 to 1* **do**
  compute $rpsk^j_{gid,i} = g^{\frac{\mu_i^j}{a_i+c_i}} h_{gid,i}^{r_{gid,i}} R_{gid,i}$;
end
for *k from 1 to K* **do**
  randomly choose $R_{gid,k,i}$ from $\mathbb{G}_{p_3}$;
  compute $\Gamma_{gid,k,i} = V_{k,i}^{(a_i+c_i)r_{gid,i}} R_{gid,k,i}$;
end
generate $\sigma_{gid,i} = Sig_{sk_i}(gid||L_{gid,i}||L'_{gid,i}|| \cup_{k=1}^{K} \Gamma_{gid,k,i})$ and $rppk_{gid,i} = (gid, L_{gid,i}, L'_{gid,i}, \{\Gamma_{gid,k,i}\}, \sigma_{gid,i})$;
add (c_i, gid) to T_i;
return $\sigma_{gid,i}$, $rppk_{gid,i}$, $< rpsk^0_{gid,i}, rpsk^1_{gid,i} >$

3.2.2 AA key Generation

When a responder submits his/her keys to AA_k for the secret key concerning some attribute $att \in U_k$ in his/her attribute set $\mathcal{AS}_{gid}$. The authorities first will verify the identity of the responder according to the formula:

$$VALID \stackrel{?}{\leftarrow} \begin{cases} e(g, \Gamma_{gid,k,i}) \stackrel{?}{=} e(V_{k,i}, L'_{gid,i} L_{gid,i}^{rpsk'_{gid,i}}) \\ Verify_{pk_i}(gid||L_{gid,i}||L'_{gid,i}|| \cup_{k=1}^{K} \Gamma_{gid,k,i}, \sigma_{gid,i}) \end{cases} \quad (1)$$

If it fails to pass one of the verification, AA_k outputs $\otimes$ which means that the responder is invalid and the system will end the whole procedure.

If the verification is correct, AA_k will run the ***Algorithm 2*** to generate $rask_{gid,i}$. After running the algorithm, AA_k transmits $\mathcal{AS}_{gid}$ 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 $m_{profile}$, the symmetric identity encryption key is K_{id}, the personal profile encryption key is $K_{profile}$, the access policy is $\mathbb{A} = (A, \rho)$, the secure symmetric encryptions are $E_{K_{id}}(\cdot)$ and $E_{K_{profile}}(\cdot)$. The access policy is defined by a LSSS matrix (A, ρ), where A is a $l \times n$ matrix and ρ will map each row A_x in A to get an attribute $\rho(x)$. ρ is

Algorithm 2: Attribute Key Generation

```
Input: responder's identifier gid, attribute att
Output: the attribute secret key rask_{gid,i}
randomly select R'_{att,gid} ∈ G_{p_i};
for i from 1 to D do
    for ∀att ∈ U_k ∩ AS_{gid} do
        compute pask_{att,gid,i} = (Γ_{gid,k,i})^{s_{att}/v_{k,i}} R'_{att,gid} = T_{att}^{(a_i+c_i)r_{gid,i}} R_{gid,k,i}^{s_{att,i}/v_{k,i}} R'_{att,gid};
        set R_{att,gid,i} = R_{gid,k,i}^{s_{att}/v_{k,i}} R'_{att,gid};
        pask_{att,gid,i} is denoted as T_{att}^{(a_i+c_i)r_{gid,i}} R_{att,gid,i};
    end
end
generate rask_{gid,i} = {rask_{att,gid,i} | att ∈ AS_{gid}};
return rask_{gid,i}
```

required that when mapping different row, 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}, $CT_{profile}$

choose a random vector $\vec{v} = (s, v_2, ..., v_n) \in Z_N^n$;

for x *from 1 to* l **do**

  select a random number r_x, where $r_x \in \mathbb{Z}_N$;

  compute $C_x = h^{A_x \cdot \vec{v}} T_{\rho(x)}^{-r_x}$;

end

compute $C' = g^*$ and $C'' = g^{a_i s}$;

for sth *in* $\{id, profile\}$ **do**

  compute $CT_{K_{sth}} = K_{sth} \prod_{i=1}^{d} e(g,g)^{\alpha_i s}$

end

compute $CT_{id} = E_{K_{id}}(ID)$ and $CT_{profile} = E_{K_{profile}}(m_{profile})$;

define $$C_{A,\rho} = \begin{pmatrix} CT_{K_{id}} = K_{id} \prod_{i=1}^{d} e(g,g)^{\alpha_i s}, \\ CT_{K_{profile}} = K_{profile} \prod_{i=1}^{d} e(g,g)^{\alpha_i s}, \\ \{C_x = h^{A_x \cdot \vec{v}} T_{\rho(x)}^{-r_x}, C'_x = g^{rx}\}, \\ C' = g^s, \\ C'' = g^{a_i s} \end{pmatrix}, x \in \{1, 2, ..., l\};$$

