Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 857405, 15 pages
http://dx.doi.org/10.1155/2015/857405

Research Article

Towards Efficient, Secure, and Fine-Grained Access Control System in MSNs with Flexible Revocations

Shi-Feng Sun,[1] Chen Lyu,[1] Dawu Gu,[1] Yuanyuan Zhang,[1] and Yanli Ren[2]

[1] *Department of Computer Science & Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*
[2] *School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China*

Correspondence should be addressed to Dawu Gu; dwgu@sjtu.edu.cn

Received 31 October 2014; Accepted 5 January 2015

Academic Editor: Bin Xiao

With the pervasiveness of mobile communications, MSNs have become a promising networking paradigm for users to share contents with others through mobile devices. This convenience comes at the cost of some serious security and privacy issues. In this work, we propose a novel privacy-preserving scheme for MSNs, which can efficiently solve some of the most serious security and privacy issues such as data confidentiality, fine-grained access control, and flexible revocation. In particular, we leverage the attribute based encryption technique to realize fine-grained access control over encrypted data. Moreover, we enhance this technique and design a flexible and fine-grained revocation mechanism which enables not only efficient user revocation but also efficient attribute revocation. As we show, our system can achieve both forward secrecy and backward secrecy using such mechanism. We compare our scheme with other related works and show that not only most of the previous works suffer from larger size of encrypted data but also their decryption time grows linearly with the complexity of access policies. In comparison, our scheme achieves higher efficiency and smaller computation time while consuming lesser storage space. We provide extensive analysis and performance evaluation to demonstrate the security, scalability, and efficiency of our proposed framework.

1. Introduction

Explosive growth and popularity of social networks have made them an indispensable part of our daily life. Mobile social networks (MSNs) have not only allowed people to share content in real-time but also enabled a myriad of mobile applications such as location based services. This excitement around MSNs has led to many companies developing their own MSN solutions.

Despite of various appealing features, MSNs face some daunting challenges in terms of security and privacy. Some of these challenges are even more severe in MSNs compared to existing online social networks (OSNs). Considering the privacy concerns of MSNs, users are likely to be reluctant to reveal their personal profile, their content, and even their presence to others. On the other hand, malicious users are constantly looking for ways to get user's personal profile information and illegally selling the information. Of course, simply anonymizing user's name does not solve the problem because most such networks (e.g., Facebook) require the users to register with their real name. Recent privacy compromises such as [1, 2] have brought these issues to light, and it has become imperative to address the privacy concerns of MSNs.

One obvious way to tackle the privacy issues is to take advantage of well-known cryptographical tools. However, the problem with this is that it becomes increasingly unscalable for users to define their privacy settings. As an example, consider a user who wants to share her data with many possible subsets of her contacts such as her family or coworkers. She needs to encrypt multiple copies of the data using different group keys and needs to know the credentials of contacts to whom she will give the access privileges. Also, any such method cannot realize flexible access control on the encrypted data. Even though some traditional public key encryption schemes [3, 4], which can realize fine-grained access control, have been proposed, multiple copies are still required to encrypt for different groups, which incurs high key management overhead.

To address the issues mentioned above, a one-to-many kind of encryption method called attribute-based encryption (ABE) was introduced by Sahai and Waters [5]. This scheme can be used to improve the scalability of the traditional solutions. Since the introduction of ABE, many of research works [6–11] have proposed many extensions to extend its functionality and refine its security. The ABE scheme is also widely adopted to secure MSN communications, cloud storage, and computing applications [12–16].

However, one of the main drawbacks of those ABE schemes used in most of current architectures such as social networks, sensor networks, data outsourcing, and e-healthy systems [12, 13, 15, 17, 18] is low efficiency. Specifically, both the ciphertext size and decryption time grow with the size of the access formula. The ABE systems are set in pairing-based groups where the ciphertext requires two group elements while the decryption requires a pairing computation for each node of the formula. Although such a task for a typical formula size can be relatively easily handled by conventional desktop computers, this still presents a big challenge for mobile devices used in MSNs. Processors of current mobile devices are one or two orders slower than the desktop computers, and the problem is further aggravated due to limited battery life of such devices. Fortunately, emerging cloud service providers (e.g., Microsoft's Azure, Amazon's EC2, etc.) have made on-demand computing and storage a reality. We propose to use such third-party in our framework to transform the encrypted data into partial data containing only two group elements. As we show, this *greatly reduces* the storage requirement and computation cost for the users with mobile devices. Moreover, using such third-party cloud infrastructure offloads the cost of building and maintaining data centers from the MSN operators.

Furthermore, the underlying ABE schemes used in most of the previous works [12–16] are not secure enough. In particular, they are only proved secure in the generic group model and so just achieve heuristic security. Thus, the applications based on these ABE schemes, such as cloud storage and e-healthy system above, are not secure enough as well. Just as indicated by [15], the systems with stronger security guarantees are desirable. In contrast to the previous works, we propose our data privacy-preserving system based on the scheme in [11] and achieve the desired *stronger security*.

In addition to the above issues, the need for access control mechanisms supporting dynamic groups and efficient revocations is also motivated. This is because groups in MSNs are often dynamic and credentials or attributes may change over time or due to someone's malicious behaviors or changes in relationships with a contact or in work environments. In previous works, although most of them realized attribute revocation, the revocation was neither flexible nor fine-grained. That is, the revocation of an attribute needs to reencrypt the encrypted data, and the revocation will result in the revocation of all users with this attribute. In our system, however, the attribute revocation is *flexible and fine-grained*. Specifically, the data owner can differentiate the malicious users from the other ones with the same attribute (the data owner wants to revoke), and the original encrypted data need not be reencrypted. Furthermore, besides forward secrecy our system also can achieve *backward secrecy* compared to most of the previous works, the notion of which is described in the following section.

In light of these discussions, it is imperative to address these privacy and efficiency concerns before mobile social networks can be securely and widely adopted. To meet all these needs simultaneously, we put forward two efficient and secure data privacy-preserving schemes for MSNs based on the ABE scheme [11], which can be proved secure in the standard model rather than the generic group model. In particular, the proposed schemes not only can protect the privacy of users' personal data and achieve stronger security guarantees but also can efficiently realize fine-grained access control over the encrypted data. In other words, users in MSNs can formulate flexible access policies over attributes of their contacts by themselves, instead of trusting social network providers and depending on them to enforce privacy protection. This enables users to selectively share their private data without knowing complete lists of contacts in advance. Meanwhile, the schemes can realize flexible and fine-grained revocations, achieving fine-grained access control even for dynamic groups. Moreover, inspired by Green et al.'s work [19], we leverage a third party (e.g., a cloud service provider) to transform the encrypted data to the partial one and thus greatly reduce the storage and computation cost for the data accessors, especially for the mobile users.

In a nutshell, we mainly make the following contributions in this work. (i) We present a novel ABE-based architecture for mobile social networks by introducing a minimally trusted party, which would provide not only storage service but also computation service. Under this new setting, *data privacy and fine-grained access control* can be efficiently and flexibly implemented, and a myriad of applications for mobile devices can be widely adopted. (ii) We put forward two secure and practical data privacy-preserving schemes with efficient and flexible revocations based on Waters' ABE scheme, one for immediate user revocation and the other for immediate attribute revocation, both of which are quite suitable for applications on mobile devices due to their *efficiency*. Moreover, our schemes can achieve *stronger security* and the revocation is *not only flexible but also fine-grained*, compared to most of the previous works. That is, whenever a user revokes a contact or an attribute in our schemes, she needs neither to renew the keys of the remaining contacts of the group, nor to reencrypt the previously stored data for preventing the revoked contacts from accessing her private data, and the revocation can *differentiate* the malicious users from the other ones with the same revoked attribute. Additionally, these schemes can resist collusion attacks and achieve *forward secrecy and backward secrecy*. (iii) We give a thorough analysis of security, complexity, and scalability of our proposed approaches, compare our schemes with several related ones with regard to storage, communication, and computation cost, and furthermore demonstrate the effectiveness of the proposed solutions through implementations.

The rest of paper is organized as follows. Section 2 describes the related work. The system architecture, threat model, and design goals of our work are described in Section 3. Section 4 introduces some preliminary knowledge of cryptosystem used in this work. The detailed solutions are proposed in Section 5, followed by the security analysis of our proposed schemes in Section 6. The efficiency analysis and the comparison with several representative works are presented in Section 7, and in Section 8, we briefly conclude this paper.

2. Related Work

With the pervasiveness of mobile devices, the popularity of mobile social networks (MSNs) has increased rapidly. Pietilainen et al. [20] put forward a middleware named MobiClique, which allows mobile phone users to connect to others over ad hoc networks to exchange social network identity information and forward messages. All users are assumed to be trusted, and both privacy and security are ignored in this middleware.

However, there exist great many malicious users in our real life. Privacy and security issues have become the users' main concerns in MSNs. Recently, more and more researchers have started to pay more attention to these issues.

FindU [21, 22] mainly focused on the private profile matching problem. Leveraging secure multiparty computation and private set-intersection technique, Li et al. [21] proposed three different privacy-level profile matching protocols. Subsequently, [22] presented a suite of novel fine-grained private matching protocols based on Paillier cryptosystem [23] and [24] gave a novel solution for secure friend discovery in MSNs.

