P2P Social Networks With Broadcast Encryption Protected Privacy

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

Existing centralized, provider-dependent networks do not provide users with mechanisms to fully protect their data. The provider has complete control of the service, and users have to rely on security mechanisms provided by the service. There is no guarantee that the trusted provider will enforce the privacy preferences of the users. Besides, there can be data mining, targeted advertisements, and even disclosure of users' data to third parties.

The aim of this work is to give full control of the data to the users in order to prevent any misuse by third parties. Protection provided by cryptographic mechanisms is more reliable than enforcement by laws or protection based on trust. Therefore, the PeerSoN project was started to design a peer-to-peer (P2P) provider-independent architecture with cryptographically protected privacy.

One of the main problems of P2P social network architectures is to achieve secure, efficient, 24/7 access control enforcement and data storage. None of the current P2P architectures for social networks manages to fully cope with this problem. They either rely on some form of trust, require each user to have his/her own constantly available server, or rely on computationally expensive cryptography such as a ciphertext-policy attribute-based encryption (CP-ABE). Current ABE schemes are very computationally intensive and produce ciphertexts that are linear in the number of attributes (too expensive for the P2P storage).

We utilize broadcast encryption schemes for an efficient encryption-based access control with high performance encryption and decryption regardless of the number of identities/groups. The storage is assumed to be a P2P untrusted storage with multiple replicas, therefore the encryption storage cost is crucial.

Access control in the proposed architecture is encryption based (everyone can download encrypted data). Broadcast encryption is used for data dissemination to groups and public key cryptography is used for user-to-user messaging. A multicast messaging (one message for several users) is realized via BE.

Broadcast encryption (BE) schemes are used to distribute encrypted data to a dynamic set of users in a cost-effective way. BE schemes with the following properties: stateless, fully collision resistant, with hidden set of receivers, dynamic, with constant size ciphertexts and keys, with computationally efficient decryption are suitable candidates for application to a social network scenario. We use a dynamic broadcast scheme that meets all these requirements. The utilized scheme is constructed as a Key Encapsulation Mechanism (KEM) which means that the encryption algorithm takes as input a set of receivers S and a group public key GPK 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 S. Data is stored in the form *(Header, encrypted data)*, and Header reveals no information about the set of receivers or any other parameters. Only a user whose ID is in the set can decrypt the *Header* using his/her private key. Users that are not members of the group can still encrypt to the group if they know GPK. Revocation and addition of users to a group are quite simple operations. Revocation of the group membership for stateless BE schemes does not require re-keying, only re-encryption of the data with a new symmetric key. Addition of a user to a group in dynamic schemes requires only re-encryption of *Headers* for the new set of receivers.

The properties of the encryption function make it more efficient to encrypt a message for one big group users than for several small. Therefore, all the contacts should be put into one BE group. The division of users into security groups is abstract: each abstract security group corresponds to some set of receivers S that can be easily modified during encryption. This flexibility allows us to create as many groups as required without effecting efficiency or manageability.

Both the encrypted data and the BE private keys for intended receivers are stored at the sender's storage. The receivers have to "pull" data from the sender's own storage when it is online or from its replicas when it is offline. The "pull" operation is preferred over the "push" operation since received but lost data (e.g. encryption key) can be *pulled* by the receiver again without requiring any action from the sender.

Although the system uses public-key cryptography to encrypt BE keys for intended recipients, there is no need for a trusted certificate authority (CA). A Web-of-Trust model with cumulative levels of trust in conjunction with a distributed hash table (DHT) for storing certificates can be used.

The composition of the public-key cryptography, broadcast encryption schemes, symmetric cryptography, and abstract security groups allows us to achieve a highly efficient and easily manageable architecture.