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 $\mathcal{AS}_{gid}$ satisfies the access policy $\mathbb{A} = (A, \rho)$, which means there exists constants $\omega_x \in \mathbb{Z}_N$ and $\sum_{\rho(x)\in\mathcal{AS}_{gid}} \omega_x A_x = (1, 0, ..., 0)$, 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

Input: $rask_{gid,i}$, μ_i^0, μ_i^1, $C_{A,\rho}$, CT_{id}, $CT_{profile}$, $< rpsk^0_{gid,i}, rpsk^1_{gid,i} >$, $D_{K_{id}}(\cdot)$, $D_{K_{profile}}(\cdot)$

Output: initiator's identity ID and personal profile $m_{profile}$

compute $\begin{cases} \frac{e((C')^{rpsk'_{gid,i}} C''_i, rpsk^0_{gid,i})}{\Pi_{\rho(x)\in\mathcal{AS}_{gid}} (e(C_x, L_{gid,i}^{rpsk'_{gid,i}}) e(C'_x, rask_{\rho(x),gid,i}))^{\omega_x}} \\ \frac{e((C')^{rpsk'_{gid,i}} C''_i, rpsk^1_{gid,i})}{\Pi_{\rho(x)\in\mathcal{AS}_{gid}} (e(C_x, L_{gid,i}^{rpsk'_{gid,i}}) e(C'_x, rask_{\rho(x),gid,i}))^{\omega_x}} \end{cases} \rightarrow \begin{cases} e(g,g)^{\alpha_{u,i}} \\ e(g,g)^{(\alpha_u - \alpha_{u,i})} \end{cases}$;

for $sth \in \{id, profile\}$ **do**

  compute $K_{sth} = \frac{CT_{K_{sth}}}{\Pi_{i=1}^{D}(e(g,g)^{\alpha_{u,i}} e(g,g)^{(\alpha_i - \alpha_{u,i})})}$

end

compute $ID = D_{K_{id}}(CT_{id})$, $m_{profile} = D_{K_{profile}}(CT_{profile})$, where $D_{K_{id}}(\cdot)$ and $D_{K_{profile}}(\cdot)$ are corresponding decryption algorithms of K_{id} and $K_{profile}$;

return ID, $m_{profile}$

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 $\mathcal{A}$ and a challenger $\mathcal{C}$.

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

Setup: the multi-authority CP-ABE scheme gives $\mathcal{C}$ the public parameters $GPK = \{N, g, h, X_3, \Sigma_{sig}\}$, $CPK_i = e(g,g)^{\alpha_i}$, $CAPK_i = VerifyKey_i$, $APK_k = (\{T_{att} | att \in U_k\})$, $ACPK_k = V_{k,i}$. $\mathcal{C}$ randomly selects $a_i \in \mathbb{Z}_N$ and gives $\mathcal{A}$ the following public parameters $GPRA = \{N, g, h, X_3, \Sigma_{sig}\}$, $CAPAR_i = (e(g,g)^{\alpha_i}, g^{a_i})$, $CAPK_i = pk_i$, $AAPAR_k = (\{T_{att} | att \in U_k\})$, $AAPK_k = V_{k,i}$ and $T_i = \emptyset$. Then, specifies the target uncorrupted CA with index i^* and a set of corrupted AAs. $\mathcal{C}$ inputs i^* and gets $CMSK_i = \{\alpha_i, \text{Sign}Key_i\}$, $AMSK_k = (v_{k,i}, \{s_{att} | att \in U_k\})$. Then $\mathcal{C}$ gives $CAMSK_i = (\alpha_i, a_i, SingKey_i)$ and $AAMSK_k = AMSK_k$ to $\mathcal{A}$.

Key query phase 1. (1) When $\mathcal{A}$ submits gid and i^* to the random oracle $O^{CAKeyGen}$ and $\mathcal{C}$ submits (gid, i^*) to the multi-authority CP-ABE scheme obtaining $ucsk^{0,MA}_{gid,i^*} =$

