Encryption for Peer-to-Peer Social Networks

Oleksandr Bodriagov
School of Computer Science and Communication
KTH - The Royal Institute of Technology
Stockholm, Sweden
obo@kth.se

Sonja Buchegger
School of Computer Science and Communication
KTH - The Royal Institute of Technology
Stockholm, Sweden
buc@csc.kth.se

***Abstract*—To address privacy concerns over online social networking services, several distributed alternatives have been proposed. These peer-to-peer (P2P) online social networks do not rely on centralized storage of user data. Instead, data can be stored not only on a computer of a profile owner but almost anywhere (friends' computers, random peers from the social network, third-party external storage, etc.). Since the external storage is often untrusted or only semi-trusted, encryption plays a fundamental role in security of P2P social networks.**

Encryption, however, also adds some overhead in both the time and space domains. To be scalable, a system that relies heavily on encryption should use as efficient algorithms as possible. It also needs to provide the functionality of changing access rights at reasonable cost, and, crucially, the system should preserve privacy properties itself. That is, beyond user data confidentiality, it has to protect against information leakage about users' access rights and traffic analysis.

In this paper we explore the requirements of encryption for P2P social networks in detail and propose a list of criteria for evaluation. We then compare a set of approaches from the literature according to these criteria. We find that none of the current P2P architectures for social networks manages to achieve secure, efficient, 24/7 access control enforcement and data storage. They either rely on trust, require constantly running servers for each user, use expensive encryption, or fail to protect privacy of access information. In the search for a solution that better fulfills the criteria, we found that some broadcast encryption (BE) schemes exhibit several desirable properties.

We thus propose to use BE schemes with high performance encryption/decryption regardless of the number of identities/groups for an efficient encryption-based access control in the P2P environment. We define relevant properties for the BE schemes to be used in the P2P social network scenario and describe advantages that such schemes have compared to encryption techniques used in existing P2P architectures.

I. Introduction

In current online social networks (OSN), users do not have complete control over who can access their data. While most OSN services provide some privacy settings to limit the audience of content published by the user, some default settings make content public. Other privacy settings can be overriden by the user's friends' decisions, for example when granting access to a third-party application. More importantly, there is no real protection against access by the service provider itself, both to user-generated content and to inadvertently generated information, such as behavioral patterns in linking, messaging, interacting, commenting, logging on and off, locations, browser types, operating systems used, among others. The content can then be mined and used for targeted advertising or be released to third parties. Whether this is done and to what extent the users' privacy preferences are honored, depends primarily on the privacy agreements of the service provider and other legal issues such as the location of the service provider, the servers, the content, or the user.

To prevent such undesired disclosure of user data, there have been efforts to circumvent the OSN service providers and give the control over the data back to the users. While some proposals use the existing infrastructure of the OSN provider, others decentralize control and take a peer-to-peer approach. In this paper, we focus on the latter type of solutions, and more narrowly on those that enforce privacy policies by cryptographic means. We use the term *encryption* throughout the paper as a shorthand for this concept, including key management and other mechanisms needed.

Access control based on encryption is fundamentally different from that in existing centralized, provider-dependent social networks, as the centralized provider has control over how their servers behave, given configurations and security measures, and can enforce policies using the operating system. If all users had their own constantly running servers, then a P2P social network could be achieved via a direct end-to-end communication and access control would be the similar to the centralized case (though performed locally on each server). Currently, however, this is an unrealistic assumption and enforcement needs to happen on a different level. Encryption is a way to do that in a P2P environment with untrusted/semi-trusted storage, and several P2P OSN proposals use it. In this paper, we discuss and compare several prominent approaches: PeerSoN, Safebook, Persona, Diaspora, and our new proposal using identity-based broadcast encryption (IBBE).

In order to come up with a good solution for how to effectively and efficiently use encryption for access control and privacy policy enforcement in P2P social networks, we first need to define requirements for such systems. We discuss these in this paper and group them into the following categories: efficiency, functionality, and privacy. By efficiency we mean how much effort the used encryption scheme creates in terms of storage, computational cost, and communications overhead. By functionality we categorize possibilities of using the encryption scheme to manage permissions. By privacy we denote the side-effects of the distributed system of leaking information *about* the user data and not only the user data itself (confidentiality).

