A Privacy-Preserving Scheme for Online Social Networks with Efficient Revocation

Jinyuan Sun*, Xiaoyan Zhu†, and Yuguang Fang*†
*Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
†National Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
Email: {stellas@, xiaoyanzhu@, fang@ece.}ufl.edu

Abstract—**Online social networks (OSNs) are attractive applications which enable a group of users to share data and stay connected. Facebook, Myspace, and Twitter are among the most popular applications of OSNs where personal information is shared among group contacts. Due to the private nature of the shared information, data privacy is an indispensable security requirement in OSN applications. In this paper, we propose a privacy-preserving scheme for data sharing in OSNs, with efficient revocation for deterring a contact's access right to the private data once the contact is removed from the social group. In addition, the proposed scheme offers advanced features such as efficient search over encrypted data files and dynamic changes to group membership. With slight modification, we extend the application of the proposed scheme to anonymous online social networks of different security and functional requirements. The proposed scheme is demonstrated to be secure, effective, and efficient.**

I. Introduction

Data sharing is finding ever-growingly broad applications and becoming an essential part of our daily life. It enables real-time communications such as instant messaging (e.g., MSN) and internet phone (e.g., Skype), and offline communications such as private message (e.g., YouTube) and blogging (e.g., Twitter), among users regardless of their physical locations. Online social networks (OSNs) are among the most popular data sharing applications and have been soaring in recent years. Some well-known general OSNs include Facebook, Myspace, and Twitter that are familiar to many of us. There are also numerous other OSNs with special focuses, for example, on photo sharing (e.g., Flickr), travel, dating, business, college, etc. OSNs facilitate families and friends to stay connected and maintain their social relations in a more convenient and affordable way than traditional phone conversations, mails and emails, and hence have gained enormous popularity among social groups.

Despite the various appealing features offered by OSNs, users' data privacy is always at risk when the network is exploited for adversarial activities, e.g., accessing private data without permissions, illegally selling private data, profiling the data owner, etc. The risk is dramatically increased especially when users are encouraged to include their real names and profiles in the OSN applications (e.g., in Facebook), rendering users vulnerable to data privacy breaches. As a result, secure and efficient privacy preservation schemes suitable for data sharing in OSNs are highly in demand. Recent research work has demonstrated interests along this line.

Related work. The OSN with user-defined privacy proposed in [1] is most relevant to our work. Baden *et al.* [1] propose to use the attribute-based encryption (ABE) to apply access control over group members. The ability to logically express the attributes associated with each member provides a convenient way for the group manager to assign keys to the members. These keys are used to encrypt the group manager's personal data, so that each member's access to the data is properly restricted and the group manager's data privacy is guaranteed. The downside of this work is the lack of an efficient membership revocation mechanism, and the computational cost associated with the ABE operations. NOYB [2] offers privacy and preserves the functionality of online services using secret dictionaries. Data privacy of the profiles can be assured in NOYB while user privacy (i.e., user relations and interactions) is not protected. Yardi *et al.* [3] proposed to use graphs in Facebook to build a photo-based authentication framework. In [4], Ferrer *et al.* leverage homomorphic encryption to enable resource access through indirect social relationships without a trusted third party, preserving the users' private social relationships from the resource owner. Recently, OSNs preserving privacy based on peer-to-peer (P2P) overlays [5], [6] are also proposed. Other research works [7]–[10] have focused on different security issues such as Sybil attack and anonymity in general social networks. Privacy and anonymity are extensively studied in many other data sharing networks such as Crowds [11], and the notable P2P networks such as Freenet [12] and Tarzan [13].

Our contributions. In this paper, we propose a secure privacy-preserving scheme for data sharing in online social networks. The primary goals of the proposed scheme are to guarantee data privacy and access control with regard to the private data stored on potentially untrusted storage sites. In addition to assuring data privacy and access control, our scheme is designed to enable secure and efficient search over shared data and to support dynamic revocation suitable for the frequently changing social group membership. We provide analysis and enhancements to show that the proposed scheme is secure, effective, and efficient. With slight modification, we extend the application of the proposed scheme to anonymous online social networks, which bear more stringent security requirements due to the additional privacy demanded for *group members* who access the shared data. Throughout the discussion of this paper, we use the basic online social networks as the context to present the proposed scheme. The application of the scheme to anonymous online social networks is the subject of Section VI.

The remainder of this paper is organized as follows. Section II introduces some preliminary knowledge of the cryptosystem

This work was partially supported by the U.S. National Science Foundation under grants CNS-0916391, CNS-0716450, CNS-0721744 and CNS-0626881, and China 111 Project under grant B08038. The work of X. Zhu was also partially supported by the National Natural Science Foundation of China under Grant 60772136 and the 863 Project of China under Grant 2007AA01Z435.

X. Zhu is currently a visiting research scholar with Department of Electrical and Computer Engineering, University of Florida. Y. Fang is also a Changjiang Scholar Chair Professor with National Key Laboratory of Integrated Services Networks, Xidian University, China.

and cryptographic schemes. Section III describes the network and threat models, and identifies the security objectives of our scheme. The proposed privacy preservation scheme is presented in detail in Section IV, followed by the analysis and possible enhancements in Section V. The application of our scheme to the anonymous online social networks is discussed in Section VI, and Section VII concludes the paper with future work.

II. Preliminaries

We introduce some preliminaries used in the subsequent development of our scheme.

A. IBC from Bilinear Pairings

Identity-based cryptography (IBC) allows the public key of an entity to be derived from its public identity information such as name, email address, etc. Boneh and Franklin [14] introduced the first functional and efficient ID-based encryption scheme based on bilinear pairings on elliptic curves. Specifically, let G_1 and G_2 be an additive group and a multiplicative group, respectively, of the same prime order q. Discrete logarithm problem (DLP) is assumed to be hard in both G_1 and G_2. Let P denote a random generator of G_1 and $e : G_1 \times G_1 \rightarrow G_2$ denote a bilinear map constructed by modified Weil or Tate pairing with the following properties:

1. Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$, $\forall P, Q \in G_1$ and $\forall a, b \in Z_q^*$.
2. Non-degenerate: $\exists P, Q \in G_1$ such that $e(P, Q) \neq 1$.
3. Computable: there exists an efficient algorithm to compute $e(P, Q), \forall P, Q \in G_1$.

IBC is used to build the secure communication backbone and to enable efficient search in our scheme.

B. Broadcast Encryption

Broadcast encryption [15] allows a central transmitter to send encrypted data to a set of users such that only a privileged subset of users can decrypt the data. Broadcast encryption is designed for and widely applied in the secure distribution of copyrighted media (e.g., TV programs, DVD) over cable TV or the Internet. Other applications of broadcast encryption include encrypted file systems (e.g., Windows EFS) for restricted file sharing, mailing list applications for sending confidential emails, and so forth.

