Privacy Preserving And Securing Using Proxy Re-signatures

Parvathy.K.S. (M.E.), PG Scholar, Department of Computer Science and Engineering, RVS Technical Campus, parvathyshibu@gmail.com.
Shuriya.B., M.E., (Ph.D)., Assistant Professor, Department of Computer Science and Engineering, RVS Technical Campus.

ABSTRACT - With the advancement in the cloud services, the data can not only be stored in the cloud, rather it is easy for the users to share or modify it in groups. Hence to preserve the integrity of data, users have to sign on all the blocks of data being shared in the group. For enforcing security once an user is revoked from the group, the existing method allows the signature of the revoked to be transformed into the signature of an existing user by utilizing the idea of proxy re-signatures. This mechanism assumes that there is no collusion. In this paper we propose a collusion resistant proxy re-signature scheme which can generate a re-signing key with an existing user’s private key and a revoked user’s public key.

Keywords: User revocation, proxy re-signatures, collusion resistant.

1 INTRODUCTION

With sharing and data services, people can easily work together in groups enabling sharing of data among others. Every user in the group is not only able to read the shared data contents but also they can modify it. Due to the hardware or software failures and human errors the integrity of the data may have to be compromised.

Various mechanisms [1][3][6], have been proposed in order to preserve the integrity of the shared data. In straightforward method the existing user has to download the particular part of shared data in case of user revocation (as shown in figure 1), which is highly inefficient and causes communication overheads.

In mechanisms such as in [2][4] a proxy re-signing method (as shown in figure 2) is used to overcome the limitations of the straightforward method. However it assumes that there is no collusion between the revoked user and the cloud. This method allows an existing user’s signature to be converted to a revoked user’s signature. Here the cloud acts as the semi trusted proxy, is only able to transform the signatures during the revocation but it cannot sign the blocks on behalf of the existing user or the revoked user. So before performing the transformation of signature, the system has to find that particular user from the ‘n’ number of users in the group to which the revoked user’s signature has to be changed. Also if any sort of collusion attacks occur the system fails to handle it.

Hence in this paper, we propose a new mechanism called collusion resistant proxy re-signature scheme to overcome the limitations mentioned above.

2 COLLUSION RESISTANT PROXY RE-SIGNATURE SCHEME

The traditional proxy re-signature method coupled with collusion resistance is designed to overcome the limitations such as

- Cannot handle collusion attacks.
- Selection of signatures of existing user, that is during revocation time the signature is converted. So only those users who currently has access rights can only re-sign it. Hence during revocation the list of eligible users has to be taken and then find one.

Before Revocation
After Revocation

**Figure 1: Straight Forward Method**

The figure shows the straightforward method where the signature of a revoked user “r” is not transferred to the signature of an existing user “e” during revocation process, rather the existing users download the data signed by the revoked user, verify it and then the existing users “e” re-sign it.

Before Revocation

After Revocation

**Figure 2: Proxy Re-Signature Scheme**

The figure shows the proxy re-signature scheme where the signature of a revoked user “r” is converted to the signature of an existing user “e”.

If a revoked user say r with the private key pk_r is able to collude with cloud which has a re-signing key ck_e_r = pk_e / pk_r, then the revoked user along with the cloud can easily reveal the private key pk_e of the existing user say e.

By generating re-signing key with an existing user’s private key and the revoked user’s public key the collusion resistant proxy re-signing scheme eliminates the collusion attack. The existing user e computes the re-signing key as ck_e_r = b^{k_e}. Then because of the hardness of computation the DL problem in G (G is the cyclic group) the private key of an existing user pk_e cannot be found out even if the revoked user with private key pk_r and the cloud collude together.

2.1 Collusion Attack

Mathematically, a collusion attack finds two different messages s_1 and s_2 such that hash(s_1) = hash(s_2). The attacker has no control over the content of the message, but they are arbitrarily chosen by the algorithm.

2.2 DL Problem

Discrete logarithm is an integer k solving the equation b^k = g, where b and g elements of a finite group. Discrete logarithm are thus the finite –group-theoretic analogue of ordinary logarithms which solve the same equation for real numbers b and g, where b is the base of logarithm and g is the value whose logarithm is being taken.