$g^{\alpha_{u,i^*}} h^{r^{MA}_{gid,i^*}} R_{gid,i^*}$, $ucsk^{1,MA}_{gid,i^*} = g^{\alpha_{i^*}-\alpha_{u,i^*}} h^{r^{MA}_{gid,i^*}} R_{gid,i^*}$, $L^{MA}_{gid,i^*} = g^{r^{MA}_{gid,i^*}} R'_{gid,i^*}$ and $\Gamma^{MA}_{gid,k,i*} = V_{k,i^*}^{r^{MA}_{gid,i^*}} R_{gid,k,i^*}$. $\mathcal{C}$ randomly selects $c_i \in \mathbb{Z}^*_N$, $t_{gid,i^*} \in \mathbb{Z}_N$, $R'' \in \mathbb{G}_{p_3}$ and sets $r_{gid,i^*} = \frac{r^{MA}_{gid,i^*}}{(a_{i^*}+c_{i^*})}$, $\mathcal{C}$ computes the following parameters and sends them to $\mathcal{A}$:

$$\begin{cases} pcsk^0_{gid,i^*} = (ucsk^{0,MA}_{gid,i^*})^{\frac{1}{a_{i^*}+c_{i^*}}} = g^{\frac{\alpha_{u,i^*}}{a_{i^*}+c_{i^*}}} h^{r_{gid,i^*}} R_{gid,i^*}^{\frac{1}{a_{i^*}+c_{i^*}}} \\ pcsk^1_{gid,i^*} = (ucsk^{1,MA}_{gid,i^*})^{\frac{1}{a_{i^*}+c_{i^*}}} = g^{\frac{\alpha_{i^*}-\alpha_{u,i^*}}{a_{i^*}+c_{i^*}}} h^{r_{gid,i^*}} R_{gid,i^*}^{\frac{1}{a_{i^*}+c_{i^*}}} \\ \Gamma_{gid,k,i^*} = V_{k,i^*}^{(a_{i^*}+c_{i^*})r_{gid,i^*}} R_{gid,k,i^*} \\ pcsk'_{gid,i^*} = c_{i^*} \\ L_{gid,i^*} = g^{r_{gid,i^*}} (R'_{gid,i^*})^{\frac{1}{a_{i^*}+c_{i^*}}} \\ L'_{gid,i^*} = (g^{a_{i^*}})^{r_{gid,i^*}} (R'_{gid,i^*})^{\frac{a_{i^*}}{a_{i^*}+c_{i^*}}} R'' \end{cases} \tag{2}$$

Then $\mathcal{C}$ adds (c_{i^*}, gid) to T_i.

(2) The adversary $\mathcal{A}$ submits $(pcpk_{gid,d}, k, att)$ to $O^{AAKeyGen}$ to obtain attribute key, $\mathcal{C}$ first verifies:

$$VALID \overset{?}{\leftarrow} \begin{cases} e(g, \Gamma_{gid,k,i^*}) \overset{?}{=} e(V_{k,i^*}, L'_{gid,i^*} L_{gid,i^*}^{pcsk'_{gid,i^*}}) \\ Verify_{pk_{i^*}}(gid||L_{gid,i^*}||L'_{gid,i^*}|| \cup_{k=1}^{K} \Gamma_{gid,k,i^*}, \sigma_{gid,i^*}) \end{cases} \tag{3}$$

If the verification is passed, $\mathcal{C}$ randomly chooses $R'_{att,gid} \in G_{p_i}$ and computes

$$pask_{att,gid,i^*} = (\Gamma_{gid,k,i^*})^{s_{att}/v_{k,i^*}} R'_{att,gid} = T_{att}^{(a_{i^*}+c_{i^*})r_{gid,i^*}} R_{att,gid,i^*} \tag{4}$$

where $R_{att,gid,i^*} = R_{gid,k,i^*}^{s_{att}/v_{k,i^*}} R'_{att,gid}$. Finally, $\mathcal{C}$ transmits $pask_{att,gid,i^*}$ to $\mathcal{A}$.

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

$$\begin{cases} CT_{K_{sth}} = m_b \prod_{i=1}^{d} e(g,g)^{\alpha_i s} (sth \in \{id, profile\}) \\ C' = g^s \\ C'' = g^{a_i s} \\ \{C_x = h^{A^*_x \vec{v}} T_{\rho(x)}^{-r_x}, C'_x = g^{r_x}\} (x \in \{1, 2, ..., l\}) \end{cases} \tag{5}$$

It is noted that the above operations are with the restriction that $\mathcal{AS}_{gid_{\mathcal{A}}}$ cannot satisfy the access policy $\mathbb{A}^*$.

Key query phase 2. $\mathcal{A}$ is once again to repeat the operations in **Key query phase 1.**

Guess. The adversary $\mathcal{A}$ outputs a guess b' of b, and $\mathcal{C}$ submits b' to the multi-authority CP-ABE scheme.