A broadcast encryption scheme is defined by the algorithm tuple (SETUP, BROADCAST, DECRYPT) where

- SETUP: takes (the identifier or public key of) a user $u \in \mathcal{U}$ ($\mathcal{U}$ being the set of all users) and constructs the user's secret $\mathbf{\Gamma_u}$;
- BROADCAST: takes the set of revoked users $\mathcal{R}$ and the decryption key K, and outputs a broadcast ciphertext C;
- DECRYPT: is run by a user $u \in \mathcal{U}$ to compute the decryption key K encrypted in the ciphertext C, if $u \in \overline{\mathcal{R}}$ where $\overline{R}$ denotes the set of non-revoked users. If $u \in \mathcal{R}$, DECRYPT fails.

Broadcast encryption is leveraged to provide efficient membership revocation for the proposed scheme.

C. Searchable Public-Key Encryption

Public key encryption with keyword search (PEKS), or simply searchable public-key encryption, allows an email server to tell if a given keyword is present in emails destined to the receiver without learning anything else about the encrypted emails. Data (encrypted with the receiver's public key) are stored in the remote server by the sender, and will be decrypted and used by the receiver. The receiver generates trapdoors for keywords of his/her choice and sends the trapdoors to the server. The server searches over the encrypted data from the sender and only returns data containing particular keywords to the receiver. The encryption and decryption in PEKS are performed by different parties, where public-key encryption should be employed.

Boneh $et\ al.$ [16] first established the security definitions of PEKS and provided a construction based on the identity-based encryption (IBE) [17]. In this paper, we will use the PEKS scheme in [16] and the role-based encryption for members with a proper role to efficiently retrieve the encrypted data files.

III. System and Security Models

In this section, we present the network model and threat model, and define the security objectives in the context of online social networks (OSNs).

A. Entities and Trust

Credential authority is in charge of cryptographically initializing an OSN domain and issuing a public/private key pair to each user in the domain. An OSN domain consists of a credential authority and all the registered users. It is necessary for users to possess legitimate credentials (i.e., key pairs) in order to perform security operations in the domain. A service provider acts as the credential authority in the OSN, e.g., the administrator in the Facebook social network. The credential authority is generally trusted by the OSN users and is provided with the users' identity information (e.g., email address) upon registration.

Storage site is a third-party provider that offers free or priced mass storage space to accommodate user data possibly from multiple OSN applications or domains. The storage site is not trusted by the OSN users because it is not directly run by the OSN. The reason that we assume the untrusted third-party storage in favor of trusted proprietary storage (owned by the OSNs), is to take a more hostile and challenging environment into account when carrying out our security design.

Data owner or group manager, is an OSN user who shares personal or private data within his/her groups of contact, controls access of the group members to the private data, and adds/removes users from his/her groups. Hereafter, we use group to represent all contacts of a data owner who classifies these contacts (or the group) into different subgroups, based on the contacts' social relationships with the data owner.

Member is an OSN user and a contact of one or more data owners' subgroup. The member may take on a different role in each data owner's subgroup (e.g., one's classmate and another's family). The member is meanwhile the data owner of his/her own group. The trust relationship between the data owner and a member is based on the social relationship of the two. For example, one trusts the family but may not trust a friend made in the online chatting room.

The topology of an online social network and the associated interactions between entities are illustrated in Fig. 1, where dashed, dotted, and dash-dotted lines denote the communications related to the Facebook domain, the Twitter domain, and both, respectively. The texts near each line describe the interactions between the connected sets of entities. For instance, the credential authority in the Facebook domain issues credentials to Facebook users in the real social network and signs business contract with the storage site. The data owner stores his/her private data files on the storage site, from which the data files accessible to legitimate members are retrieved by them. The solid lines in the real social network denote people's social relationships. The person shown in purple, Alice, is the data owner of our interest (although everyone in the figure is a data owner of his/her private data). Alice, having four contacts in the figure, distributes her domain credentials and necessary secret keys to each contact based on the role of that contact in her group (not shown in Fig. 3). Each person in the real social network may be a Facebook user, a Twitter user, or both.

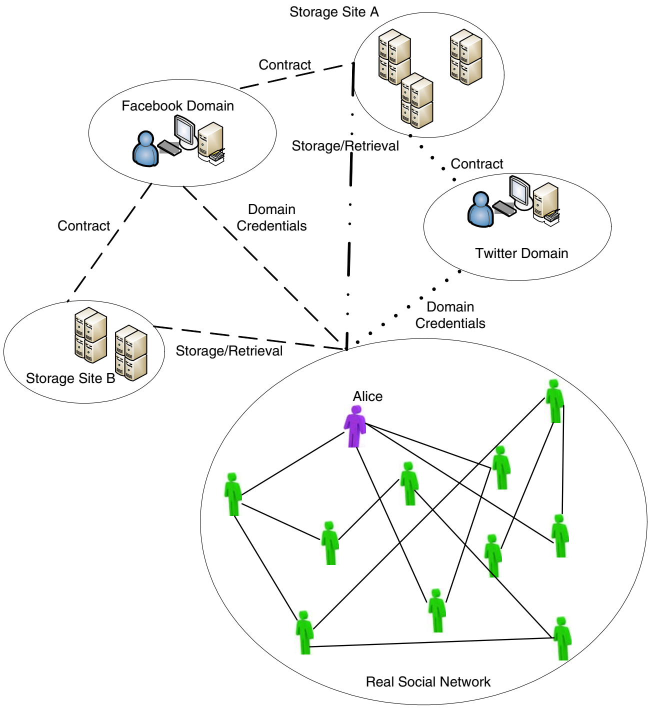

Fig. 1. Topology of An Online Social Network.

B. Security Objectives

The main security objectives of the proposed scheme for OSNs are data privacy and access control, which will be defined and specified together with other security objectives in what follows.

Data privacy: This requirement is two-fold. First, unauthorized entities (i.e., who are not granted access to the private data) must not learn the content of the private data which reveals identifying information of the data owner. This aspect of data privacy implies data confidentiality and (the owner's) anonymity. Second, unauthorized entities must not be able to link multiple private data files to profile the owner, indicating that the stored or transmitted private data should appear random and leak no useful information. This aspect is essentially the unlinkability requirement.

Access control: It requires that unauthorized entities are unable to access the private data at all times, and that revoked entities (i.e., who once had access privilege) will be unable to access the private data after the revocation.

Authentication: It requires that the communicating entities are assured of each other's legitimacy (i.e., the key pair used for authentication is indeed assigned by the credential authority) and authenticity (i.e., an entity is in possession of his/her claimed identity).

Collusion resistance: This requirement states that data privacy is preserved in the presence of collusion attacks where two or more entities collude to obtain more information on the victim than what is available to each colluding individual. We also call a scheme collusion resistant if the colluding entities can later be traced and held responsible for the privacy breach that has occurred. The second interpretation of collusion resistance captures the notion of a preventative approach in which little incentive is provided to the colluders.