In general let G be any group with group operations denoted by multiplication. Let b and g be any elements of G. then any integer k that solves the equation of form b^k = g is termed as discrete logarithm of g to the base b.

Computing discrete logarithm is believed to be difficult. No efficient general method for computing discrete logarithms on conventional computers is known, hence we take the security assumption that DL problem has no efficient solution.

3 SYSTEM DESIGN

The system model is as shown in the figure 3. The group admin has the complete authority of adding members to the group. When a user shares a data, a signature is attached to it. Here we have used the RSA algorithm for signature generation. Once a file
is being shared all the other group members can access, download or modify it. Once the content is modified by another user, they have to attach their signature along with it.

When a user is revoked or removed, to preserve the integrity of the data the revoked user should no longer have access to the files shared in the group. Considering the chances of a revoked user collusion to access the private key of an existing user, a re-signing key with an existing user’s private key and the revoked user’s public key is used. It can be implemented using the Elliptic Curve Cryptosystem.

4 RESULTS

![Figure 4 Time taken to re-sign against the number of users revoked.](image)

4.1 Performance Evaluation

Comparison between the two methods Collusion resistant proxy re-signature (CRPR) schema and traditional proxy re-signature (PR) Schema are done. The performance is measured in terms of time by plotting graph against revocation time and the number of re-signed blocks. Here we assume that the computational ability of existing user and the cloud are the same. It is clear from the graph that our method CRPR schema can reduce the computation and communication overheads and the cloud is able to re-sign the blocks efficiently. The PR scheme takes 25 seconds to re-sign the 400 blocks whereas our mechanism takes much lesser time. It takes lesser seconds to re-sign blocks providing security over collusion attacks.

5 RELATED WORK

In 1998, Blaze, Bleumer, and Strauss (BBS) proposed proxy re-signatures, in which a semi trusted proxy acts as a translator between two users say Alice and Bob. To translate, the proxy converts a signature from Alice into a signature from Bob on the same message. The proxy, however, does not learn any signing key and cannot sign arbitrary messages on behalf of either Alice or Bob.

A proxy re-signature scheme is a tuple of (possibly probabilistic) polynomial time algorithms \((\text{KeyGen}, \text{ReKey}, \text{Sign}, \text{ReSign}, \text{Verify})\), where:\((\text{KeyGen}, \text{Sign}, \text{Verify})\) form the standard key generation, signing, and verification algorithms.

- On input \((\text{pk}_A, \text{sk}_A, \text{pk}_B, \text{sk}_B)\), the re-encryption key generation algorithm, ReKey, outputs a key \(r \text{pk}_A \rightarrow B\) for the proxy. (Note: \(r \text{pk}_A \rightarrow B\) allows to transform A’s signatures into B’s signatures thus B is the delegator.) The input marked with a ‘*’ is optional.

- On input \(r \text{pk}_A \rightarrow B\) a public key \(\text{pk}_A\), a signature \(\sigma\), and a message \(m\), the re-signature function, ReSign, outputs \(\text{ReB}(m)\) if Verify\((\text{pk}_B, m, \sigma)\) and \(\perp\) otherwise \([2]\).

Other than \([4]\) none of the papers discussed about the issue of user revocation assuming that there is no collusion occurs. Our mechanism adds on to its performance and security to handle the collusion attacks.

Allowing a public auditor to audit the entire data stored on the cloud without downloading the entire content could add to the efficiency of this work. A homomorphic authenticators \([4][5]\) also known as homomorphic verifiable tags is able to check the integrity of the data stored without downloading the entire data. Homomorphic authenticable proxy re-signatures with the property of blockless verifiability and non-malleability can provide efficient auditing mechanisms during the revocation process.

6 CONCLUSIONS

In this paper, we propose a collusion resistant proxy re-signature scheme for providing efficient user revocation by preserving the data integrity using proxy re-signatures with collusion resistance. So that even if the revoked user collude with the system, the private keys of the users in the group cannot be revealed. Thus making it more efficient than the traditional proxy re-signing methods in terms of re-signing of
blocks as well as computation and communication overheads.

Figure 3. System Design

REFERENCES