In addition, the data privacy and access control in the MSNs is also a challenging problem, which is much severer than in OSNs. Recently, several privacy-preserving architectures [4, 15, 25–28] for OSNs have been proposed based on the cryptographic tools.

In FlyByNight [25], users can securely communicate with each other by encrypting sensitive messages using JavaScript on the client side. It relies on the OSN provider for key management, and so it is vulnerable to active attacks launched by the provider. A more challenging issue is the revocation in FlyByNight, where rekeying is required for the owner's remaining friends. The architecture NOYB [27] uses an encryption technique and dictionaries to provide privacy. It has some limitations such as having no option for flexible classification of user's friends and no efficient revocation mechanism. FaceCloak [26] has similar goals, but it achieves these goals by storing the encrypted data on a third-party server and fake data on the OSN provider. In this architecture, fine-grained access control is not achieved, the owner's friends having the same access privilege.

Combining attribute base encryption and traditional public key cryptography, Baden et al. [4] proposed Persona, which allowed the data owner to flexibly manage the memberships of his friends and realized the fine-grained access control on her encrypted data. As referred by [28], the downsides of this work are the lack of an efficient membership revocation mechanism and the high computational cost associated with the ABE operations. In [28], Sun et al. proposed a new and efficient framework based on the broadcast encryption [29] and searchable public key encryption [30]. This framework can achieve efficient revocation, but it cannot achieve backward secrecy. It also cannot realize the fine-grained access control over her contacts, because each one of her contacts has only one role in this framework. Thus, if the owner wants to share her data with multiple groups, she has to produce multiple copies of the encrypted data.

Recently, Jahid et al. [15] also presented an ABE-based access control mechanism EASiER for OSNs. This paradigm can provide efficient and immediate user/attribute revocation, thereby avoiding the overhead of rekeying with group members and reencryption of old data. However, most of these works such as [13, 15], except for [17], are all based on the not secure enough ABE [6], which only can be proved secure in the generic group model. In [17], they proposed a secure and efficient user revocation scheme for MSNs, where the mobile user's data decryption capability is controlled by a trusted authority, and it only can provide a solution with regard to user revocation. Additionally, besides [4, 13], they still suffer from the drawback of expensive computational cost associated with ABE operations; that is, both the ciphertext size and the computational cost for the mobile social users grow linearly with the size of the access formula.

In this study, we tackle the problems above and put forward an efficient and secure data privacy-preserving solution for mobile social networks, which realizes fine-grained access control over the encrypted data, flexible and immediate user/attribute revocation, and satisfies the main security requirements, such as collusion resistance and forward and backward secrecy.

3. System and Threat Models

In this section, we first present the system architecture and threat model and then define our design goals in the context of mobile social networks.

3.1. System Architecture. By introducing a minimally trusted service provider, a novel system architecture for MSNs is put forward in this section. The service provider in this system has great storage and computing resources, which provides not only on-demand storage service but also on-demand computing service for others. This architecture mainly consists of the following four entities, the associated interactions among which and the topology are illustrated in Figure 1.

(i) Credential Authority. This entity is in charge of cryptographically initialization of an MSN domain that consists of a credential authority and all the registered mobile users and issues legitimate credentials with the corresponding private/public key pair to each user. These key pairs can be used to perform such security operations as authentication in this domain.

(ii) Service Provider. It is a third-party provider that offers the on-demand storage and computing services for possibly multiple MSN applications or domains. The service provider is not fully trusted by users in the domain. It is in charge

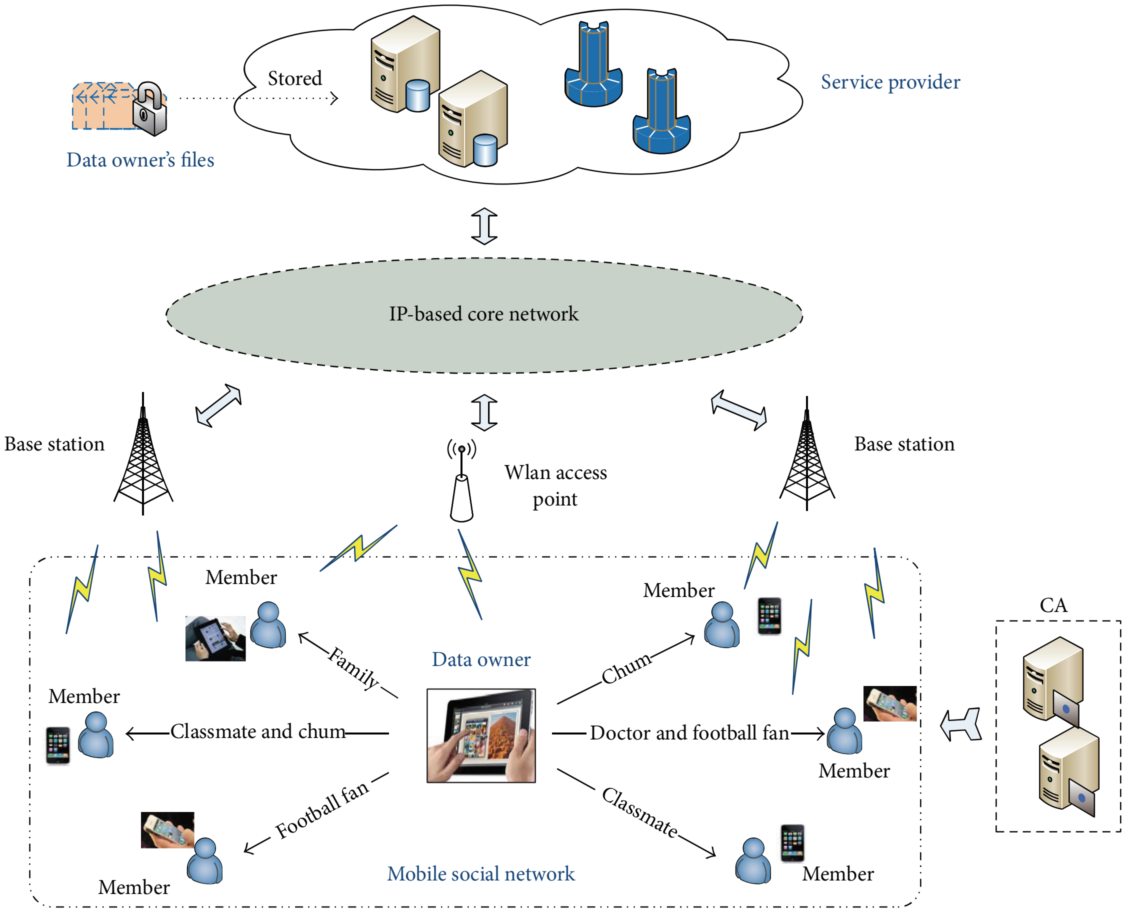


Figure 1: System architecture.

of computing the transformed data and providing corresponding contents to the accessor. Similar to the previous assumptions in [28, 31, 32], we assume that the service provider is honest-but-curious; that is, it will try to find out as much secret information from the outsourced data as possible, but it will honestly execute the tasks assigned by legitimate parties in the system.

(iii) Data Owner. This entity is a MSN user who wishes to share her private data with her contacts. In this system, she can define the access policy by herself based on the contacts' attributes and enforce it by encrypting her own data under the policy before outsourcing it to the service provider.

(iv) Member. This entity is a MSN user and one or more data owners' contact, who have many different attributes such as Ph.D., classmate, and football fan. According to his different attributes, he might belong to one data owner's different attribute groups. If he wants to access one data owner's personal or private data, the attribute set he possesses or that after the revocation must satisfy the access policy of the encrypted data. Meanwhile, the member is the data owner of his own group.

3.2. Threat Model. In this section, we consider some mainly potential attackers and their attacks to our proposed schemes in a MSN domain. The *service provider*, assumed to be honest-but-curious, will attempt to compromise data privacy by learning the content of the private data, but he will not maliciously delete or modify user data and honestly execute all the tasks assigned by legitimate parties. As to some *unauthorized users or members* whose attributes do not satisfy the access policy, they may also try to access the data beyond their access privileges. Furthermore, some users may collude together with other users or even the service provider to compromise some data owner's privacy, which is usually called collusion attacks. According to the colluding parties, we classify it as the following three meaningful types.

(1) Attribute Collusion Attack. It is launched by a group of nonrevoked yet unauthorized users. Given a target ciphertext, they aim to decrypt it in collaboration in that each of them is unable to decrypt it individually.

(2) Revocation and Attribute Collusion Attack. This kind of attack is launched by the revoked yet authorized users and the unrevoked yet unauthorized users. The former refers to

the users who can decrypt some ciphertexts before they or some of their attributes are revoked but will be unable to decrypt the encrypted data under the same policy ever after the revocation. The other means that they cannot decrypt the encrypted data under which policy their attributes do not satisfy even though they are not revoked.

(3) Revocation and Provider Collusion Attack. It is launched by the authorized yet revoked user and the service provider. The user's attributes satisfy the access policy enforced on the encrypted data. After some of his attributes or the user himself are revoked, he attempts to collude together with the provider to violate the data owner's privacy.

3.3. Design Goals. Based on the discussions and attacks above, the design goals with respect to security and performance are described as follows.