C. Threat Model

The threat model defines the attackers and their possible attacks to the proposed scheme for OSNs. The storage site is honest but curious, in that it will not maliciously delete or modify user data, but will attempt to compromise data privacy by learning the content of the private data. Members are curious but non-malicious in that they will attempt to access the private data files to which they are not authorized, while having no incentives to impede the functioning of the OSNs (e.g., by launching Denial-of-Service attacks to the storage site to prevent others' access). Revoked members are allowed to collude with legitimate members to continue accessing the private data. Legitimate members can collude with the storage site trying to access the data files beyond his/her authorization. Legitimate members of different subgroups can also collude together in order to obtain more private data, which is the most powerful form of collusion attack, the insider collusion attack. In addition, active outsiders can inject bogus data or modify legitimate data to disrupt network operations. Passive eavesdroppers will attempt to intercept data on the fly and access the private content.

IV. Privacy Preservation Scheme with Efficient Revocation for Online Social Networks

We first focus on the core techniques of the proposed scheme, i.e., the construction of privacy-preserving data files that achieves privacy and revocation.

A. Overview of The Proposed Scheme

The construction of privacy-preserving data is performed by the data owner whenever personal data need be shared. The basic idea is that the data owner has proper control over access to his/her personal data, especially those revealing identity information and personal life (e.g., photos, videos, copyrighted materials). Typically, the data owner would act as a group manager who classifies contacts according to their roles (e.g., family, coworkers, high school classmates, sports club members) and grants them the corresponding memberships. Each role defines a subgroup, the members of which are restricted to certain data categories. A data category is created by the data owner describing the set of data files that can be accessed as a whole by one or more subgroups. The granularity of data categories is adjustable depending on the fineness of desired access control. For example, when the categories are coarsely defined as music, movies, photos, my stories, etc, a subgroup of members who are permitted to a category can access all the data in that category. This is usually undesirable since the data owner may want to release certain data only to related people (e.g., family photos or videos only accessible to family members). The data owners will have the freedom to create their own categories based on the number and type of their subgroups, which is a design issue and will not be elaborated further.

Since personal data are stored on the potentially untrusted storage site, no information (besides that for efficient search) should be leaked to the site in prevention of data privacy breaches. Furthermore, the group membership should be highly dynamic, in that the data owner needs to constantly add and remove contacts. For instance, contacts are added as old acquaintances are reconnected and new friends are made through OSNs. The feature of making new friends online necessitates membership revocation, as such "e-friends" are not real-life friends and are thus likely to misbehave (e.g., offending the data owner by rude language or vandalism, illegally sharing personal data with unauthorized users). As a result, the proposed scheme should provide the data owner with: 1) a convenient way to cope with membership changes without rebuilding the group or rekeying group members, and 2) an efficient revocation mechanism against misbehaving members, taking into account the dynamically changing group membership and producing minimal impact on data privacy preservation.

B. Constructing Secure and Privacy-Preserving Data

Based on the aforementioned main goals, the privacy-preserving data to be stored are constructed by the data owner as follows:

1. Generating subgroup secret keys α's: For each role-based subgroup r, the data owner randomly selects α_r, $\alpha_r \in_R \{0,1\}^{\varrho}$ where ϱ is a security parameter. The secret key α_r is role-specific and will be used in the encryptions discussed below.
2. Generating $SKE_{\alpha_r}(Data\_Category)$: The data owner encrypts the content of a data category (as defined above), denoted by $Data\_Category$, using any semantically secure symmetric key encryption (SKE) scheme and the secret key α_r. An example of $Data\_Category$ could be all the songs from *The "Thriller" Album.*
3. Generating $BE_{U \setminus R}(\alpha_r)$: The data owner generates a broadcast encryption (BE) on the secret key α_r using the algorithm BROADCAST (cf. Section II B) for the set of non-revoked members $U \setminus R$, taking as input the set of revoked members R and all members U in a subgroup (or role).
4. Distributing role-based key pair $\breve{ID}_{role}/\breve{\Omega}_{role}$: The data owner assigns $\breve{ID}_{role}/\ \breve{\Omega}_{role}$ to each member in the role-based subgroup. This key pair will be used for the role-based searchable public key encryption $PEKS$.
5. Generating $PEKS_{\sigma}(\breve{ID}_{role}, kw)$: The data owner generates the searchable public-key encryption $PEKS$ as $PEKS_{\sigma}(\breve{ID}_{role}, kw) = (\sigma P, H_1(e(H_2(kw), \breve{ID}_{role})^{\sigma}))$ where $\sigma \in_R Z_q^*$ is randomly selected by the data owner, $P \in G_1$ (a generator of G_1, cf. Section II A) is the data owner's public domain parameter, H_1 and H_2 are two publicly-known hash functions defined as $H_1 : G_2 \rightarrow Z_q^*$ and $H_2 : KW \rightarrow G_1$ in which KW represents the keyword space. In addition, kw denotes the keyword used for efficient search and proper return of the above BE-encrypted data. The keyword kw is predefined by the data owner and later used by group members in their queries submitted to the storage site.
6. Storing $PEKS_{\sigma}(\breve{ID}_{role}, kw) \parallel BE_{U \setminus R}(\alpha_r) \parallel SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$: This information is stored by the data owner for later retrieval by members, where the last term in the concatenation denotes the encryption of all the keywords $\omega_1, \cdots, \omega_k$ used for searching the stored data and their corresponding trapdoors $TD_1, \cdots, TD_k$. Trapdoors enable the storage site to test for the occurrence of a keyword in an encrypted file without learning the actual keyword. After successfully decrypting this term, the keywords and trapdoors can be used by members for searching on the encrypted data categories.
7. Storing $SI \parallel SKE_{\alpha_r}(Data\_Category)$'s: Finally, the data owner constructs the secure index SI (cf. Section IV D) from plaintext data categories, and stores the secure index and ciphertext data categories $SKE_{\alpha_r}(Data\_Category)$'s on the storage site.

C. Broadcast Encryption for Efficient Revocation

One of the main contributions that distinguish our scheme from others, is the use of broadcast encryption coupling with role-based searchable encryption that enables the data owner to exercise desired access control, based on his/her Facebook or Twitter subgroups. In our design, each member is assigned with only one role, which is the major difference between our role-based approach and the attribute-based approach [1] for online social networks. In [1], each user can be a member of several social groups of the same owner, e.g., being both a "*neighbor*" and a "*football fan*". The private data are encrypted using logic expressions of attributes such as *neighbor AND football fan*, *neighbor OR coworker*, etc. Each possible combination of all attributes defines a group of members. The advantage of this approach is that each set of data is encrypted once using proper combinations of attributes for a group of members under these attributes. However, complexity is left to defining the attributes for each member and distributing the corresponding keys. The drawbacks lie in the computational cost in attribute-based encryption (a public key based scheme that is 100-1000 times slower than RSA [1]), and the cumbersome membership revocation mechanism which involves rekeying all the non-revoked members in the same group. These drawbacks can be overcome in our role-based approach. Specifically, our approach involves mostly symmetric encryption operations. The costs associated with the PEKS operations and the construction of secure index are due to the additional searchability feature of our scheme.