The rest of the paper is organized as follows. First we state the criteria that are crucial for the encryption schemes in P2P social networks, then we describe existing P2P social network architectures and what encryption they use. We continue by evaluating existing encryption schemes according to the stated criteria. Then we explain broadcast encryption and how it works in a P2P social network and evaluate it according to the criteria. We summarize the results of our evaluation in a table. We finish by drawing some conclusions.

II. Essential criteria for the P2P encryption systems

The P2P environment and the absence of a trusted party put many security constraints on the encryption-based access control system. Moreover, for decent usability it is imperative that all actions are executed fast enough in order to achieve a positive user experience. In this section we analyze these constraints and afterward state requirements for encryption systems.

A. *Efficiency*

In the ordinary centralized access control the security subsystem authenticates the user and enforces policies given by access control lists (ACLs) or capabilities. In contrast, in the P2P system we cannot rely on the untrusted storage for authentication and authorization. We do not have access to the operating system of a replica holder and thus use encryption and key management to replace this functionality. Encryption-based access control relies on authentication during a key setup phase, when a decryption key is given to the user after authentication. The key has a similar role to the access token in systems such as Kerberos, while the encryption scheme in P2P social networks plays the role of the security subsystem in centralized systems in a sense that it takes a user's key and authorizes access to the data.

Access tokens have a short lifetime and can be easily renewed, while users' keys are given out for a much longer period and there is therefore a higher probability that they might be stolen or lost. Cryptographic keys are also prone to aging. And although the user key renovation event is not so frequent, it can have big consequences because of the fined-grained access control requirement.

To achieve fine-grained access control, each object should be encrypted separately for different sets of recipients, such that encrypted objects are completely unrelated and a change in one of them does not influence the others. That is why, all objects to which this key gives access should be re-encrypted during a key renovation procedure, which is clearly a performance issue. It might be better to do such a procedure for all the user's keys in the system at once, so that a single object gets re-encrypted only once. With a large number of objects, however, re-encryption of all data might be quite time consuming. It is thus important to have fast encryption. Also from the usability perspective the speed of encryption/decryption has great importance. Operations like posting a single message, a photo, etc. even with inefficient encryption will not take much time, but retrieval of recent wall posts, messages, whole photo albums can be more time consuming. The speed of encryption/decryption depends not only on the speed of the underlying cipher, but also on the scalability of the scheme. Therefore, the first requirement is a constant cost encryption/decryption that does not depend on the number of recipients. To the best of our knowledge, there are, however, no encryption schemes that have both encryption and decryption that do not depend on the number of recipients.

In the centralized system an addition/removal of a group member influences all objects to which this group has access, but this is not generally true for encryption-based access control systems. Some encryption schemes require all objects to be re-encrypted if the group changes. Such behavior is not scalable and might have a strong impact especially in P2P networks because the number of objects (posts, photos, etc.) can be quite big and groups can be changed quite often (e.g. addition of new friends). Thus, the second requirement is that addition/removal of users from a group should not depend on the number of subjects/objects and should have constant cost as in centralized systems. If the encryption system does not have constant cost addition/removal of users from a group, then re-encryption should be as fast as possible.

Another issue is encryption overhead in terms of storage. For P2P storage with replication it is crucial to save as much space as possible, because otherwise the system will not be scalable. The encryption overhead (headers) can be quite considerable for short messages and may require more space than the encrypted data itself. If the size of the header depends on the number of receivers, then such encryption scheme is not suitable for a P2P social network with considerable amount of possible recipients. Therefore, the next criterion is the scalability of the header in terms of the number of recipients. Another concern is the storage cost of the encrypted data itself.

B. *Functionality*

Different types of encryption schemes (symmetric, asymmetric, etc.) have different properties and thus can be used to realize different features of a P2P social network. Yet, the encryption system that combines different encryption schemes should be able to realize all functions of the social network. The encryption system defines the functionality of the P2P social network, the provided security and privacy levels.

A P2P social network's encryption system should be able to encrypt objects for a single subject as well as for any possible set of subjects in a cost-effective way. Efficient encryption for the conjunction/disjunction of groups, however, is not supported by all encryption systems. It is quite a useful operation, though, for users of social networks since users' connections can have different origins (colleagues, family, etc.) and different levels of trust. Such operations as encryption for a group that one is not a member of and encryption for "friends of friends" are even less frequently supported, though there are analogies to these operations in every-day life.