(i) Data Confidentiality. Unauthorized users whose attributes do not satisfy the access policy should be prevented from learning the content of the private data encrypted under the policy. Additionally, the revoked users should also be prevented from accessing the private data, except that the remaining attributes still satisfy the access policy for the attribute revocation case.

(ii) Collusion Resistance. For a group of users who cannot decrypt a given ciphertext alone, they still cannot decrypt the ciphertext even though they collude together by combining their attributes. In the case where the revoked users collude with the service provider, they cannot decrypt the ciphertext as well.

(iii) Forward and Backward Secrecy. In MSNs, when the data owner finds some user's malicious behaviors, she may decide to revoke this user or some of his attributes to prevent him from accessing the plaintext of subsequent data uploaded afterwards (unless his remaining attributes still satisfy the access policy), which we call forward secrecy. Moreover, the data owner may want to prevent him from accessing the previous data that are encrypted under the access policy satisfied by his previous attribute set and he did not access before (unless his remaining attributes satisfy the access policy), which we refer to as backward secrecy in our context. The data owner may encrypt many files under the same policy, and the user typically accesses the data on the fly due to the storage space. Therefore, the backward secrecy is important for the owners' privacy protection.

(iv) Fine-Grained Access Control. The access policy can be made flexibly by the data owner herself based on her contacts' attributes. Moreover, the contacts' access privileges on her private data can be easily controlled by the data owner, and the unauthorized users should not be able to access the private data.

(v) Flexible and Immediate Revocation. Once one user or his some attribute(s) is revoked due to his malicious behaviors, the decryption capability of this user should be revoked without affecting any other user's decryption capability. Furthermore, for the users having the same attributes, the revocation should be able to differentiate these users and only be applied to the malicious one.

(vi) Scalability and Efficiency. The proposed solution should support dynamic groups and have high scalability with respect to low cost on storage, communication, and computation, especially for the mobile users in this system.

4. Preliminaries

In this part, we will give a brief introduction to some preliminaries used throughout the paper, mainly including linear secret sharing schemes (LSSS), ciphertext-policy attribute-based encryption (CP-ABE) schemes, and revocation schemes.

4.1. Linear Secret Sharing Scheme. We will give the notion of linear secret sharing scheme in the following, which is adapted from the definitions in [33]. Firstly, we describe the definition of access structure that will be used in this notion.

Definition 1 (access structure [33]). Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}}$ is monotone if for all B, C: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An access structure (resp., monotone access structure) is a collection (resp., monotone collection) $\mathbb{A}$ of nonempty subsets of $\{P_1, P_2, \ldots, P_n\}$; that is, $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

In our context, the role of the parties is taken by the attributes. Correspondingly, the contact whose attribute set S belongs to $\mathbb{A}$ is called the authorized user, and the one not in $\mathbb{A}$ is called the unauthorized user.

Definition 2 (linear secret-sharing scheme (LSSS)). A secret-sharing scheme Π over a set of parties $\mathscr{P}$ is called linear (over Z_p) if

(1) the shares for each party in $\mathscr{P}$ form a vector over Z_p;

(2) there exists a matrix M with l rows and n columns called the share-generating matrix for Π. For all $i = 1, \ldots, l$, we let $\rho(i)$ denote the party corresponding to row i, where ρ is the function mapping each row of the matrix M to one associated party. When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in Z_p$ is the secret to be shared, and $r_2, \ldots, r_n \in Z_p$ are chosen at random, then Mv is the vector of l shares of the secret s according to the scheme Π. The share $(Mv)_i$ belongs to party $\rho(i)$.

It is shown in [9] that, according to the above definition, every linear secret sharing scheme also enjoys the *linear reconstruction* property, defined as follows: suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set: $I \subset \{1, 2, \ldots, l\}$ being defined as $I = \{i : \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in Z_p\}_{i \in I}$ that can be found in time polynomial in the size of the share-generating matrix

M, such that $\Sigma_{i\in I}\omega_i\lambda_i = s$ if $\{\lambda_i\}$ are valid shares of any secret s according to Π.

4.2. Ciphertext-Policy ABE. Attribute-based encryption is a promising primitive for many applications, in which the provider may want to share date with some users just according to some policy. Compared with the traditional public key encryption, it can be used to implement encryption to groups and realize fine-grained access control over encrypted data. In this setting, a user's private key is always associated with an attribute set (or an access structure), and a ciphertext is associated with an access policy (or an attribute set); she can decrypt a ciphertext if and only if the attribute set corresponding to her private key (or the ciphertext) satisfies the access policy corresponding to the ciphertext (or the private key). The first case is usually referred to as ciphertext-policy attribute-based encryption (CP-ABE) and the other key-policy attribute-based encryption (KP-ABE). Since the data owner usually prefers to define the access policy by herself, the CP-ABE is much more appropriate than KP-ABE for many applications.

4.3. Revocation Scheme. A revocation scheme is a basic tool for designing encryption schemes for mass distribution of data. Using this scheme, a group controller can use it to revoke the keys of users who leak their personal keys for some illegal purposes. The scheme usually consists of the following two algorithms [34].

(i) Initialization. The group controller generates a random polynomial P of degree t over Z_p (this polynomial can be used for $(t+1)$-out-of-n secret sharing), and then it provides each user u with a personal key $\langle I_u, P(I_u)\rangle$ through a private channel, where I_u is the user's identity.

(ii) Revocation. The group controller decides to revoke the t users $I_{u_1}, I_{u_2}, \ldots, I_{u_t}$, and it broadcasts the identities and the personal keys of these users: $\langle I_{u_i}, P(I_{u_i})\rangle_{i\in[t]}$. Each non-revoked user u can combine these t keys with its personal key and use these $t+1$ shares to interpolate P and compute $P(0)$.

If the group controller prepares a scheme to revoke t users, and only $t' < t$ users need to be removed, then the revocation can be performed by sending the shares of these t' users and additional $t - t'$ values of P at locations that are different from any other user's identity I_u.

5. Fine-Grained Access Control System with Efficient and Flexible Revocation

5.1. Overview of Our Proposed Schemes. In order to achieve our design goals, we leverage the promising primitive ABE as the basic cryptographic tool and combine the efficient revocation and outsourcing techniques to tackle the focusing issues on efficiency, fine-gained access control, and immediate revocation. First, we present a secure and efficient scheme with user revocation, and then based on this basic scheme, we solve some nontrivial problems and put forward an advanced scheme with any attribute revocation.

In our basic solution, when a data owner wants to share her personal data according to the access policy made by herself, especially those data revealing her private information and personal life, she first initializes the system and generates the private key $sk = (\gamma, tk)$ for each of her contacts. Then she encrypts the data based on the policy she made and uploads it to the service provider. For the purpose of user revocation, she creates the revocation list RL, generates the corresponding revocation keys rk, and sends them to the service provider who will help her enforce access policy using these revocation keys. Whenever a member or contact wants to access the owner's data, he first sends his transformation key tk to the service provider, which is part of his private key sk. Actually, he just needs to send it for one time, unless his private key is regenerated. Then the provider checks the revocation key, transforms the encrypted data CT to CT', and transfers it to the accessor, provided that he is not revoked and his attribute set S satisfies the access structure $\mathbb{A}$. Receiving the transformed ciphertext CT', the member can efficiently retrieve the plaintext using his secret value γ by executing only one exponent operation. Based on the basic scheme, our advanced scheme goes one step further by providing the attribute revocation. Its procedure is similar to the basic one, but we must try to solve some difficulties encountered in the concrete construction.

5.2. Concrete Construction of the Basic Scheme. At first, we present the basic scheme with a complete key revocation. This scheme is described as follows.

5.2.1. System Initialization. To initialize the system, the data owner runs the following steps:

(1) setting the security parameter κ and attribute universe description $U = \{1, 2, \ldots, |U|\}$ and choosing two multiplicative cyclic groups G and G_T of prime order p with an admissible bilinear map $e : G \times G \rightarrow G_T$ and a hash function $H : \{0,1\}^* \rightarrow G$;

(2) randomly choosing $\alpha, \beta, a \in Z_p$ and a polynomial P of degree d (the maximum number of revoked users at a time) over Z_p such that $P(0) = \beta$. The public parameters are published as

$$\text{params} = \left\{G, g, e(g,g)^{\alpha}, g^{a}, H\right\}, \quad (1)$$

where g is a generator of G. The master secret key is set as $msk = (g^{\alpha}, \beta, P)$.

5.2.2. Key Generation. In this section, the data owner generates her user's private key and the revocation key in the following steps.

(1) Take as input the master secret key $msk = (g^{\alpha}, \beta, P)$ and an attribute set S belonging to the user u_k, and choose $t, \gamma \in Z_p^*$ at random. Then output u_k's transformation key tk as follows:

$$K = g^{(\alpha/\gamma)} g^{at\beta}, \qquad L = g^{t\beta},$$
$$\left\{K_x = H(x)^{t}\right\}_{\forall x\in S}, \qquad \left\{K'_x = H(x)^{tP(u_k)}\right\}_{\forall x\in S}. \quad (2)$$

The u_k's private key sk consists of both the transformation key tk and the randomly chosen secret value γ; that is, $sk = (\gamma, tk) = (\gamma, (K, L, \{K_x, K'_x\}_{\forall x \in S}))$.