Efficient revocation can be carried out by our scheme through the adoption of broadcast encryption. Each member u of the data owner's subgroup initially receives a set of secrets (or simply, secret) Γ_u, $\Gamma_u \in \mathbf{\Gamma}$, from the data owner, where $\mathbf{\Gamma}$ is a key space. Upon detecting misbehavior of a member v, the data owner includes v's public key (e.g., v's identity) in the set of revoked member R. The data owner then chooses a new subgroup secret α_r' in the same way as α_r. A new secret is generated for a subgroup only if some member in that subgroup (or role) is revoked. The algorithm BROADCAST will then be executed by inputting the updated R and the new secret α_r'. From this point on, the data owner's private data will be encrypted and stored using α_r' to prevent revoked members from further access. When attempting to decrypt the ciphertext generated by BROADCAST, the revoked member v will input the secret Γ_v to the algorithm DECRYPT. In this case, v will fail to obtain α_r' since v's public key is in R. On the other hand, a non-revoked member u will be able to obtain α_r' (using his/her member-specific secret Γ_u) and continue retrieving the encrypted data. This revocation mechanism is very suitable for OSNs where members join and leave (e.g., due to revocation or withdrawal) frequently. As a result, mechanisms such as [1] which involve rekeying group members are highly inefficiently.

With broadcast encryption, the long-term secret Γ_* ($*$ denotes any member) initially distributed to each member need not be changed when membership changes (only data intended for the revoked member's role r need be re-encrypted with the new secret α'_r). Moreover, rekeying the role-based key pair $\breve{ID}_{role}/\breve{\Omega}_{role}$ is unnecessary either, in that no useful information can be obtained from the role-based PEKS even if the key pair is available to revoked members (cf. Sectioin IV D). Without the knowledge of the new secret α'_r encrypted by $BE_{U\setminus R}(\alpha_{r'})$, the private data of the owner cannot be retrieved or decrypted.

D. Role-based PEKS and Secure Index for Efficient Search

Compared with the attribute-based approach [1] where encryption is performed only once for a data set, our approach may incur multiple encryptions on a same data category. One encryption is needed for each role that is allowed to access the category, using the role-specific secret α_r. However, these multiple encryptions are quite affordable because of the following observations: 1) in reality, there are only a few roles (or subgroups) in a data owner's group, 2) in general, subgroups are formed based on specific data the owner would like to share within each subgroup. If the data are generally accessible, the owner could publicly post them (e.g., on YouTube) or encrypt them under the role *"general"* for which every member has the associated private key, and 3) the encryptions in our scheme can be efficiently carried out since only symmetric encryption operations are involved in producing ciphertext for private data. Compared to the very expensive ABE operations, there is still a performance gain in terms of computational overhead in our scheme.

1) Role-based PEKS: The complexity of our scheme stems from the searchability feature (i.e., PEKS operations and the construction of secure index), which allows members to efficiently search their interested data possibly over multiple owners' private data. The role-based PEKS operations enable the data owner to outsource the private data to the storage site before knowing the potential recipients. This is achieved by the fixed roles in the group even though the membership is dynamic. When a new member m with role r is added to a subgroup, the data owner simply assigns the role-based key pair $\breve{ID}_r/\breve{\Omega}_r$, the set of secrets Γ_m, and the secret α_r currently being used. The role-based key pair is first used to retrieve $BE_{U\setminus R}(\alpha_r) \parallel SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$ associated with $PEKS_\sigma(\breve{ID}_{role}, kw)$, from Step 6 of the privacy preservation scheme. For example, the data owner Alice creates a PEKS-encrypted message for the role "*Family*" in his group by setting $kw = Alice's\_Family$, and stores $PEKS_\sigma(\breve{ID}_{Family}, Alice's\_Family) \parallel BE_{U\setminus R}(\alpha_{Family}) \parallel SKE_{\alpha_{Family}}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$. Knowing the syntax $kw = Alice's\_Family$ a priori, a family member computes a trapdoor $\breve{TD}_{Family}(Alice's\_Family) = \breve{\Omega}_{Family} \cdot H_2(Alice's\_Family)$, and submits the trapdoor to the storage site for the retrieval of $BE_{U\setminus R}(\alpha_{Family}) \parallel SKE_{\alpha_{Family}}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$. Without the proper role assigned by Alice, a non-family member cannot successfully compute the trapdoor where the role-based private key $\breve{\Omega}_{Family}$ is needed.

2) Construction of Secure Index: Next, if member m in role r is not revoked, m obtains the secret key α_r from $BE_{U\setminus R}(\alpha_r)$ and uses α_r to decrypt $SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$. The set of keywords and trapdoors $(\omega_1, TD_1), \cdots, (\omega_k, TD_k)$ is used to search for encrypted private data $SKE_{\alpha_r}(Data\_Category)$'s. Revoked members could have recorded the set of keywords and trapdoors before revocation since keywords are rarely changed. Then the revoked member will be able to query the storage site with trapdoors of interest. However, the search returns private data encrypted with the new secret α'_r which prevents the revoked member from learning the data content.

The search for $SKE_{\alpha_r}(Data\_Category)$'s relies on the secure index, the construction of which is shown in Fig. 2 for our OSN scenario. A secure index [18] is a data structure used by the server to return query results containing a keyword, the trapdoor of which is provided by the querier. With secure index, the server can determine if an encrypted file contains certain keyword without decrypting the file or learning the keyword. In the absence of trapdoors which are computed by the data owner's secrets, the secure index reveals no information about the content of both the index and the encrypted data. The secure index is a common technique used in secure search schemes [18], [19]. Search on encrypted data using indexes is extensively studied in the literature [18]–[21], with different efficiencies and security guarantees. We base our secure index construction on the techniques introduced in [19] that are of stronger security guarantee and improved efficiency.

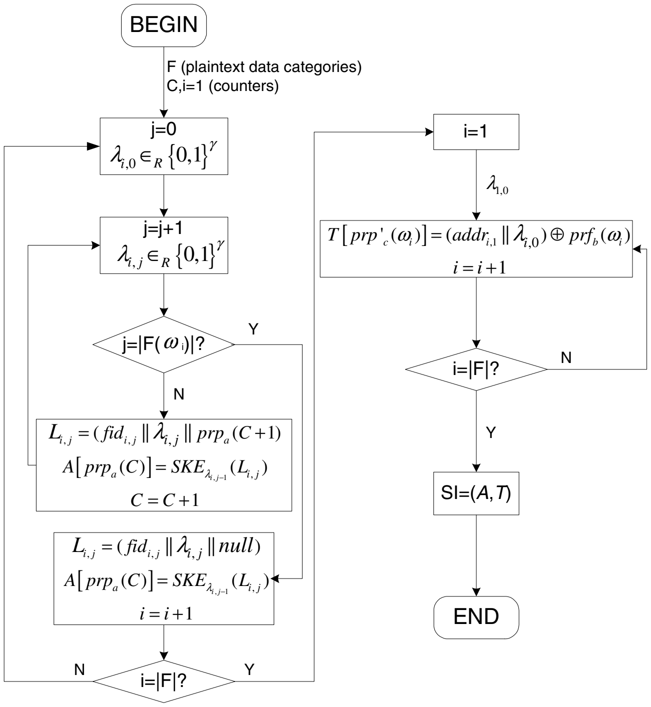
BEGIN
F (plaintext data categories)
C,i=1 (counters)
j=0
$\lambda_{i,0} \in_R \{0,1\}^\gamma$
j=j+1
$\lambda_{i,j} \in_R \{0,1\}^\gamma$
j=|F(ω_i)|?
Y
N
$L_{i,j} = (fid_{i,j} \parallel \lambda_{i,j} \parallel prp_a(C+1)$
$A[prp_a(C)] = SKE_{\lambda_{i,j-1}}(L_{i,j})$
$C = C+1$
$L_{i,j} = (fid_{i,j} \parallel \lambda_{i,j} \parallel null)$
$A[prp_a(C)] = SKE_{\lambda_{i,j-1}}(L_{i,j})$
$i = i+1$
N
i=|F|?
Y
i=1
$\lambda_{1,0}$
$T[prp'_c(\omega_i)] = (addr_{i,1} \parallel \lambda_{i,0}) \oplus prf_b(\omega_i)$
$i = i+1$
i=|F|?
N
Y
SI=(A,T)
END