C. Privacy

The security subsystem in a centralized environment controls all flows of information from a single point of control. It is, however, much harder to implement such control with the encryption system in a P2P network with untrusted storage, because the content of the encrypted objects is not the only thing that requires protection. It is also important to protect information about which subjects have access to what objects, about the quantity of objects, about their type. Moreover, it should not be possible to verify if a particular user has access to some particular object. The user should be able to see only the objects that are encrypted for her or for the group that she is a member of. The requirement of fine-grained access control results in a set of separately encrypted objects. The users should be able to determine which files they are able to open without checking all the files, otherwise the system looses scalability. At the same time, the encryption header of the object should have the ability not to reveal subjects who have access to this object. If the access list goes along with the encrypted data, malicious users can completely reconstruct a network of contacts from those lists. Additionally, they will know who can access which encrypted objects and can infer some information from that knowledge.

D. List of Criteria

To sum up, we have come up with the following evaluation criteria: efficiency of addition/removal of users from a group, efficiency of user key revocation, encryption/decryption efficiency, encryption header overhead, ability to encrypt for the conjunction/disjunction of groups, ability to encrypt for a group that one is not a member of, ability to encrypt for "friends of friends", ability to not reveal access structures in the header.

III. Existing P2P OSN Architectures

In this section we describe and analyze existing P2P architectures for social networks.

An early version of the P2P social network developed under the PeerSoN project [1], [2] relied on the conjunction of symmetric and asymmetric cryptography. Data was first encrypted using a symmetric key and then this key was encrypted with public keys of recipients. Users Ids and encrypted symmetric keys were stored alongside the encrypted data.

Safebook [3] is based on two design principles: decentralization and exploitation of a real-life trust. It relies on matryoshkas which provide data storage, profile data retrieval, and communication obfuscation. A matryoshka consists of a set of nodes grouped in several concentric rings according to the level of trust that the node associated with the matryoshka has towards them. The innermost layer/ring is the most trusted and consists of 'friends'. It is actually responsible for storing replicated data for the node associated with the matryoshka. The innermost layer stores published data in encrypted and unencrypted forms, but private data is stored by the owner himself and is not replicated to the innermost layer. According to [4], a "*simple group-based encryption scheme*" is used for encryption, and users get opportune keys [3] to decrypt the published data. The owner should explicitly authorize and republish to the inner ring every message written by other users.

Anonymity in Safebook is achieved by using a multi-hop routing. A distributed hash table holds pointers to nodes on the outermost ring of the matryoshka. The incoming request is routed from the outermost ring to the core of the matryoshka. The rooted messages are encrypted on a hop-by-hop basis using asymmetric cryptography.

Diaspora [5] is a project that uses a client-server architecture but does it in a decentralized way. It requires a constantly running server for each user to achieve end-to-end communications. Users without servers can choose from one of the existing servers to store their data. To ensure confidentiality of the stored data, encryption is used. Not many people may be willing to run a server and provide storage for other users for free. Even if the user finds such a server there is no guarantee that server will not be shut-down later in the future potentially resulting in a complete loss of all data for the user. This risk is mitigated in systems that use multiple replicas held by peers, rather than having one instance as is done in Diaspora.

According to Diaspora's security architecture proposal there are 3 levels of data security [6]: unencrypted information that is available to everyone, information encrypted by the server for some intended receivers, information encrypted by the owner herself for some intended receivers.

The encryption process is executed in two stages [7]. First, a random encryption key is generated (symmetric cryptography) and the message is encrypted with this key. Then the sender encrypts this secret key for each of the receivers with a corresponding public key and sends it to them. Currently, an AES-256-CBC cipher is used for symmetric encryption and RSA for public key encryption [8].

Diaspora works according to the push model [9], [8]. Data posted by a user is encrypted for the recipients and pushed to the servers of recipients in encrypted form. To delete posted data, a *retraction* request is sent to the recipients' servers.