(2) Take as input the master secret key msk and the revocation list RL, where RL is a list of social contacts with identities u_i, $i \in \{1, \ldots, d\}$ whose keys the data owner decides to revoke and create the revocation key rk as follows:

$$rk = \left(\langle u_i, P(u_i) \rangle_{\forall u_i \in RL} \right). \tag{3}$$

Note that in the case of the number $|RL|$ of revocations less than d, data owner creates $|RL|$ shares and additional $d - |RL|$ values of P, at locations that are different from any other user's identity.

Then, the data owner sends the private key sk to the user u_k through a secure communication channel and updates the revocation key rk to the service provider. Similar to the assumption in previous works such as [15, 18, 31], the service provider just keeps the latest revocation key in its memory, while erasing all the previous ones.

5.2.3. Data Storage. During this part, the data owner encrypts her private data as follows.

(1) Take as input $params$, a message m, and an access structure (M, ρ), where M denotes a matrix with l rows and n columns and the function ρ associates rows of M to attribute universe U.

(2) First choose a random vector $v = (s, y_2, \ldots, y_n) \in Z_p^n$ where the values $y_2, \ldots, y_n$ will be used to share the encryption exponent s, and then calculate $\lambda_i = vM_i$ for $i = 1$ to l, where M_i is the i row of M.

(3) Randomly choose $r_1, \ldots, r_l \in Z_n$ at first, and then compute the ciphertext CT as follows:

$$\begin{gathered} C = m \cdot e(g, g)^{\alpha s}, \qquad C' = g^s, \\ \left(C_i = g^{a \lambda_i} H(\rho(i))^{-r_i},\ D_i = g^{r_i} \right)_{\forall i \in \{1, \ldots, l\}}. \end{gathered} \tag{4}$$

Eventually, the data owner uploads the encrypted data $CT = ((M, \rho), C, C', (C_i, D_i)_{\forall i \in \{1, \ldots, l\}})$ to the service provider.

5.2.4. Data Access. When some user u_k wants to access the owner's data, he sends the request message and transformation key tk to the service provider. Then, the service provider transforms the encrypted data CT to CT'. The process is simply shown in Figure 2, and the details are described as follows.

(1) Take as input rk for a revocation list RL, tk for an attribute set S, and u_k and CT for an access structure (M, ρ), and output $\perp$ if S does not satisfy the access structure (M, ρ) or u_k belongs to the revocation list RL associated with rk. Otherwise, calculate D'_i:

$$D'_i = D_i^{\sum_{j=1}^{t} \mu_j P(u_j)}, \quad \forall i \in \{1, \ldots, l\}, \tag{5}$$

where $\mu_j = (u_k/(u_k - u_j)) \prod_{n \neq j} (u_n/(u_n - u_j))$, for all $j, n \in \{1, \ldots, t\}$, $k \notin \{1, \ldots, t\}$, and then compute μ_k:

$$\mu_k = \prod_{n \neq k} \frac{u_n}{u_n - u_k}, \quad \forall n \in \{1, \ldots, t\},\ k \notin \{1, \ldots, t\}. \tag{6}$$

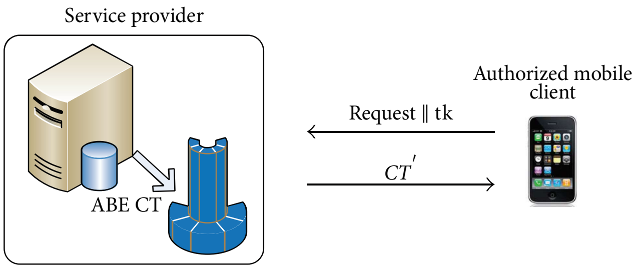


Figure 2: Data access process.

(2) Terminate the transformation process if the output in step (1) is $\perp$. Otherwise, let $I \subset \{1, \ldots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$ and let $\{\omega_i \in Z_p\}_{i \in I}$ be a set of constants, such that $\sum_{i \in I} \omega_i \lambda_i = s$ if $\{\lambda_i\}$ are valid shares of any secret s according to M (note that there could be different ways to choose the values ω_i to satisfy this). Then, calculate TC_1 as follows:

$$\begin{aligned} TC_1 &= \frac{e(C', K)}{\prod_{i \in I} \left(e(L, C_i)\, e\left(D_i, K'_{\rho(i)}\right)^{\mu_k} e\left(D'_i, K_{\rho(i)}\right) \right)^{\omega_i}} \\ &= \frac{e(C', K)}{\prod_{i \in I} \left(e(g, g)^{at\lambda_i \beta} e(g, H(\rho(i)))^{-tr_i \beta} \right)^{\omega_i}} \\ &\quad \cdot \frac{1}{\prod_{i \in I} \left(e(g, H(\rho(i)))^{tr_i \mu_k P(u_k) + tr_i \sum_{j=1}^{t} \mu_j P(u_j)} \right)^{\omega_i}} \\ &= \frac{e(C', K)}{\prod_{i \in I} e(g, g)^{at\lambda_i \omega_i \beta}} = \frac{e(g, g)^{(\alpha/\gamma)s} e(g, g)^{at\beta s}}{e(g, g)^{at\beta \sum_{i \in I} \lambda_i \omega_i}} \\ &= e(g, g)^{\alpha s/\gamma}. \end{aligned} \tag{7}$$

At last, the service provider sends back the transformed ciphertext $CT' = (TC_0 = C,\ TC_1 = e(g, g)^{\alpha s/\gamma})$ to the user u_k. Note that, the user's transformation key does not need to be transmitted once again, until it is regenerated.

5.2.5. Data Decryption. Receiving the transformed ciphertext CT', the authorized user u_k can easily retrieve the owner's data in the following way.

(1) Take as input the transformed ciphertext CT' and his private key sk associated with an attribute set S which satisfies the access structure enforced on the encrypted data.

(2) Retrieve the message m by simply computing

$$\frac{TC_0}{TC_1^{\gamma}} = \frac{m \cdot e(g,g)^{\alpha s}}{e(g,g)^{\alpha s}} = m. \tag{8}$$

5.3. Concrete Construction of the Advanced Scheme. The basic scheme presented above can only achieve user revocation as the schemes proposed in [15, 17, 28], which lack revocation flexibility. That is, when a user is revoked, he loses all the access rights to the data even if his attribute set satisfies the access policy. To enhance the granularity of revocation level, we put forward an advanced scheme with any attribute revocation, which is described as follows. For brevity, we just present the parts that are different from our basic scheme.

5.3.1. System Initialization. On initialization, the data owner executes the following steps.

(1) This step is the same as that in our basic scheme described in Section 5.2.1.

(2) This is almost the same as step (2) of Section 5.2.1 in our basic scheme; except that, instead of choosing one polynomial P of degree d, choose a polynomial P_x of degree t_x over Z_p for each attribute x such that $P_x(0) = \beta$, where t_x is the maximum number of revoked users for attribute x at a time. The public parameters are published as

$$\text{params} = \left\{G, g, e(g,g)^{\alpha}, g^{a}, H\right\}, \tag{9}$$

and the master secret key is set as $msk = (g^{\alpha}, \beta, \{P_x\}_{\forall x})$.

5.3.2. Key Generation. In this part, the data owner generates her user's private key and the revocation key as follows.

(1) The private key generation process for user u_k is similar to step (1) of Section 5.2.2 in our basic scheme. Take as input the msk and calculate u_k's transformation key tk as follows:

$$\begin{gathered} K = g^{(\alpha/\gamma)} g^{at\beta}, \qquad L = g^{t\beta}, \\ \left\{K_x = H(x)^{t}\right\}_{\forall x \in S}, \qquad \left\{K'_x = H(x)^{tP_x(u_k)}\right\}_{\forall x \in S}. \end{gathered} \tag{10}$$

Then, set $sk = (\gamma, tk) = (\gamma, (K, L, \{K_x, K'_x\}_{\forall x \in S}))$.

(2) Take as input msk and $\{RL_x\}_{\forall x \in U'}$, where $U' \subset U$ is the set of attributes that the data owner decides to revoke and RL_x is a list of social contacts $\{u_1, \ldots, u_{t_x}\}$ whose attribute x will be revoked by the data owner. Then, create the revocation key $rk = (rk_1, rk_2)$ as follows:

$$\begin{aligned} rk_1 &= \left\{\langle u_i, P_x(u_i)\rangle_{\forall u_i \in RL_x}\right\}_{\forall x \in U'}, \\ rk_2 &= \left\{\langle x_i, P_x(x_i)\rangle_{\forall i \in \{1,\ldots,t_x\}}\right\}_{\forall x \in U - U'}, \end{aligned} \tag{11}$$

where $\{x_i\}$ are chosen from Z_p and they are different from any user's identity.

Finally, the data owner sends the private key sk to the user u_k through a secure communication channel and updates the revocation key rk to the service provider.

5.3.3. Data Storage. This process is the same as that in the basic scheme, which is presented in Section 5.2.3.

5.3.4. Data Access. When wanting to access the owner's data, the user u_k sends the request message and his transformation key tk to the service provider. Then, the service provider transforms the encrypted data CT to CT' as follows.