Fig. 2. Construction of Secure Index by Data Owner.

The most important data structure in the secure index is the linked list L_i. It is a pointer structure used in search schemes [19], [20]. As shown in Fig. 2, the subscripts i and j in $L_{i,j}$ denote the ith linked list stored (encrypted) in the array A, and the jth node in the linked list L_i, respectively. A node $L_{i,j}$ consists of a data category identifier fid, secret key λ for encrypting the next node $L_{i,j+1}$ in L_i, and pointer $prp_a(C+1)$ which is the output of a pseudorandom permutation prp (with secret input a). The pointer points to the next node, $L_{i,j+1}$, in L_i. The array address $addr_{i,1}$ of the first node in each linked list (e.g., $L_{i,1}$) is stored in a lookup table T, together with the secret key $\lambda_{i,0}$ for decrypting the (encrypted) first node $SKE_{\lambda_{i,0}}(L_{i,1})$. The trapdoor for queries in this case is $(prp'_c(\omega_i), prf_b(\omega_i))$ where prp'_c and prf_b are pseudorandom permutation with secret input c and pseudorandom function with secret input b, respectively. A member in role r specifies this trapdoor (on keyword ω_i), by which the address $addr_{i,1}$ and decryption key $\lambda_{i,0}$ will be recovered by the storage site for locating and decrypting the linked list L_i (or all nodes $L_{i,j}$'s in L_i). The nodes $L_{i,j}$'s point to the addresses of the encrypted data categories containing ω_i (i.e., $SKE^{\omega_i}_{\alpha_r}(Data\_Category)$'s). We can see that in order to compute the trapdoor, the data owner's secrets (b, c) are needed. Therefore, the trapdoors are stored by the data owner in ciphertext (as shown in Step 6 of our scheme) because they cannot be computed by members without knowing (b, c). In addition, in the above figure, $F(\omega_i)$ denotes the identifiers of data categories containing keyword ω_i from the entire plaintext data categories $\mathbf{F}$, $addr_{i,1}$ and $A[prp_a(C)]$ denote the address of $L_{i,1}$ in A and the element stored at address $prp_a(C)$ of A, respectively, and γ is a security parameter.

Note that a data category is assumed in this paper as a document which cannot be further decomposed for search purposes. Moreover, a single secure index is built on the entire set of plaintext data categories to improve efficiency, since building secure index can be costly. This will not cause the problem of retrieving unwanted data (i.e., data intended for other roles) due to the employment of the role-based PEKS as the general access control (more details below). When the data owner has new data to update, the secure index can be updated correspondingly [19], [20]. Due to space limitations, we do not further elaborate on the secure index construction. Interested readers are referred to [19] for more details on the techniques and parameter definitions.

3) Discussion: The design rationale for searchable private data is now clear from the above descriptions. The final stored items in Steps 6 and 7 in the proposed privacy preservation scheme are the key elements enabling efficient search and revocation which are the major objectives of this paper. To be specific, the items in Steps 6 and 7 are stored in a tiered fashion, with the first tier being the item in Step 6, i.e., the PEKS-encrypted message. It serves as a general index to accessing a data owner's private data. Its role-based nature also confines the retrieval of useful information (i.e., $BE_{U\backslash R}(\alpha_r) \parallel SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$ in Step 6) to only the specified role. The PEKS-encrypted message has a fixed keyword (e.g., $kw = Alice's\_Family$) that will lead a member with the proper role to $BE_{U\backslash R}(\alpha_r)$ and $SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$, where the BE-encrypted message is used to obtain the up-to-date secret key α_r. This key is necessary for decrypting the SKE-encrypted message to acquire the keyword/trapdoor set $(\omega_1, TD_1), \cdots, (\omega_k, TD_k)$. The reason for encrypting the secret key α_r with broadcast encryption is the dynamic revocation functionality offered by this primitive.

The keyword/trapdoor set serves as the second index to the retrieval of the encrypted private data $SKE_{\alpha_r}(Data\_Category)$'s in Step 7, with the aid of the secure index (SI) stored also in Step 7. Besides the security property (i.e., data categories containing certain keyword from a data owner are unlinkable), the SI in our proposed scheme also provides efficient storage and search. Due to the pointer data structure used in the SI, each encrypted data category $SKE_{\alpha_r}(Data\_Category)$ need be stored only once while multiple linked lists can yield the address of $SKE_{\alpha_r}(Data\_Category)$ for multiple keywords contained in $Data\_Category$. Using the SI, the storage site can perform searches in $O(1)$ time for each returned data category containing a keyword. The total search time is proportional to the magnitude of the set of data categories containing a keyword. This high efficiency renders the proposed scheme very affordable. Without the second tier (i.e., the keyword/trapdoor set and the SI) for searching over the encrypted private data, it will be very difficult, if not impossible, to realize the searchability feature of the proposed scheme. The reason is that if only one tier is used to store the private data, the data owner will have to spell out all combinations of possible keywords to create a set of PEKS-encrypted messages for each combination. The resulting encrypted private data corresponding to each combination of keywords will then be concatenated to the set of PEKS-encrypted messages and the BE-encrypted message, e.g., $PEKS_\sigma(\breve{ID}_{role}, kw_1) \parallel \cdots \parallel PEKS_\sigma(\breve{ID}_{role}, kw_i) \parallel BE_{U\backslash R}(\alpha_r) \parallel SKE_{\alpha_r}(Data\_Category_1) \parallel \cdots \parallel SKE_{\alpha_r}(Data\_Category_j)$, where $(kw_1, \cdots, kw_i)$ is a keyword combination with $1 \leq i \leq k$ and $k = |KW|$ (KW was defined before as the keyword space), $(Data\_Category_1, \cdots, Data\_Category_j)$ is the set of data categories containing the above combination of keywords with $1 \leq j \leq n$ and $n = |\mathbf{F}|$ ($\mathbf{F}$ was defined before as the set of data categories).