Another P2P architecture is Persona [10]. It stores data in encrypted form, thus access control is encryption based. As in PeerSoN, the storage is not trusted, confidentiality is ensured by encryption. To provide specific rights to stored objects the profile owner defines access control lists (ACLs) and instructs the storage to set them. ACLs contain public keys of users and their access rights. The storage authenticates the users and authorizes their actions based on the entries in the ACL. This scheme provides limited data integrity protection, but the credibility of access control enforced by untrusted storage is not that strong, so the main protection mechanism is encryption, and it ensures only confidentiality.

Persona relies on a ciphertext-policy attribute-based encryption (CP-ABE) [11] scheme. In CP-ABE a user's private key is associated with a number of attributes (e.g. 'friend', 'family'). During encryption an access structure over attributes is attached to a ciphertext. The user is able to decrypt the ciphertext if that user's attributes pass through the ciphertext's access structure.

Encryption in Persona is a two-stage process because ABE is computationally expensive. First data is encrypted with a symmetric key, and then it is ABE-encrypted.

IV. Evaluation of existing encryption schemes based on our criteria

In this section we evaluate the suitability of encryption systems in existing architectures for a P2P social network scenario. The evaluation is based on the criteria described in Section II.

As described in the previous section, existing P2P architectures for social networks have used the following two types of encryption systems: the conjunction of symmetric and asymmetric cryptography (i.e. trivial broadcast encryption scheme [12]) used in Diaspora and the early version of PeerSoN, and CP-ABE used in Persona. There is not enough information about the encryption system of Safebook, except that it is a "*simple group-based encryption scheme*" and users get opportune keys [3] to decrypt the published data. Besides, expensive asymmetric cryptography that is used for communication between hops inside of the matryoshka defines the time cost of information retrieval and posting by other users. The time required for encryption/decryption of the data itself is negligible compared to the time of information retrieval and posting. Other relevant drawbacks of Safebook are its reliance on the trust relationship, authorization and republishing of every message written by others.

A. Efficiency

It is well-known that asymmetric cryptography is much more computationally intensive than symmetric cryptography, and even Elliptic Curve Cryptography (ECC) that is the most efficient public-key cryptography [13], is much slower than symmetric cryptography. Moreover, the storage efficiency is inadequate because for each of the receivers the object has to be encrypted separately. That is why encryption systems based solely on asymmetric cryptography are not used.

Compared to a purely asymmetric cryptography scheme, an encryption system based on the conjunction of symmetric and asymmetric cryptography is much more efficient since the object itself is encrypted using a symmetric cipher, and then this symmetric key is encrypted multiple times with a public key of each of the receivers. Even such an approach is, however, not quite suitable because the number of objects in the typical profile is very big, thus the overhead connected with encryption of the same keys multiple times is quite significant both in terms of time and space. Encryption for the group means that data is first encrypted with a symmetric key and then this symmetric key is encrypted with a public key of each member. The addition of a user to a group is very simple and means encryption of the symmetric key(s) of that group with the public key of that user. Conversely, the removal of a user/a set of user from a group requires the re-encryption of all data encrypted for that group, which is considerably harder.

Both the early version of PeerSoN and Diaspora use this encryption system. In case of PeerSoN it is used in the Pull model, while in Diaspora it is used in the Push model which has the disadvantage that the same encrypted object has to be transferred multiple times, separately to each of the recipients.

The CP-ABE scheme used in Persona [14] was the first introduced CP-ABE scheme and has many drawbacks. The size of the cipher text and the speed of encryption/decryption are crucial parameters for P2P social network scenario, and in the original scheme they are linear in the number of attributes in the access structure, which is not adequate for a P2P social network. Besides, as far as we know, there are no CP-ABE schemes with constant size ciphertexts or decryption that do not depend on the number of attributes and have constant cost, and the encryption time in existing schemes scales linearly with the size of the access formula [15], [16]. To our knowledge, the most efficient CP-ABE schemes (e.g. scheme described in [16]) in terms of decryption are linear in the set of attributes from the user's key that satisfy the access structure. Encryption for one contact can be done using the public key of that person (or as in the case of conjunction of symmetric and asymmetric cryptography), while encryption to groups uses CP-ABE because of advantages in efficiency and functionality. An attribute defines a group, and to encrypt for the group one encrypts for that attribute. If all receivers have a common attribute, then CP-ABE is quite efficient from the ciphertext size point of view, but if the receivers are members of different groups then the overhead can be suboptimal (depending on the number of attributes) even though the encryption time is very favorable.