(1) Take as input rk, tk, u_k, and CT and output $\perp$ if u_k's attribute set S' after revocation does not satisfy the access structure (M, ρ). Otherwise, let $I' \subset \{1, \ldots, l\}$ be defined as $I' = \{i : \rho(i) \in S'\}$ and let $\{\omega'_i \in Z_p\}_{i \in I'}$ be a set of constants, such that $\sum_{i \in I'} \omega'_i \lambda_i = s$ if $\{\lambda_i\}$ are valid shares of any secret s according to M. First, calculate D'_i:

$$D'_i = D_i^{\sum_{j=1}^{t_{\rho(i)}} \mu_{\rho(i),j} P_{\rho(i)}(u_j)}, \quad \forall i \in I', \tag{12}$$

where $\mu_{\rho(i),j} = (u_k/(u_k - u_j)) \prod_{n \neq j} (u_n/(u_n - u_j))$, for all $j, n \in \{1, \ldots, t_{\rho(i)}\}, k \notin \{1, \ldots, t_{\rho(i)}\}$, and then compute $\mu_{\rho(i),k}$:

$$\mu_{\rho(i),k} = \prod_{n \neq k} \frac{u_n}{u_n - u_k}, \quad \forall n \in \left\{1, \ldots, t_{\rho(i)}\right\}, \tag{13}$$

where $k \notin \{1, \ldots, t_{\rho(i)}\}$.

(2) Terminate the transformation process if the output in step (1) is $\perp$. Otherwise, calculate TC_1 as follows:

$$\begin{aligned} TC_1 &= \frac{e(C', K)}{\prod_{i \in I'} \left(e(L, C_i)\, e\left(D_i, K'_{\rho(i)}\right)^{\mu_{\rho(i),k}} e\left(D'_i, K_{\rho(i)}\right)\right)^{\omega'_i}} \\ &= \frac{e(C', K)}{\prod_{i \in I'} \left(e(g,g)^{at\lambda_i\beta} e(g, H(\rho(i)))^{-tr_i\beta}\right)^{\omega'_i}} \\ &\quad \cdot \frac{1}{\prod_{i \in I'} \left(e(g, H(\rho(i)))^{tr_i \mu_{\rho(i),k} P_{\rho(i)}(u_k)}\right)^{\omega'_i}} \\ &\quad \cdot \frac{1}{\prod_{i \in I'} \left(e(g, H(\rho(i)))^{tr_i \sum_{j=1}^{t_{\rho(i)}} \mu_{\rho(i),j} P_{\rho(i)}(u_j)}\right)^{\omega'_i}} \\ &= \frac{e(C', K)}{\prod_{i \in I'} e(g,g)^{at\lambda_i\omega'_i\beta}} = \frac{e(g,g)^{(\alpha/\gamma)s} e(g,g)^{at\beta s}}{e(g,g)^{at\beta \sum_{i \in I'} \lambda_i \omega'_i}} \\ &= e(g,g)^{\alpha s/\gamma}. \end{aligned} \tag{14}$$

At last, the service provider sends back the transformed ciphertext $CT' = (TC_0 = C, TC_1 = e(g,g)^{\alpha s/\gamma})$ to the user u_k. Note that, the user's transformation key does not need to be transmitted once again, until it is regenerated.

5.3.5. Data Decryption. This is the same as the corresponding part in our basic scheme presented in Section 5.2.5.

6. Security Analysis

In this section, we analyze the security properties of our proposed solution in MSNs with regard to data confidentiality, collusion resistance, and forward and backward secrecy, which are proved in the following three theorems.

6.1. Data Confidentiality. In our context, data privacy is one primary security goal which is to keep the owners' personal data confidential with regard to unauthorized users and service provider. In the following, we show that our proposed solution achieves this goal.

Theorem 3. *The proposed schemes guarantee data confidentiality of the owners' personal data against unauthorized users and the curious service provider.*

Proof. We present the detailed analysis of this property as follows. For the unauthorized user whose attribute set S cannot satisfy the access policy enforced on the ciphertext, he cannot recover the desired value $e(g,g)^{\alpha s}$, which is required for the decryption in both the user revocation and the attribute revocation case. In fact, when trying to retrieve $e(g,g)^{at\beta s}$ from pieces $e(g,g)^{at\lambda_i\beta}$ during the decryption process, the user is required to have the private key that is associated with an attribute set satisfying the access structure (M,ρ). On the other hand, when a user is revoked, he cannot decrypt the ciphertext in the user revocation case. This is because his private key in this case is completely revoked; that is, $e(g,H(\rho(i)))^{tr_i\beta}$ cannot be retrieved using Lagrange interpolation for every attribute $\rho(i)$, $i \in I$. For the attribute revocation case, where partial private key corresponding to some attributes that satisfy the access policy is revoked, he also cannot decrypt the ciphertext (unless the rest of his attributes still satisfy the policy). This is because $e(g,H(\rho(i)))^{tr_i\beta}$ cannot be obtained for the revoked attributes $\rho(i)$, $i \in I'$. Hence, the user in both cases cannot obtain the required value $e(g,g)^{\alpha s}$.

As to the service provider, it is not totally trusted by the users in the MSN domain. Even though it possesses the revocation keys, it cannot decrypt any ciphertext either, since any of private keys for the set of attributes is not given to the service provider from the data owner in MSNs. On the other hand, even if the service provider can help some user transform the ciphertext CT to CT' and obtain $e(g,g)^{\alpha s/\gamma}$ using the user's transformation key tk, he still cannot decrypt the ciphertext, because he does not know the secret value γ. Therefore, data confidentiality against the service provider is also guaranteed. □

6.2. Collusion Resistance. In the following, we will give a detailed proof for the collusion-resistant property of our proposed schemes.

Theorem 4. *The proposed schemes are collusion-resistant against colluders, including the users and the service provider.*

Proof. The proposed schemes are collusion-resistant against the three meaningful collusion attacks considered in the threat model. We will show how they resist the three main attacks in detail as below. For the attribute collusion attack launched by multiple users who cannot decrypt the ciphertext alone, $e(g,g)^{at\lambda_i\beta}$ can be retrieved by the colluding users who endeavor to recover the required value $e(g,g)^{at\beta s}$ for decryption process. Obviously, it is easy to obtain $e(g,g)^{as}$ with enough shares $e(g,g)^{a\lambda_i}$ of the exponent s according to the *linear reconstruction* property of LSSS. However, the value t is a random and unique exponent for each user, so the reconstruction of s is prevented by the distinct exponents. Hence, attribute collusion attack can be precluded in the proposed schemes.

Next, we consider the revocation and attribute collusion attack launched by the revoked yet authorized user and the unrevoked yet unauthorized user, which are described as in the threat model. In this attack, although the former's attribute set S satisfies the access policy under which the given ciphertext is encrypted, he still cannot decrypt the ciphertext. This is because his private key is completely revoked (for the user revocation case). Actually, since he cannot get the coefficient μ_k and D_i' from the service provider, he has no means to obtain $e(g,H(\rho(i)))^{tr_i\beta}$. Thus, he cannot retrieve $e(g,g)^{at\lambda_i\beta}$ and consequently cannot reconstruct the required value $e(g,g)^{at\beta s}$ for decryption. Even though he colludes with the latter who can obtain $e(g,H(\rho(i)))^{tr_i\beta}$ for some attribute $\rho(i)$, the former still cannot leverage this value to retrieve $e(g,g)^{at\lambda_i\beta}$, since t is a random and unique exponent for each user. Furthermore, the latter can only obtain finite $e(g,H(\rho(i)))^{tr_i\beta}$ for his attribute set which does not satisfy the policy. Due to the similar reasons, the former cannot help the latter retrieve $e(g,g)^{at\beta s}$ as well. Therefore, even though they collude together, they are still incapable of decrypting the ciphertext. As to the attribute case, the analysis is similar.

At last, we analyze the collusion-resistant property against revocation and provider attack. In this attack, the authorized yet revoked user u_k aims at decrypting the ciphertext by colluding with the service provider. However, the service provider just keeps the latest revocation key in its memory and the old one is erased each time the revocation takes place. As a result, D_i' and μ_k cannot be calculated and thus the revoked user is unable to decrypt the ciphertext even if his attributes satisfy the access policy. Hence, the proposed schemes are collusion-resistant against this attack.

From the above, the proposed schemes are secure against the collusion attacks. □

6.3. Forward and Backward Secrecy. In the following, we will show that the proposed schemes can achieve the property of forward and backward secrecy.

Theorem 5. *The proposed schemes guarantee forward and backward secrecy of the data owners' private data against any revoked user.*

Proof. Suppose the data owner decides to revoke a user u_k or some of his attributes for the user revocation case or attribute revocation case, respectively. The owner then creates the revocation list RL including this malicious user u_k, generates the revocation key rk, and sends them to the service provider. For the user revocation case, since u_k belongs to the revocation list, the service provider cannot calculate μ_k and D_i' for u_k. Thus, the service provider is incapable of canceling the random part $e(g, H(\rho(i)))^{-tr_i\beta}$ appearing in the transformation process. Hence, the user's private key is completely revoked, and he will have no access to the plaintext of subsequent data uploaded afterwards. With regard to the attribute revocation case, the service provider will be unable to compute the $\mu_{\rho(i),k}$ and D_i' if u_k's attribute $\rho(i)$ is revoked, so it will have no ability to cancel $e(g, H(\rho(i)))^{-tr_i\beta}$. Thus, the user will have no chance to access the plaintext of subsequent data as well (unless his remaining attributes satisfy the access policy).