E. Secure Communication Backbone

A secure communication backbone is an essential element in the design of secure networks to ensure authentication, data confidentiality, and data integrity (i.e., authentic data are not modified illegally). In the proposed scheme for online social networks, the secure backbone is needed when 1) the credential authority issues credentials ID_*/Ω_* to each and every user $*$ in the OSN domain, 2) the data owner adds a new member to his/her subgroup where authentication and key establishment may take place, 3) the data owner distributes the role-based credentials $\breve{ID}_r/\breve{\Omega}_r$ and secret Γ_u to each member u in role r for broadcast encryption, and 4) the final data are stored by the owner to the storage site where the integrity of the data should be guaranteed (by digital signatures, cf. Section V). In the above descriptions, key establishment occurs when the parties involved in authentication need establish shared secret key to facilitate further efficient communications.

Domain setup. Identity-based (ID-based) public key infrastructure (PKI) [17] should be employed in the proposed scheme, since the role-based encryption relies on the unique property of ID-based PKI, i.e., both the public key and the corresponding private key can be assigned posterior to the ID-based public key encryption using $\breve{ID}_r$, so long as this public key is known at the time of encryption. The ID-based PKI also enables authentication and key establishment and is used by many security applications in the literature [22], [23]. It serves as the cryptosystem to setup the secure backbone of our OSN. We apply the domain initialization of the ID-based cryptography [17]. Specifically, the credential authority acts as the PKG (private key generator) and performs the following domain initialization algorithm when the network is bootstrapped, where P_0 is a generator of G_1.

1) Input security parameter $\xi \in Z^+$ into domain parameter generator and output the parameter tuple $(q, G_1, G_2, e, P_0, H_0)$.
2) Randomly select a domain master secret $s_0 \in Z_q^*$ and calculate the domain public key $P_{pub} = s_0 P_0$.

The PKG publishes the domain parameters $(q, G_1, G_2, e, P_0, H_0, P_{pub})$ and maintains s_0 confidential, where H_0 is a hash function defined as $H_0 : \{0,1\}^* \rightarrow G_1$, $P_0 \in G_1$ is a generator of G_1, and the remaining parameters are defined in Section II A. After the domain is initialized, the credential distributes a public/private key pair to each and every user $*$ in the domain as $ID_*/s_0 \cdot H_0(ID_*)$. A similar domain initialization procedure is executed by each data owner (with a different set of domain parameters) for the distribution of the role-based public/private key pair.

V. Analysis and Enhancements for The Proposed Scheme in The OSN Context

This section elaborates on how the security requirements are achieved in our OSN context, and how to enhance the resilience of the proposed scheme to some attacks.

Data privacy: It is regarding the final stored data $PEKS_\sigma(\breve{ID}_{role}, kw) \parallel BE_{U\setminus R}(\alpha_r) \parallel SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$ and $SI \parallel SKE_{\alpha_r}(Data\_Category)$'s. Although the actual plaintext data for which the privacy should be protected are the data owner's personal data $Data\_Category$'s, other items which are linked to $Data\_Category$'s may be exploited to access or deduce the content of $Data\_Category$'s if they are not properly protected. For instance, the subgroup secret key α_r, if not protected by $BE_{U\setminus R}(\alpha_r)$, can be used to decrypt $SKE_{\alpha_r}(Data\_Category)$'s. However, since these data items are stored in ciphertext as a result of different encryption schemes, unauthorized users who are not in possession of the proper key for decrypting the corresponding ciphertext cannot access the plaintext data. Using the above example, a user v without the secret Γ_v is unable to obtain α_r from $BE_{U\setminus R}(\alpha_r)$, where $v \in R$ or $v \in \overline{U}$ ($\overline{U}$ is the set of non-members). Another desirable property of the encryption schemes is that the produced ciphertext appears random and thus unlinkable, preventing any entity without proper keys from linking multiple $SKE_{\alpha_r}(Data\_Category)$'s to profile the data owner. The secure index also ensures that no useful information can be obtained by the storage site to profile the data owner or members.

Access control: An unauthorized user v not in the subgroup of role r will not be granted the key pair $\breve{ID}_r/\breve{\Omega}_r$. Consequently, the trapdoor cannot be computed for the PEKS-encrypted data to search for $BE_{U\setminus R}(\alpha_r) \parallel SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$ that is necessary in further retrieving $SI \parallel SKE_{\alpha_r}(Data\_Category)$. There is a subtle collusion attack in which the storage site illegally returns data to a colluding member without the member's query (i.e., the storage site skips the requirement for a trapdoor and simply returns an arbitrary bulk of encrypted data). It is possible when neither colluder has the legitimate role-based credentials. However, this attack produces no impact on our scheme due to the fact that obtaining the ciphertext $BE_{U\setminus R}(\alpha_r) \parallel SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$ does not increase the colluders' chance to compromise data privacy. The reason is that the colluders are not able to obtain the secret Γ_* or α_r for the above ciphertext if they are not members of the role-based subgroup. A possible meaningful attack would be to further collude with an insider (i.e., a revoked or non-revoked member), which will be discussed shortly.

As defined in Section III B, access control should also be exercised on revoked members who were once authorized. This is mainly achieved by broadcast encryption which ensures that revoked members will no long acquire the correct secret key α_r', to search (based on $(\omega_1, TD_1), \cdots, (\omega_k, TD_k)$) or decrypt $SKE_{\alpha_r}(Data\_Category)$'s. It is effective in restricting the revoked members' access rights to the owner's data updates encrypted under the new key α_r'. Nevertheless, the owner's previous data encrypted under the revoked key α_r are still accessible to revoked members in the role r. In fact, this problem exists in any revocation mechanism in the sense that users are allowed access to data and services before the revocation. After all, the users are privileged for such access when they are not revoked.

This problem does not pose serious threats to the data privacy in our scheme. In reality, given limited storage space, the storage site will keep the data for a period of time, and erases old data to make room for new data. In addition, users of online social networks tend to update their data frequent enough to maintain social relations and stay connected with their contacts. Therefore, it is likely that the data accessible to revoked members are not available at the storage site any more. To enhance the privacy preservation scheme, however, the data owner can set important or highly sensitive data to view-only and disable the download option. In fact, it is observed that certain social contacts such as family members, relatives, and close friends, are unlikely to be revoked. These contacts normally have the access rights to the most private data. In this case, it is acceptable if the less sensitive or important data are exposed to revoked members. If it is undesirable for the data owner to leave the old data accessible to revoked users, the data owner can actively instruct the storage site to delete those data by providing the site with trapdoors $(TD_1, \cdots, TD_k)$ for locating the associated old data $SKE_{\alpha_r}(Data\_Category)$'s.