The CP-ABE scheme used in Persona is limited to monotonic access structures (no negations) which leads to inefficient encryption, for example in cases when access is allowed for the whole group except for a few members. It has troublesome user revocation since several different users might match the decryption policy [14] and there are no negations to prohibit access for some exact users. Revocation of a user's access rights in the worst case scenario basically means creation of a new group of users that corresponds to some new attribute that is in common for the members of this new group, and re-encryption of all data of the old group with new symmetric keys. To the best of our knowledge, the only CP-ABE scheme that allows negations is [17]. This scheme in conjunction with usage of identities as attributes yields simple revocation, but the decryption time and the storage cost are still linear in that scheme.

B. Functionality

The combination of symmetric and asymmetric cryptography allows for encrypting for the disjunction of several groups by encrypting the data with symmetric keys corresponding to those groups, but it is impossible to encrypt for the conjunction of groups. Encryption for some arbitrary set of users that do not belong to the same group is equivalent to creating a new group, since after the encryption these users will share the symmetric key used for encryption. We described the encryption procedure from the group creator/group member perspective, but it is impossible to encrypt for a group that one is not a member of unless the group has a public/private key pair shared among its members. Encryption for the ”friends of friends” is not supported either.

The CP-ABE schemes seamlessly support encryption for the conjunction/disjunction of two groups using the conjunction/disjunction of the attributes. They also allow encrypting for ”friends of friends”. Additionally, a user can encrypt to a group even if he is not a member of that group.

C. Privacy

In the early version of PeerSoN it was possible to find out which subjects could access what objects from the encryption headers that contained Ids of subjects. The cryptosystem based on conjunction of symmetric and asymmetric cryptography does, however, not need to reveal for whom the data is encrypted (the problem of identifying what one can decrypt can be solved by different means). In contrast, all CP-ABE schemes, as far as we know, have the access structures in clear, thus anyone who can download encrypted data can learn which groups have access to it. At the same time, there are BE schemes with hidden access structures, and the scheme described in [18] is one of them.

D. Summary

To summarize our evaluation, the encryption system that combines symmetric and asymmetric cryptography is quite efficient, but from the functionality point of view it is quite limited. The encryption system based on CP-ABE schemes is moderately adequate from the efficiency point of view, and it has quite rich functionality. However, since none of the CP-ABE schemes achieves non-monotonic, hidden access structures and low (constant) storage and computational cost at the same time, we propose to use an encryption system based on broadcast encryption schemes. We describe them in the following section.

V. Broadcast Encryption

Broadcast encryption (BE) schemes are used to distribute encrypted data to a dynamic set of users in a cost-effective way. In general, a BE scheme consists of a sender and a group of recipients. Each recipient has her own private decryption key to decrypt encrypted data sent by the sender.

BE schemes can either be symmetric or public-key based. In the first case, only a trusted source/broadcaster of the system that generated all the private keys can broadcast data to receivers. If the system is public-key based, then anyone who knows a public key of the system can broadcast.

The efficiency of BE schemes is measured in terms of transmission, storage, and computational cost. Besides efficiency, one of the main requirements for BE schemes is that it should be easy to revoke a key or a set of keys. Other important security concepts are collusion resistance and statelessness. A fully collusion-resistant scheme is robust against collusion of any number of revoked users. A BE scheme is said to be stateless if after revocation of some subset of users the remaining users do not have to update their private keys.

A BE is called dynamic [12] if new users can join without a need to modify existing users’ decryption keys, if the ciphertext size and the system’s initial key setup do not depend on the number of users, if for a symmetric key based scheme the encryption key should not be changed and for a public key scheme the group public key should be incrementally updated with complexity at most $O(1)$. Since dynamic BE schemes give so many advantages from the key management and efficiency perspectives, we will narrow the discussion to only this type of BE schemes.

It is obvious that the described properties of BE schemes are very desired, thus suitable candidates for application to a social network scenario are BE schemes with the following properties: stateless, fully collision resistant, dynamic, with constant size ciphertexts and keys, with computationally efficient encryption/decryption.

We use a dynamic identity-based broadcast encryption (IBBE) scheme that meets all these requirements [18]. Although IBBE schemes involve a third-party authority - a Private Key Generator (PKG), this role is given to the profile owner when adjusting this scheme for our scenario. Thus, the profile owner is responsible for creating a group of receivers and assigning private BE keys.