As to the backward secrecy, if the revoked user u_k wants to access the previous data, he will transmit u_k and tk to the service provider. Nevertheless, it cannot calculate $\mu_k/\mu_{\rho(i),k}$ and D_i' for u_k, since the revocation key rk is updated and u_k is included in the revocation list. Therefore, the user will have no access to the previous data, except that his remaining attributes still satisfy the access policy (for the attribute revocation case). Additionally, even though $\mu_k/\mu_{\rho(i),k}$ for the revocation list $RL/RL_{\rho(i)}$ and D_i' for the ciphertext D_i have been calculated before he/his some attributes was revoked, they will have no help to decrypt the subsequent or previous encrypted data. This is because rk is updated and D_i' for the new D_i is needed to be calculated. Therefore, the forward and backward secrecy can be guaranteed in the proposed schemes. □

7. Scalability and Performance Analysis

In this section, we analyze the scalability and performance of the proposed solutions in detail and compare them with some related works. First, we give a comprehensive analysis and comparison in terms of access policy, revocation level, efficiency, and security (analyzed in the last section). Then, we give a detailed performance analysis with respect to storage overhead, communication cost, and computation cost.

7.1. Comprehensive Analysis. To the best of our knowledge, there exists only one access control scheme with revocation [17] designed for MSNs. Here, we will compare our schemes with [17] and the representatively related works [15, 28], which are designed for OSNs.

The results are shown in Table 1. To tackle the revocation and efficiency issues in [4], Sun et al. [28] presented a secure privacy-preserving architecture for OSNs with efficient revocation. However, each member in this design is assigned with only one role, so access policy is quite inflexible. Thus, this method cannot achieve fined-grained access control. Additionally, it may incur multiple encryptions on the same data if many roles are allowed to access the data. Due to the ABE's expressiveness, [15, 17] and our schemes solve the problems above and achieve efficient revocation. [15] is the state-of-the-art architecture for social network privacy, which can realize efficient user revocation without requiring key update. In contrast, [17] needs an update procedure, in which the trusted authority (TA) broadcasts the update key generated by itself to each user in the system. However, the security of the basic ABE used by [17] is much stronger than that in [15], which guarantees the stronger security in [17] than in [15]. With regard to revocation, [15] only gave a scheme with user revocation, although they claimed that the attribute revocation can be realized similarly. [17] can only achieve user revocation as well. The additional downside of these two approaches is the expensive decryption cost which is linearly increasing with the size of the access formula. Exploiting new techniques, our schemes not only can achieve attribute revocation and stronger security, but also can be quite efficient. As to the detailed efficiency analysis, especially for mobile users, please refer to the following section.

In conclusion, the proposed schemes are much more efficient, scalable, and secure than the previous ones. Since the concrete construction is not given in [28] and it is not based on ABE, its security and efficiency are not considered here.

7.2. Performance Analysis. From the above table, it is easy to find that only Jahid et al.'s work [15] and Liang et al.'s work [17] have almost the same scalability with our work. In the following, we mainly conduct the detailed performance analysis among our work and the above two works in terms of the metrics of storage overhead, communication cost, and computation cost. Before going into the concrete analysis, we give some notations in Table 2.

First of all, we compare each component involved in schemes [15, 17] and our schemes, including the size of public key, user private key, ciphertext and Rev (ocation) message, and the computation cost of encrypt by data owner and decrypt by mobile user.

The results are given in Table 3. The public key is used for encryption by the data owner. We can see that the size of public key in our schemes is almost the same as Jahid's scheme but much less than Liang's scheme. This will save a lot storage cost for the data owner. For the private key used by the user for decryption, its size in our scheme is similar to Liang's scheme. However, it is much less than Jahid's scheme. The size of ciphertext is almost the same in all the schemes above, which is linear with the number of attributes used in encryption.

With regard to the revocation, the size of revocation message in our scheme$_1$ is the same as that in [15] but less than that in [17]. About the revocation level, our advanced scheme$_2$ can realize attribute revocation. However, [17] cannot achieve this level, and [15] does not give the concrete construction for this either. What is more important, the revocation message in both our schemes and Jahid's scheme is stored in the service provider and need not to be broadcasted to each user. In contrast, the revocation message in [17] needs to be broadcasted to each user in the system. This puts a large burden either in storage or communication cost on the system.

As to the computation cost, the encrypt cost for the data owner is almost the same in all these schemes, which

Table 1: Comprehensive comparison of related work.

Scheme	Flexible access policy	Revocation level	Revocation security		Key update	Efficiency	Security
			Forward	Backward			
Sun et al. [28]	No	User, immediate	Yes	No	Unnecessary	—	—
Jahid et al. [15]	Yes	User, immediate	Yes	Yes	Unnecessary	Inefficient	Weak
Liang et al. [17]	Yes	Attribute, periodical	Yes	No	Necessary	Inefficient	Strong
Our scheme_1	Yes	User, immediate	Yes	Yes	Unnecessary	Efficient	Strong
Our scheme_2	Yes	Attribute, immediate	Yes	Yes	Unnecessary	Efficient	Strong

Table 2: Notations for efficiency comparison.

$\|G\|$	Bit size of the element in $G/G_0/G_1$
$\|G_T\|$	Bit size of the element in G_T
$\|Z_p\|$	Bit size of the element in Z_p
$\|\mathbb{A}\|$	Bit size of the access policy/structure $\mathbb{A}$ in ciphertext
u	Number of attributes in the system
n_u	Number of one user's attributes in the system
l	The total number of attributes used for encryption
$\|I\|$	The total number of attributes used for decryption
t	The maximum number of users revoked one time
t_x	The maximum number of users revoked for one attribute each time
n	The maximum number of users in the system
T_e	The time used for computing one modular exponentiation operation in G/G_T
T_p	The time used for computing one bilinear pairing operation

increases linearly with the number of attributes used in the data encryption. However, the decrypt cost for the user in our schemes is much less than other schemes, which just includes one modular exponentiation.

These indicates that our solution has much better efficiency compared with these representatively state-of-the-art works. Then, we conduct the detailed performance analysis from each aspect of storage overhead, communication cost, and computation cost in the following part.

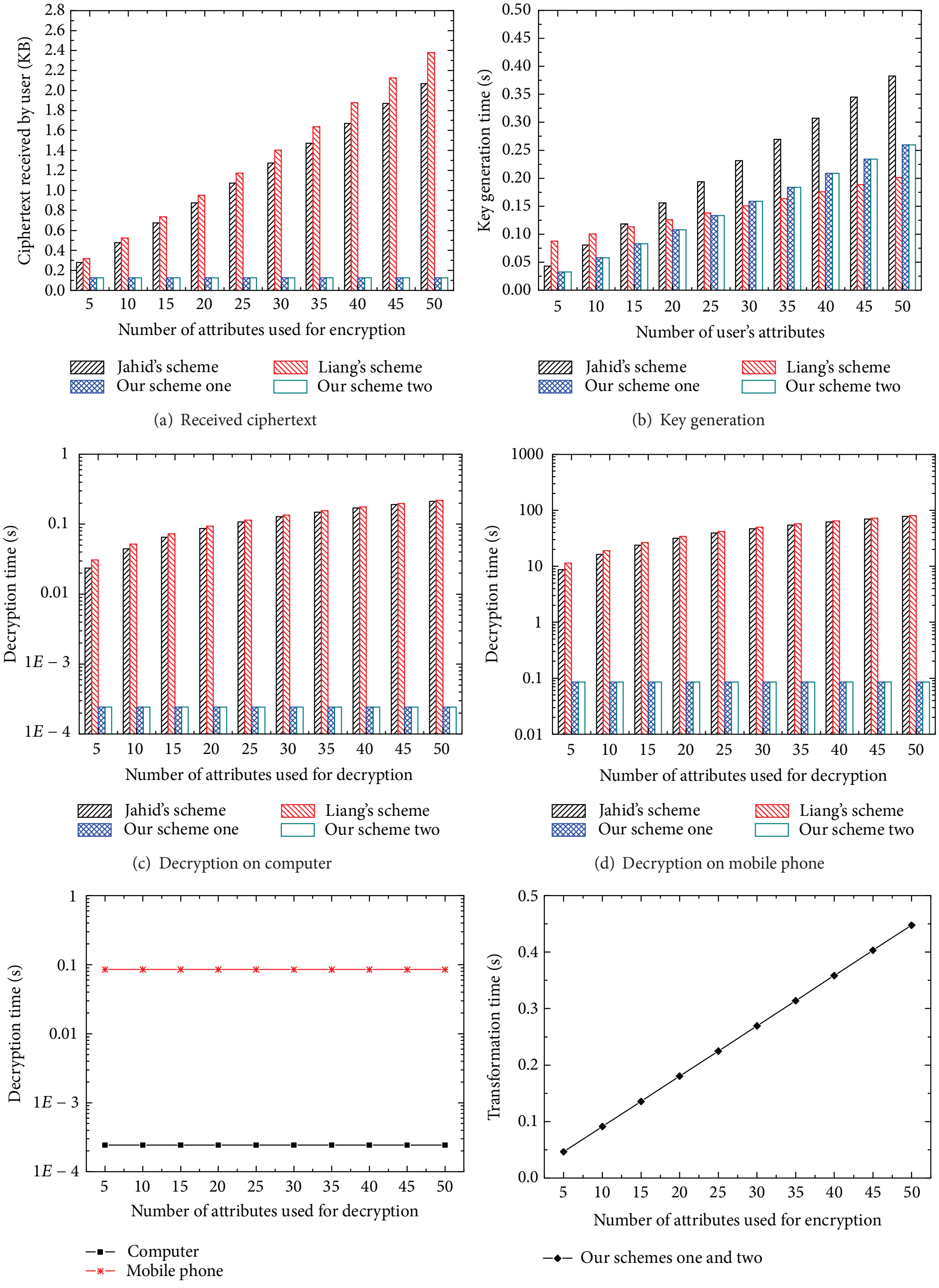
7.2.1. Storage Overhead. The storage overhead is one of the most important concerns in MSNs, especially for the data owners and mobile users in the system. In the comparison, we adapt the works [15, 17] to fit our setting by assuming their using one service provider to store the ciphertext. We compare the overhead on each entity in this system.