Authentication: It is assured by the cryptographic domains managed by the credential authority and each data owner, based on the ID-based PKI. When a user who is not an acquaintance of the data owner (e.g., a friend made online) requests to become a new member, the data owner needs to verify that this user is registered with the credential authority (i.e., this user is legitimate in the OSN). The data owner also needs to verify that this user is who he/she claims to be (i.e., the authenticity of the user). The authenticity verification is generally required for all roles in the data owner's group. For example, it is required when the data owner distributes the role-based credentials and the secret Γ_* for broadcast encryption to the members. The credential authority-issued public/private key pair ID_*/Ω_* which reflects a user's identity, can be used to fulfill the above verification tasks by means of a secure digital signature scheme [24]. An implicit authentication takes place when a member retrieves the information $(BE_{U\setminus R}(\alpha_r) \parallel SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k)))$ associated with the PEKS-encrypted data, where the member uses his/her role-based public/private key pair $\breve{ID}_r/\breve{\Omega}_r$ to compute the trapdoor for the PEKS-encrypted data. As explained before, there is no need for anonymous authentication in the OSN context since the data owner and his/her member know the identity of each other. Note that we do not consider authentication between the storage site and the data owner/member, because the storage site provided by a third party is not a user of the OSN domain and is simply untrusted for the worst-case scenario.

Collusion attacks and countermeasures: As specified in the threat model, legitimate members can collude with the storage site, revoked members, or other legitimate members of different roles. Collusion is a very powerful attack and will severely threaten the security of our scheme if it is not defeated. Since the data privacy requirement guarantees that the storage site cannot learn any useful information about the private data it stores, colluding with the storage site will not result in any gain to users. However, the storage site can benefit from the collusion with any legitimate member(s) to gain access to the private data. In the case of rational colluders where one colludes only if he/she can gain, the collusion attack with storage site will not occur. Nonetheless, we do not assume rational colluders and consider the general case. We have the following observations. In effective collusion attacks against our scheme, at least one colluder should be an insider or legitimate member, who we call the traitor following the cryptographic convention. The other colluders who do not have access rights are called pirates. The purpose of the collusion attacks is for the pirates to access the encrypted private data, which, upon success, will undermine the data privacy guarantee and the access control of our scheme.

In fact, the effective collusion attacks present in our context are nothing strange but fall into a broad class of attacks named data piracy. In a data piracy, the traitor(s) can either share copies of digital data directly with pirates, or reveal the decryption keys to pirates for them to freely access any data intended for legitimate users. As a result, the countermeasure for data piracy, namely, traitor tracing [25], can be readily employed in our scheme to combat the collusion attacks. In a traitor tracing scheme, the traitor(s) can be traced and held responsible for any privacy leakage. It should be noted that employing a traitor tracing scheme, especially a more secure one (e.g., k-resilient traitor tracing [26]), incurs high computational complexity. It directly results from the power of collusion attacks (or piracy) and the difficulty in combating them. In real applications, there is always tradeoff between security and complexity. Due to limited space, we will not elaborate on the traitor tracing scheme. Interested readers are referred to the literature [25]–[27] and the references therein for a variety of such schemes.

Other attacks and countermeasures: These are launched mainly to compromise data confidentiality and integrity, and to impersonate a legitimate user. We have mentioned in Section III B that data privacy implies confidentiality. The guarantee for data privacy thereby also assures data confidentiality. Data integrity can be protected by having the data owner digitally sign the stored data as: $SIG_{\Omega_{Data\_Owner}}(PEKS_{\sigma}(\breve{ID}_{role}, kw) \parallel BE_{U \backslash R}(\alpha_r) \parallel SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k)))$, and $SIG_{\Omega_{Data\_Owner}}(SI \parallel SKE_{\alpha_r}(Data\_Category))$, where $SIG_{\Omega_{Data\_Owner}}$ denotes the digital signature generated by the data owner using his/her identity-based private key Ω_{Data_Owner} assigned by the credential authority. A possible instantiation of secure digital signature scheme can be found in [24]. In an impersonation attack, the attacker claims to be a legitimate user in order to spoof the communicating party, which is similar to the crime of identity theft in reality. Successful impersonation attacks are caused by the lack of authentication, and hence are not present in the proposed scheme.

VI. Application of The Proposed Scheme to Anonymous Online Social Networks

In this section, we extend the application to anonymous online social networks and show the suitability of the proposed privacy preservation scheme for such OSNs that have more stringent security requirements.

A. Overview

OSN applications such as Facebook that have been discussed so far are not anonymous in nature, in the sense that members in social groups know the identity of one another in order to create their groups of contacts (e.g., "*Family*", "*coworker*"). Furthermore, private or personal data are identity-revealing in nature (e.g., photos or videos showing the face and location of the data owner). By allowing access to these data, the data owner willingly reveals his/her identity to the contacts.

Another type of online social networks store data such as computer programs, e-books, audio files, and videos for interested user groups to access, such as YouTube, Napster, BitTorrent, and various online chatting rooms and forums, where users can opt to retain their anonymity and privacy (provided that the shared data contain no identifying information). We call this type of online social networks anonymous OSNs. In anonymous OSNs, shared data are stored in the storage site (e.g., central servers, peers), which inevitably involves secure and efficient search and retrieval, as in the basic OSNs. As a result, the proposed privacy preservation scheme can be easily applied in the anonymous OSNs. Nonetheless, since the focus has been extended from preserving data privacy in basic OSNs to also preserving user privacy in anonymous OSNs, some technical details of the proposed scheme must be modified to suit the new application scenario. However, the core design rationale and key technical components remain the same for both application scenarios, indicating the potential general applicability of our design to OSNs with individual features and security requirements.

B. Modified Privacy Preservation Scheme for Anonymous OSNs

In the modified scheme, the credential authority is a service provider of anonymous OSNs, e.g., the administrator of YouTube, who is in charge of the service domain. Users register for the service (e.g., registered users who upload videos to YouTube) and form one single group. For the purpose of demonstration, we consider in this paper only the general case where no special classification of users is needed as opposed to basic OSNs. All users in a service provider's domain take on the same role, e.g., "*YouTube User*". Note that we do not consider users who use the service without registration (i.e., users with no role in the application), such as those who only browse videos on YouTube without sharing their own data (e.g., videos, comments), since it is not a typical scenario in anonymous data sharing applications.

1) Description: We will use notations from the original scheme and spell out the differences in the modified scheme shown in Fig. 3, where → denotes "sends to", CA stands for credential authority, and Ψ denotes the set of secrets $\{a, b, c\}$ used to construct the secure index and trapdoors (cf. Section IV D). As indicated in Fig. 3, the main difference in the technical procedure between the original and modified schemes is two-fold. First of all, the difference in the two application scenarios contributes to some functional variation. The data owner in the original scheme is responsible for both assigning roles to the members and storing data to share. Whereas in the modified

1. $CA \rightarrow User$: ID_{role}, Ω_{role};
2. $CA \rightarrow Storage$: $PEKS_{\sigma}(ID_{role}, kw) \parallel BE_{U \setminus R}(\alpha_r \parallel \Psi) \parallel SKE_{\alpha_r}(\Psi)$;
3. $User \rightarrow Storage$ (*storing*): $SI_{\Psi} \parallel SKE_{\alpha_r}(Data\_Category)$;
4. $User \rightarrow Storage$ (*retrieving*): $TD_{\Psi}(\omega_1), \cdots, TD_{\Psi}(\omega_k)$.