The IBBE scheme is formaly defined as a tuple of algorithms *IBBE = (Setup, Extract, Encrypt, Decrypt)* [19]. Although the DIBBE scheme defined in [18] has the same structure, there are some differences in the algorithms’ input parameters that reflect a dynamic nature of the scheme. The algorithms of the DIBBE scheme have the following form:

The *Setup* algorithm generates some system parameters, a secret master key, and a group public key *GPK*. The *Extract* algorithm takes as input a secret master key *MK* known only to the broadcaster and produces a private key for each user. The encryption algorithm *Encrypt* takes as input a set of receivers *S* and a group public key *GPK* (for the DIBBE scheme it additionally takes) and outputs a pair *(Header,K)*, where *K* is a symmetric secret key to encrypt data and *Header* is an encryption of this symmetric key for the set of receivers. Data is stored in the form *(Header, encrypted data)*, and only a user whose ID/label is in the set can decrypt the *Header* using his/her private key. Some schemes work with plain Headers that show who can decrypt the data, while other schemes are more privacy preserving and reveal no information about the set of receivers or any other parameters (e.g. scheme described in [20], [18]). The *Decrypt* algorithm for the privacy preserving scheme mentioned above takes *GPK*, *Header*, the user’s private key, and the user ID as input and outputs a symmetric key *K*; while for the ordinary IBBE schemes the *Decrypt* algorithm additionally requires the set of receivers *S*.

Since each time during encryption the user can create a set of receivers on the fly in the IBBE scheme, it is possible to encrypt to any conjunction/disjunction of groups because a group is merely an arbitrary set of users. Besides, in IBBE schemes users that are not members of the group can still encrypt to the group if they know *GPK*. However, the DIBBE scheme defined in [18] requires the secret master key *MK* as input for the encryption algorithm. Thus, encryption to the group that one is not a member of and encryption for the "friends of friends" are not supported.

Revocation of a user for stateless BE schemes does not require re-keying for other users, thus for the stateless IBBE schemes it means re-encryption of data with a new symmetric key and consequent regeneration of *Headers* for the new set of receivers. Addition of a user in any IBBE scheme requires re-encryption of *Headers* for the new set of receivers in addition to creating a private key for that user.

Some broadcast encryption schemes (e.g. [19], [18]) achieve constant size ciphertext. In addition, the scheme described in [18] has the decryption complexity of $O(1)$, while the encryption cost is linear in the number of receivers.

VI. Comparison and Discussion

In Table I we summarize the evaluation of different encryption schemes according to the stated criteria and transmission cost. Transmission cost is defined as a number of decryption/encryption operations that have to be performed in order to transfer an encrypted object from the source to the destination. Of course, there is also a cost of sending messages. So, when for example in Diaspora's case all the keys and data are sent to recipients and in early PeerSoN's case they are stored locally, it is evident that the difference is very significant. However, the transmission cost depicted in the table does not take the cost of sending messages into account, and focuses solely on encryption/decryption.

The following notation is used in the table: d_{own} - own data of the profile owner, $d_{friends}$ - data that was received from friends as posts, a - the size of access structure in a CP-ABE scheme, n - the number of recipients, $O(1)_{symm}+O(n)_{asymm}$ - a symmetric operation with a constant cost followed by an asymmetric operation with a linear in the number of recipients n cost, $O(a)_{ABE}$ - ABE operation with a linear in the size of access formula a cost, s - number of shells in the Matryoshka, enc_{asymm} - one asymmetric encryption operation, dec_{asymm} - one asymmetric decryption operation, enc_{symm} - one symmetric encryption operation, k - a number of affected objects.

As we mentioned in Section IV, there is not enough information about the encryption system of Safebook. Any assumption about the encryption system that may be used in Safebook would lead to the same evaluation results as for other P2P systems that use the same encryption system and follow the Pull model.

We would like to note that in the table the encryption operation is performed for a group of receivers, and the decryption operation is performed by one receiver. So, although the encryption operation in Diaspora for one receiver requires only one symmetric and one asymmetric encryptions, for the group of receivers one needs one symmetric and n asymmetric encryption operations, thus the cost is linear.