The results are shown in Table 4. In these schemes, the storage overhead on each owner mainly comes from the public key used for data encryption. Except for the scheme in [17], our schemes and schemes of [15] have almost the same storage overhead. For the storage overhead on each user, the user private key makes a main contribution to it. In Liang's scheme, however, the overhead consists of both the private key and the revocation message. When a revocation event occurs in [17], the revocation information needs to be broadcasted to each user in the system, and only the nonrevoked user can use this message to decrypt the ciphertext. This incurs a heavy storage cost for the user. In both our schemes and those of [15], the storage overhead increases linearly with the attributes of the user, but our scheme's overhead is much less than that of [15]. Additionally, when the user decrypts one ciphertext, he just needs to store two group elements CT' received from the service provider in our schemes. However, the user in other schemes has to store the complete ciphertext. The detailed comparison of the size of ciphertext received by the user is presented in Figure 3(a). Overall, the storage overhead on the user in our schemes is much less than that in other schemes.

As to the storage overhead on the service provider, the ciphertext contributes the main storage overhead on it, with the exception that our schemes and those of [15] additionally include the revocation message. However, the service provider typically possesses powerful storage and computing resources, so it is not a big issue for the service provider. From the discussion and the table above, we can see that our schemes can save much more storage overhead contrast to other schemes.

7.2.2. Communication Cost. The comparison of communication cost for each user and owner in the system is shown in Table 5.

The communication cost in the system is mainly caused by the private keys, revocation message, and ciphertext. The private key contributes the main communication cost between the user and the authority. However, the revocation message in [17] also leads to a heavy communication cost between the authority and the user. By contrast, the communication cost between the authority and the user in our schemes is less than that in other schemes. Since each data owner in our schemes and [15] acts as one authority, responsible for distributing private keys for her contacts, there is no communication cost between the authority and the owner. The communication cost between the service provider and the user mainly consists of the (transformed) ciphertext. Even if in Jahid's scheme and our schemes the user needs to transmit his transformation key tk to the service provider when he wants to decrypt some ciphertext, he just needs to send it for one time, unless his private key is regenerated. As to the communication cost between the service provider and



(a) Received ciphertext

(b) Key generation

(c) Decryption on computer

(d) Decryption on mobile phone

(e) Decryption for our scheme

(f) Transformation for our scheme

Figure 3: Computation cost with different number of attributes.

Table 3: Comparison of efficiency.

Component	Jahid et al. [15]	Liang et al. [17]	Our scheme$_1$	Our scheme$_2$
Public key	$3\|G\| + \|G_T\| + \|H\|$	$(u+3)\|G\| + \|G_T\| + \|Z_p\| + \|H\|$	$2\|G\| + \|G_T\| + \|H\|$	$2\|G\| + \|G_T\| + \|H\|$
Private key	$(3n_u + 1)\|G\|$	$(n_u + 2)\|G\| + 2(\log n)\|G\|$	$(2n_u + 2)\|G\| + \|Z_p\|$	$(2n_u + 2)\|G\| + \|Z_p\|$
Ciphertext	$(2l+1)\|G\| + \|G_T\| + \|\mathbb{A}\|$	$(2l+3)\|G\| + \|G_T\| + \|\mathbb{A}\|$	$(2l+1)\|G\| + \|G_T\| + \|\mathbb{A}\|$	$(2l+1)\|G\| + \|G_T\| + \|\mathbb{A}\|$
Rev message	$2t\|Z_p\|$	$2t(\log(n/t))\|G\|$	$2t\|Z_p\|$	$2u\|Z_p\|$
Encrypt cost	$(2l+2)T_e$	$(2l+4)T_e$	$(2l+2)T_e$	$(2l+2)T_e$
Decrypt cost	$(2\|I\|+1)T_p + (\|I\| + \log l)T_e$	$(2\|I\|+5)T_p + (\|I\|+1)T_e$	T_e	T_e

Table 4: Comparison of storage overhead.

Scheme	Owner	User	Service provider
Jahid et al. [15]	$3\|G\| + \|G_T\| + \|H\|$	$(3n_u + 1)\|G\|$	$(2l+1)\|G\| + \|G_T\| + \|\mathbb{A}\| + 2t\|Z_p\|$
Liang et al. [17]	$(u+3)\|G\| + \|G_T\| + \|Z_p\| + \|H\|$	$(n_u + 2)\|G\| + 2(\log n)\|G\| + 2t(\log(n/t))\|G\|$	$(2l+3)\|G\| + \|G_T\| + \|\mathbb{A}\|$
Our scheme$_1$	$2\|G\| + \|G_T\| + \|H\|$	$(2n_u + 2)\|G\| + \|Z_p\|$	$(2l+1)\|G\| + \|G_T\| + \|\mathbb{A}\| + 2t\|Z_p\|$
Our scheme$_2$	$2\|G\| + \|G_T\| + \|H\|$	$(2n_u + 2)\|G\| + \|Z_p\|$	$(2l+1)\|G\| + \|G_T\| + \|\mathbb{A}\| + 2u\|Z_p\|$

the owner, it mainly comes from the ciphertext stored to the service provider by owner. In our schemes and [15], the owner also needs to send the renewed revocation key to the service provider when a revocation event occurs. However, the cost for it is much less than that for the ciphertext. On the whole, our schemes have a little more communication cost between the service provider and the owner than [17] and have the same communication cost as [15].

From Table 5 and the discussion, our schemes have a comparable communication cost for each owner and lower communication cost for each user in contrast to schemes in [15, 17].

7.2.3. Computation Cost. We simulate the computation time of key generation, encryption, decryption, and ciphertext transformation in our schemes and the representatively related works [15, 17]. The simulations are performed on two hardware platforms: a 2.40 GHz Intel Core 4 Xeon platform with 4.00 GB RAM running 64 bit Fedora17 and 1.4 GHz SAMSUNG Galaxy SIII I9300 3G with 1 GB of RAM running Android 4.0.4. The code uses the Stanford Pairing-Based Cryptography (PBC) library version 0.5.12 [35] to implement the privacy-preserving schemes. We use a pairing-friendly symmetric elliptic curve type a $y^2 = x^3 + x$ providing the groups where a bilinear map $e : G \times G \rightarrow G_T$ is defined. The elliptic curve group has a 160 bit group order and the embedding degree is 2. This setting has also been adopted in other related works such as [6, 13].

The comparisons on computation cost are shown in Figure 3. The comparison of key generation time with different number of user attributes is presented in Figure 3(b). The key generation time in these schemes increases linearly with the number of user attributes. It is not hard to observe that the time in our schemes grows slower than other schemes in the case that the attribute number is not large enough and that in other case the time consumed in our schemes rises a little faster than Liang's scheme but much slower than Jahid's scheme. When the number of attributes is up to 50, the time is just a little more than 0.25 s, which is reasonable in practice. The encryption time (not shown in the figure) in all these schemes are increased linearly with the number of user attributes, and they take almost the same amount of time.

The decryption time on computer is shown in Figure 3(c), from which we can see that our schemes are much more efficient than other schemes. Additionally, all these schemes' decryption time on mobile phone is simulated and shown in Figure 3(d). We also compare our schemes' decryption time on computer with that on mobile phone in Figure 3(e). It is easy to find that our schemes are much more suitable for applications on mobile phones compared to the other schemes. Even though the decryption time of our schemes on mobile phone is much more than that on computer, it is acceptable and feasible for mobile devices. This demonstrates that our schemes can be used not only for the computer users, but also for the mobile users. Moreover, our schemes have an additional advantage that the size of ciphertext received by the user is much less than other schemes'. Hence, this can save a lot storage cost for mobile devices.

As to the computation cost consumed by service provider, it mainly comes from the exponentiation and the pairing operation, which is increased linearly with the number of attributes appeared in the encryption. However, the total time for 50 attributes is less than 0.45 s, shown in Figure 3(f), which is quite easy to calculate, especially for the service provider who possesses powerful computing resources. Additionally, μ_k and $\mu_i s$ also need to be computed. However, this cost can be ignored contrast to the exponentiation and pairing operations. Actually, an optimization can be performed by allowing the service provider to precompute a portion $\mu_i' s$ of the $\mu_i s$: $\mu_i' = \prod_{u_i, u_j \in RL, i \neq j}(u_j/(u_j - u_i))$, and $l_i' = \mu_i' P(u_i)$. With the optimization, the service provider just needs to compute one multiplication when the user $u_k \notin RL$ wants to access the data: $l_i = l_i'(u_k/(u_l - u_i)) = \mu_i' P(u_i)(u_k/(u_l - u_i)) = \mu_i P(u_i)$, for all $u_i \in RL$.