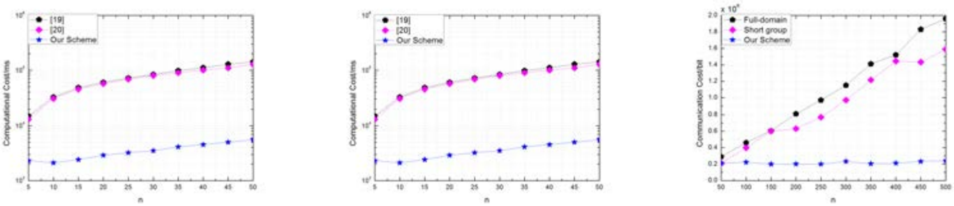
From the above analysis, it is obviously that the distribution of parameters, keys and ciphertexts are the same as the real scheme, there we can get $Adv_{\mathcal{C}} = Adv_{\mathcal{A}}$.

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 appatently that the communication cost of [19] and [20] sharply grows as the number of responders increases from 50 to 500.



(a) Computational Cost on Initiator's End (b) Computational Cost on Responder's End (c) Communication Cost

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

[1] Lan Zhang, Xiang-Yang Li, and Yunhao Liu. Message in a sealed bottle: Privacy preserving friending in social networks. In *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, pages 327–336. IEEE, 2013.

[2] E Noah. Mobile social networking shows promise, but rich media has higher engagement, 2011. Available at `http://www.emarketer.com/Articles`.

[3] Yufeng Wang and Jing Xu. Overview on privacy-preserving profile-matching mechanisms in mobile social networks in proximity (msnp). In *Information Security (ASIA JCIS), 2014 Ninth Asia Joint Conference on*, pages 133–140. IEEE, 2014.

[4] Guojun Wang, Qin Liu, and Jie Wu. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In *in Proc. of ACM Conference on Computer and Communications Security (CCS*, pages 735–737, 2010.

[5] Justin Manweiler, Ryan Scudellari, and Landon P Cox. Smile: encounter-based trust for mobile social services. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 246–255. ACM, 2009.

[6] Anna-Kaisa Pietiläinen, Earl Oliver, Jason LeBrun, George Varghese, and Christophe Diot. Mobiclique: middleware for mobile social networking. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 49–54. ACM, 2009.

[7] Rongxing Lu, Xiaodong Lin, Xiaohui Liang, and Xuemin Shen. A secure handshake scheme with symptoms-matching for mhealthcare social network. *Mobile Networks and Applications*, 16(6):683–694, 2011.

[8] Rui Zhang, Rui Zhang, Jinyuan Sun, and Uanhua Yan. Fine-grained private matching for proximity-based mobile social networking. In *INFOCOM, 2012 Proceedings IEEE*, pages 1969–1977. IEEE, 2012.

[9] Ming Li, Ning Cao, Shucheng Yu, and Wenjing Lou. Findu: Privacy-preserving personal profile matching in mobile social networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 2435–2443. IEEE, 2011.

[10] Fang Qi and Wenbo Wang. Efficient private matching scheme for friend information exchange. In *Algorithms and Architectures for Parallel Processing*, pages 492–503. Springer, 2015.

[11] Ben Niu, Tanran Zhang, Xiaoyan Zhu, Hui Li, and Zongqing Lu. Priority-aware private matching schemes for proximity-based mobile social networks. *arXiv preprint arXiv:1401.8064*, 2014.

[12] Gianpiero Costantino, Fabio Martinelli, and Paolo Santi. Privacy-preserving interest-casting in opportunistic networks. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 2829–2834. IEEE, 2012.

[13] Jin Li, Man Ho Au, Willy Susilo, Dongqing Xie, and Kui Ren. Attribute-based signature and its applications. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 60–69. ACM, 2010.

[14] Siamak F Shahandashti and Reihaneh Safavi-Naini. Threshold attribute-based signatures and their application to anonymous credential systems. In *Progress in Cryptology–AFRICACRYPT 2009*, pages 198–216. Springer, 2009.

[15] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98. Acm, 2006.

[16] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 321–334. IEEE, 2007.

[17] Jun Zhou, Zhenfu Cao, Xiaolei Dong, Xiaodong Lin, and Athanasios V Vasilakos. Securing m-healthcare social networks: Challenges, countermeasures and future directions. *Wireless Communications, IEEE*, 20(4):12–21, 2013.

[18] Carmit Hazay and Tomas Toft. Computationally secure pattern matching in the presence of malicious adversaries. In *Advances in Cryptology-ASIACRYPT 2010*, pages 195–212. Springer, 2010.

[19] Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In *Public Key Cryptography–PKC 2007*, pages 1–15. Springer, 2007.

[20] Xiaohui Liang, Zhenfu Cao, Jun Shao, and Huang Lin. Short group signature without random oracles. In *Information and Communications Security*, pages 69–82. Springer, 2007.