Fig. 3. The Modified Privacy Preservation Scheme.

scheme, the service provider (or credential authority) assigns the role to users among whom the data are shared. As a result, the item used as the general index $PEKS_{\sigma}(ID_{role}, kw)$ and the item for revocation $BE_{U \setminus R}(\alpha_r \parallel \Psi)$ are stored by the credential authority on the untrusted storage site. For instance, the credential authority of YouTube performs PEKS-encryption as $PEKS_{\sigma}(ID_{\text{“YouTubeUser”}}, YouTube\_Data)$. Moreover, there is only one α_r in the system for the general role of "*YouTube User*". The shared data and associated secure index are stored by the user who owns the data, shown in Step 3 of Fig. 3, with the corresponding retrieval by other users shown in Step 4.

Second, together with the functional variation is the change in technical details. In Step 2, $SKE_{\alpha_r}(\Psi)$ is stored with the two encryptions instead of $SKE_{\alpha_r}((\omega_1, TD_1), \cdots, (\omega_k, TD_k))$ in the original scheme. The reason is that the credential authority is not the owner of data and thus cannot generate keyword/trapdoor pairs $(\omega_1, TD_1), \cdots, (\omega_k, TD_k)$ for searching over the stored data. Without proper trapdoors generated by the owner of the data, other users are unable to search over $SI_{\Psi} \parallel SKE_{\alpha_r}(Data\_Category)$ for the shared data (denoted by $Data\_Category$). Therefore, we propose for the credential authority to generate the secret Ψ used for computing the trapdoors and constructing the secure index. The secret is then distributed to non-revoked users through the BE-encrypted message $BE_{U \setminus R}(\alpha_r \parallel \Psi)$ in the same way that α_r is distributed. It ensures that a revoked user can perform neither correct retrieval nor useful storage (i.e., the keywords used to build the secure index will be different from those used by other users to compute trapdoors). The credential authority can publish the dictionary of all keywords for users to choose from and compute the corresponding trapdoors when performing searches.

It should be noted that although insensitive data are shared which need not be encrypted, storing plaintext data will enable the untrusted server to link multiple data files to a same owner (by data content) and thereby destroy the linkability property of user privacy. The encryption applied on $Data\_Category$ is primarily for preserving user privacy in anonymous OSNs, rather than protecting data privacy as in OSNs. Due to the space limitation, we will not further discuss other design considerations on incorporating the modified scheme into an anonymous OSN application which are covered elsewhere.

VII. Conclusion and Future Work

In this paper, we proposed a privacy-preserving scheme for online social networks (OSNs). Our scheme enables secure and efficient searches over shared data within social groups, and facilitates dynamic revocation in face of the frequently changing group membership. It offers a functional, secure, and sound solution for data sharing in social group applications, and is shown to be applicable to anonymous OSNs with more stringent security requirements. Our future work includes the implementation of the proposed scheme as a plug-in for Facebook, and the experiments on the efficacy and performance of our scheme in practical applications.

References

[1] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," *Proc. ACM SIGCOMM*, Aug. 2009.

[2] S. Guha, K. Tang, and P. Francis, "NOYB: privacy in online social networks," *An ACM SIGCOMM 2008 Workshop (WOSN'08)*, pp. 49–54, 2008.

[3] S. Yardi, N. Feamster, and A. Bruckman, "Photo-based authentication using social networks," *An ACM SIGCOMM 2008 Workshop (WOSN'08)*, pp. 55–59, Aug. 2008.

[4] J. D.-Ferrer, A. Viejo, F. Sebé, and Ú. G.-Nicolás, "Privacy homomorphisms for social networks with private relationship," *Comput. Netw.*, pp. 3007–3016, 2008.

[5] L. A. Cutillo and R. Molva, "Safebook: a privacy-preserving online social network leveraging on real-life trust," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 94–101, Dec. 2009.

[6] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "PeerSoN: p2p social networking," *Proc. 2nd ACM EuroSys Workshop on Social Network Systems*, pp. 46–52, 2009.

[7] G. Danezis and P. Mittal, "Sybilinfer: detecting sybil nodes using social networks," *in Proc. Network and Distributed System Security Symposium (NDSS)*, Feb. 2009.

[8] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: a near-optimal social network defense against sybil attacks," *in Proc. IEEE Symposium on Security and Privacy*, pp. 3–17, Oct. 2008.

[9] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," *SIGCOMM, Computing Communication Review*, vol. 36, no. 4, pp. 267–278, 2006.

[10] C. Diaz, C. Troncoso, and A. Serjantov, *On the impact of social network profiling on anonymity*, in N. Borisov and I. Goldberg (Eds.): PETS 2008, LNCS 5134, 2008.

[11] M. Reiter and A. Rubin, "Crowds: anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, June 1998.

[12] "The freenet project," *http://freenetproject.org/*.

[13] M. Freedman and R. Morris, "Tarzan: a peer-to-peer anonymizing network layer," *in ACM Conference on Computer and Communications Security (CCS)*, pp. 193–206, Nov. 2002.

[14] D. Boneh and M. Franklin, *Identity-based encryption from the weil pairings*, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp. 514-532, Springer-Verlag, 2001.

[15] A. Fiat and M. Naor, *Broadcast Encryption*, in Advances in Cryptology: CRYPTO'93, LNCS 773, 1993.

[16] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," *in EUROCRYPT 2004, LNCS 3027. Springer*, 2004.

[17] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing. extended abstract in CRYPTO 2001," *SIAM J. of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[18] E.-J. Goh, *Secure indexes*, in Cryptology ePrint Archive: Report 2003/216, available at http://eprint.iacr.org/2003/216/, 2003.

[19] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *in ACM Conference on Computer and Communications Security (CCS)*, 2006.

[20] Y. C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," *in Applied Cryptography and Network Security Conference*, 2005.

[21] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searching on encrypted data," *in Proc. IEEE Symposium on Security and Privacy*, pp. 44–55, 2000.

[22] J. Sun, C. Zhang, and Y. Fang, "A security architecture achieving anonymity and traceability in wireless mesh networks," *IEEE Conf. on Computer Communications (INFOCOM)*, pp. 1687–1695, Apr. 2008.

[23] J. Sun and Y. Fang, "Defense against misbehavior in anonymous vehicular ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1515–1525, Nov. 2009.

[24] F. Hess, *Efficient identity-based signature schemes based on pairings*, SAC 2002, LNCS 2595, pp. 310-324, Springer-Verlag, 2002.

[25] B. Chor, A. Fiat, and M. Naor, *Tracing Traitors*, in Advances in Cryptology: CRYPTO'94, LNCS 839, 1994.

[26] M. Naor and B. Pinkas, *Threshold Traitor Tracing*, in Advances in Cryptology: CRYPTO'98, LNCS 1462, 1998.

[27] D. Boneh and M. Franklin, *An Efficient Public Key Traitor Tracing Scheme*, in Advances in Cryptology: CRYPTO'99, LNCS 1666, 1999.