Table 5: Comparison of communication cost.

Scheme	Authority and user	Authority and owner	Service provider and user	Service provider and owner
Jahid et al. [15]	$(3n_u + 1)\lvert G\rvert$	\	$(2l + 1)\lvert G\rvert + \lvert G_T\rvert + 2\lvert Z_p\rvert + \lvert\mathbb{A}\rvert$	$(2l + 1)\lvert G\rvert + \lvert G_T\rvert + 2t\lvert Z_p\rvert + \lvert\mathbb{A}\rvert$
Liang et al. [17]	$(n_u + 2)\lvert G\rvert + 2(\log n)\lvert G\rvert + 2t(\log(n/t))\lvert G\rvert$	$(u + 3)\lvert G\rvert + \lvert G_T\rvert + \lvert Z_p\rvert + \lvert H\rvert$	$(2l + 3)\lvert G\rvert + \lvert G_T\rvert + \lvert\mathbb{A}\rvert$	$(2l + 3)\lvert G\rvert + \lvert G_T\rvert + \lvert\mathbb{A}\rvert$
Our scheme$_1$	$(2n_u + 2)\lvert G\rvert + \lvert Z_p\rvert$	\	$2\lvert G_T\rvert + 2\lvert Z_p\rvert$	$(2l + 1)\lvert G\rvert + \lvert G_T\rvert + 2t\lvert Z_p\rvert + \lvert\mathbb{A}\rvert$
Our scheme$_2$	$(2n_u + 2)\lvert G\rvert + \lvert Z_p\rvert$	\	$2\lvert G_T\rvert + 2\lvert Z_p\rvert$	$(2l + 1)\lvert G\rvert + \lvert G_T\rvert + 2u\lvert Z_p\rvert + \lvert\mathbb{A}\rvert$

From the above, it is not hard to find that the proposed schemes incur less computation cost overall, especially for the decryption procedure. Thus, they are much more suitable for the MSN users who always like to access the data from their mobile devices.

8. Conclusion

In this paper, we present a secure, efficient, and fine-grained access control system for mobile social networks (MSNs). Compared with most of the previous works, the proposed solution can effectively address the system efficiency, data confidentiality, access control, and membership revocation issues simultaneously. In our system, not only can the data owner in MSNs flexibly define the access policy and enforce it over his private data by herself, but also can efficiently implement immediate user/attribute revocation. Moreover, the mobile users can quite efficiently access the encrypted data with the help of service provider, and the revocation is both flexible and fine-grained. The detailed security analysis shows that the proposed framework can achieve the main security requirements, and the theoretical analysis and the implementations demonstrate our system's efficiency, scalability, and functionality.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to all the anonymous reviewers for valuable suggestions and comments. The work described in this paper was supported by the Doctoral Fund of Ministry of Education of China (20120073110094), National Natural Science Foundation of China (61202371 and 61202367), Natural Science Foundation of Shanghai (12ZR1443700), and the Innovation Program of Shanghai Municipal Education Commission (14YZ020).

References

[1] M. O'Connor, "Facebook revealed private email addresses last night," 2010, http://gawker.com/5505967/facebook-revealed-privateemail-addresses-last-night.

[2] P. Wong, *Conversations about the Internet # 5: Anonymous Facebook Employee*, The Rumpus, 2010.

[3] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient controlled encryption: ensuring privacy of electronic medical records," in *Proceedings of the ACM Workshop on Cloud Computing Security (CCSW '09)*, pp. 103–114, ACM, 2009.

[4] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," in *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '09)*, pp. 135–146, ACM, New York, NY, USA, 2009.

[5] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT 2005*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer, Berlin, Germany, 2005.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in *Proceedings of the Symposium on Security and Privacy (SP '07)*, pp. 321–334, IEEE Computer Society, Washington, DC, USA, 2007.

[7] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 456–465, ACM, New York, NY, USA, 2007.

[8] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology—EUROCRYPT 2010*, pp. 62–91, Springer, Berlin, Germany, 2010.

[9] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 121–130, November 2009.

[10] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology—EUROCRYPT 2011*, vol. 6632 of *Lecture Notes in Computer Science*, pp. 568–588, Springer, Berlin, Germany, 2011.

[11] B. Waters, “Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization,” in *Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography (PKC ’11)*, pp. 53–70, Taormina, Italy, March 2011.

[12] X. Liang, R. Lu, L. Chen, X. Lin, and X. Shen, “PEC: a privacy-preserving emergency call scheme for mobile healthcare social networks,” *Journal of Communications and Networks*, vol. 13, no. 2, pp. 102–112, 2011.

[13] J. Hur and D. K. Noh, “Attribute-based access control with efficient revocation in data outsourcing systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[14] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in *Proceedings of the INFOCOM ’10*, pp. 534–542, IEEE, San Diego, Calif, USA, 2010.

[15] S. Jahid, P. Mittal, and N. Borisov, “EASiER: encryption-based access control in social networks with efficient revocation,” in *Proceedings of the 6th International Symposium on Information, Computer and Communications Security (ASIACCS ’11)*, pp. 411–415, March 2011.

[16] G. Wang, Q. Liu, and J. Wu, “Hierarchical attribute-based encryption for fine-grained access control in cloud storage services,” in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS ’10)*, pp. 735–737, October 2010.

[17] X. Liang, X. Li, R. Lu, X. Lin, and X. Shen, “An efficient and secure user revocation scheme in mobile social networks,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’11)*, pp. 1–5, Houston, Tex, USA, December 2011.

[18] S. Yu, K. Ren, and W. Lou, “FDAC: toward fine-grained distributed data access control in wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 4, pp. 673–686, 2011.

[19] M. Green, S. Hohenberger, and B. Waters, “Outsourcing the decryption of ABE ciphertexts,” in *Proceedings of the 20th USENIX Conference on Security (SEC ’11)*, p. 34.

[20] A. Pietilainen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, “MobiClique: middleware for mobile social networking,” in *Proceedings of the 2nd ACM Workshop on Online Social Networks (WOSN ’09)*, pp. 49–54, 2009.

[21] M. Li, N. Cao, S. Yu, and W. Lou, “FindU: privacy-preserving personal profile matching in mobile social networks,” in *Proceedings of the IEEE INFOCOM*, pp. 2435–2443, Shanghai, China, April 2011.

[22] R. Zhang, Y. Zhang, J. Sun, and G. Yan, “Fine-grained private matching for proximity-based mobile social networking,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM ’12)*, pp. 1969–1977, IEEE, Orlando, Fla, USA, March 2012.

[23] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *Advances in Cryptology—EUROCRYPT ’99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings*, Lecture Notes in Computer Science, pp. 223–238, Springer, Berlin, Germany, 1999.

[24] W. Dong, V. Dave, L. Qiu, and Y. Zhang, “Secure friend discovery in mobile social networks,” in *Proceedings of the IEEE INFOCOM*, pp. 1647–1655, April 2011.

[25] M. Lucas and N. Borisov, “FlyByNight: mitigating the privacy risks of social networking,” in *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society (WPES ’08)*, pp. 1–8, 2008.

[26] W. Luo, Q. Xie, and U. Hengartner, “FaceCloak: an architecture for user privacy on social networking sites,” in *Proceedings of the IEEE International Conference on Privacy, Security, Risk, and Trust (PASSAT ’09)*, pp. 26–33, Vancouver, Canada, August 2009.

[27] S. Guha, K. Tang, and P. Francis, “NOYB: privacy in online social networks,” in *Proceedings of the ACM 1st Workshop on Online Social Networks (WOSP ’08)*, pp. 210–230, August 2008.

[28] J. Sun, X. Zhu, and Y. Fang, “A privacy-preserving scheme for online social networks with efficient revocation,” in *Proceedings of the IEEE INFOCOM 2010*, pp. 1–9, San Diego, Calif, USA, March 2010.

[29] A. Fiat and M. Naor, “Broadcast encryption,” in *Advances in Cryptology—CRYPTO ’93: Proceedings of the 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22–26, 1993*, vol. 773 of *Lecture Notes in Computer Science*, pp. 480–491, Springer, Berlin, Germany, 1994.

[30] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in *Advances in Cryptology—EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 506–522, Springer, Berlin, Germany, 2004.

[31] S. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, “Over-encryption: management of access control evolution on outsourced data,” in *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB ’07)*, pp. 123–134, 2007.

[32] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, “Mediated ciphertext-policy attribute-based encryption and its application,” in *Proceedings of the 10th International Workshop on Information Security Applications (WISA ’09)*, pp. 309–323, August 2009.

[33] A. Beimel, *Secure schemes for secret sharing and key distribution [Ph.D. thesis]*, Technion: Israel Institute of Technology, Haifa, Israel, 1996.

[34] M. Naor and B. Pinkas, “Efficient trace and revoke schemes,” *International Journal of Information Security*, vol. 9, no. 6, pp. 411–424, 2010.

[35] B. Lynn, “The standard pairing based crypto library,” http://crypto.stanford.edu/pbc.