From the table we can see that Diaspora (because of the Push model) has the highest storage cost storing not only own data, but also data received from others. The early PeerSoN with trivial BE has the worst storage cost for headers, though CP-ABE schemes are also not optimal from this point of view. The encryption cost of Persona with its underlying CP-ABE scheme is generally lower than for the rest systems, since the number of attributes is usually smaller than the number of receivers. At the same time, it has the worst decryption cost that depends on the number of attributes, while all the rest systems have a constant decryption cost. Moreover, CP-ABE decryption contains bilinear pairing operations, and since they are computationally expensive and their number linearly depends on the number of attributes, we can conclude that this operation is quite expensive.

The permissions modification cost is defined as a cost of changing permissions (set of receivers) in one object. Since all objects are pushed in Diaspora, it is impossible to modify a set of receivers of the already shared object, thus there is a dash in the table.

The cost of user addition to a group (a set of identities for identity-based schemes) is the highest for IBBE, and the cost of user removal from a group is the highest for CP-ABE (unless the scheme allows negations). Nevertheless, both of these schemes can encrypt for the conjunction / disjunction of groups. And while CP-ABE provides also the ability to encrypt for the group one is not a member of and for the friends of friends, the IBBE scheme is more secure and has the ability not to reveal access structures in encryption headers.

VII. Conclusions

We analyzed the scenario of P2P social networks without trusted parties and the impact this environment has on encryption-based access control systems. Based on this analysis we stated the following evaluation criteria that encompass efficiency, functionality, and privacy areas: efficiency of addition/removal of users from a group, efficiency of user key revocation, encryption/decryption efficiency, encryption header overhead, ability to encrypt for the conjunction/disjunction of groups, ability to encrypt for a group that one is not a member of, ability to encrypt for "friends of friends", ability not to reveal access structures in the header.

We analysed existing P2P architectures for social networks that focus on encryption as a means of ensuring data confidentiality. We evaluated the types of encryption systems that these architectures use (combination of asymmetric and symmetric cryptographies, CP-ABE) according to the stated criteria.

We also evaluated existing broadcast encryption (BE) schemes accordingly looking at the stated criteri and defined properties that are crucial for the BE schemes to be used in the P2P social network scenario. We found one BE scheme that meets all the requirements and adapted it to the social network scenario.

[1] some CP-ABE schemes are linear in the set of attributes from the user's key that satisfy the access structure

[2] for schemas with monotonic access structures

TABLE I
COMPARISON OF ENCRYPTION SYSTEMS OF P2P SOCIAL NETWORKS

	Safebook (unknown encryption system)	**Diaspora (trivial BE with Push model)**	**Persona (CP-ABE)**	**Early PeerSoN (trivial BE with Pull)**	**dynamic IBBE**
storage cost (data)	size(d_{own})	size(d_{own} + $d_{friends}$)	size(d)	size(d)	size(d)
storage cost (header)	unknown	$O(1)$	$O(a)$	$O(n)$	$O(1)$
encryption time	unknown	$O(1)_{symm} + O(n)_{asymm}$	$O(1)_{symm} + O(a)_{ABE}$	$O(1)_{symm} + O(n)_{asymm}$	$O(1)_{symm}+O(n)_{BE}$
decryption time	unknown	$O(1)$	$O(a)^1$	$O(1)$	$O(1)$
transmission cost	$2(s-1) \cdot enc_{asymm} + 2(s-1) \cdot dec_{asymm}$	0	0	0	0
permissions mod cost (add / remove)	unknown	$O(1)$ / —	$O(a)_{ABE}$ / $O(1)_{symm} + O(a)_{ABE}$	$O(1)$ / $O(1)_{symm} + O(n)_{asymm}$	$O(n)_{BE}$ / $O(1)_{symm}+O(n)_{BE}$
cost of user addition / removal to a group	unknown	$1 \cdot enc_{asymm}$ / $k \cdot enc_{symm} + n \cdot enc_{asymm}$	$1 \cdot keyCreate$ / $k \cdot enc_{symm} + k \cdot enc_{ABE} + (n \cdot keyCreate$ ^{2})	$1 \cdot enc_{asymm}$ / $k \cdot enc_{symm} + n \cdot enc_{asymm}$	$k \cdot enc_{BE}$ / $k \cdot enc_{symm} + k \cdot enc_{BE}$
ability to encrypt for the conjunction / disjunction of groups	unknown	✘/✓	✓/ ✓	✘/✓	✓/ ✓
ability to encrypt for a group that one is not a member of	unknown	✘	✓	✘	✘
ability to encrypt for "friends of friends"	unknown	✘	✓	✘	✘
ability not to reveal access structures in the encryption header	unknown	✓	✘	✘	✓

The combination of asymmetric and symmetric cryptography does not have sufficient efficiency and functionality for the P2P social network scenario, while CP-ABE schemes are inferior to BE schemes because none of the current CP-ABE schemes achieve non-monotonic, hidden access structures and low storage and computational cost at the same time. Therefore, we proposed to use broadcast encryption for the P2P social network scenario, since it does not have the mentioned drawbacks, even though it does not support the ability to encrypt for a group that one is not a member of and the ability to encrypt for "friends of friends". These issues are to be investigated in future work.

ACKNOWLEDGMENTS

This research has been funded by the Swedish Foundation for Strategic Research grant SSF FFL09-0086 and the Swedish Research Council grant VR 2009-3793.

REFERENCES

[1] Y. Afify, "Access control in a peer-to-peer social network," Master's thesis, EPFL, Lausanne, Switzerland, 2008.

[2] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "Peerson: P2p social networking: early experiences and insights," in *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, ser. SNS '09, 2009, pp. 46–52. [Online]. Available: http://doi.acm.org/10.1145/1578002.1578010

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

[4] ——, "On the security and feasibility of safebook: A distributed privacy-preserving online social network," in *Privacy and Identity Management for Life*, ser. IFIP Advances in Information and Communication Technology. Springer Boston, 2010, vol. 320, pp. 86–101.

[5] (2010, Apr.) A little more about the project. [Online]. Available: http://blog.joindiaspora.com/2010/04/21/a-little-more-about-the-project.html

[6] (2010, Nov.) Diaspora security architecture proposal. [Online]. Available: https://github.com/diaspora/diaspora/wiki/Security-Architecture-Proposal

[7] (2010, Oct.) Encryption. [Online]. Available: https://github.com/diaspora/diaspora/wiki/Encryption

[8] (2011, Aug.) Diaspora's federation protocol. [Online]. Available: https://github.com/diaspora/diaspora/wiki/Diaspora%27s-federation-protocol

[9] (2011, July) Diaspora roadmap. [Online]. Available: https://github.com/diaspora/diaspora/wiki/Roadmap

[10] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 135–146, August 2009. [Online]. Available: http://doi.acm.org/10.1145/1594977.1592585

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, ser. CCS '06. ACM, 2006, pp. 89–98.

[12] C. Delerable, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in *Pairing-Based Cryptography Pairing 2007*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, vol. 4575, pp. 39–59.

[13] V. Gupta, S. Gupta, S. Chang, and D. Stebila, "Performance analysis of elliptic curve cryptography for ssl," in *Proceedings of the 1st ACM workshop on Wireless security*, ser. WiSE '02. ACM, 2002, pp. 87–94.

[14] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, ser. SP '07. IEEE Computer Society, 2007, pp. 321–334. [Online]. Available: http://dx.doi.org/10.1109/SP.2007.11

[15] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography PKC 2011*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6571, pp. 53–70.

[16] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," in *Proceedings of the 5th International Conference on Information Security Practice and Experience*, ser. ISPEC '09. Springer-Verlag, 2009, pp. 1–12.

[17] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th*

ACM conference on Computer and communications security, ser. CCS '07, New York, NY, USA, 2007, pp. 195–203. [Online]. Available: http://doi.acm.org/10.1145/1315245.1315270

[18] H. Jiang, Q. Xu, and J. Shang, "An efficient dynamic identity-based broadcast encryption scheme," in *Data, Privacy and E-Commerce (ISDPE), 2010 Second International Symposium on*, 2010, pp. 27 –32.

[19] C. Delerable, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Advances in Cryptology ASIACRYPT 2007*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, vol. 4833, pp. 200–215.

[20] W. Zhang, Q. Xu, and P. He, "Identity-based broadcast encryption with recipient privacy," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol. 8, 2010, pp. 483 –487.