Volume 4, Issue 11, November 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

IEURC: Integrity of Shared data with Efficient User Revocation in the Cloud

Supriya Gade*
Department of Computer Engineering
SCOE, Sudumbare, Pune, India

Prashant Kumbharkar
Department of Computer Engineering
SCOE, Sudumbare, Pune, India

Abstract— With popularization of cloud services, multiple users easily share and update their data through cloud storage .For data integrity and security in the cloud storage, users in the group need to compute signatures on all the blocks in shared data, and due to data modifications performed by different users , Different users are sign by different blocks .For security reasons, when a user is revoked from the group, the blocks which were previously signed by this revoked user must be re-signed by an existing user. The straightforward method, which allows an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient due to the large size of shared data in the cloud. In this paper, we propose the integrity of shared data with efficient user revocation in mind. By utilizing the idea of proxy re-signatures, we allow the cloud to re-sign blocks on behalf of existing users during user revocation, so that existing users do not need to download and re-sign blocks by themselves. In addition, a public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud.

Keywords— integrity, Efficiency, user revocation, re-sign.

I. INTRODUCTION

In recent years, the emerging cloud-computing paradigm is rapidly gaining momentum as an alternative to traditional Information technology. Cloud storage, an important service of cloud computing, allows users to move data from their local storage systems to the cloud and enjoy the on-demand high quality cloud services. For data storage and sharing services like Google Drive and Drop box provided by cloud, people can easily shared data and work together in group. Once a user creates shared data in cloud, every user in the group is able to not only access data but also modify shared data. Although cloud providers promise a more secure and reliable environment to the users, the integrity of data in the cloud may still be compromised, due to the existence of hardware/software failures and human errors [2].

To protect the integrity of data in the cloud, a signature is attached to each block in data, and the integrity of data relies on the correctness of all the signatures. One of the most significant and common features of these it allow a public verifier to efficiently check data integrity in the cloud without downloading the entire data, referred to as public auditing. This public verifier could be A) Client: who would like to utilize cloud data for particular purposes (e.g., search, computation, data mining, etc.)Or B) a third-party auditor (TPA) who is able to provide verification services on data integrity to users. For security reasons, when a user leaves the group or is behave, this user must be revoked from the group. As a result, this revoked user should no longer be able to access and modify shared data, and the signatures generated by this revoked user are no longer valid to the group. Therefore, although the content of shared data is not changed during user revocation, the blocks, which were previously signed by the revoked user, still need to be re-signed by an existing user in the group. As a result, the integrity of the entire data can still be verified with the public keys of existing users only. Since shared data is outsourced to the cloud and users no longer store it on local devices, a straightforward method to re-compute these signatures during user revocation (as shown in Fig. 1) is to ask an existing user (i.e., Alice) to first download the blocks previously signed by the revoked user (i.e., Bob), verify the correctness of these blocks, then re-sign these blocks, and finally upload the new signatures to the cloud. However, this straightforward method may cost the existing user a huge amount of communication and computation resources by downloading and verifying blocks, and by re-computing and uploading signatures, especially when the number of re-signed blocks is quite large or the membership of the group is frequently changing.

Clearly, if the cloud could possess each user's private key, it can easily finish the re-signing task for existing users without asking them to download and re-sign blocks. However, since the cloud is not in the same trusted domain with each user in the group, outsourcing every user's private key to the cloud would introduce significant security issues. Another important problem we need to consider is that the re-computation of any signature during user revocation should not affect the most attractive property of public auditing — auditing data integrity publicly without retrieving the entire data. Therefore, how to efficiently reduce the significant burden to existing users introduced by user revocation, and still allow a public verifier to check the integrity of shared data without downloading the entire data from the cloud, is a challenging task. In this paper, we propose by utilizing the idea of proxy re-signatures, once a user in the group is revoked, the cloud is able to resign the blocks, which were signed by the revoked user, with a re-signing key (as

presented in Fig. 2). As a result, the efficiency of user revocation can be significantly improved, and computation and communication resources of existing users can be easily saved. Meanwhile, the cloud, which is not in the same trusted domain with each user, is only able to convert a signature of the revoked user into a signature of an existing user on the same block, but it cannot sign arbitrary blocks on behalf of either the revoked user or an existing user.

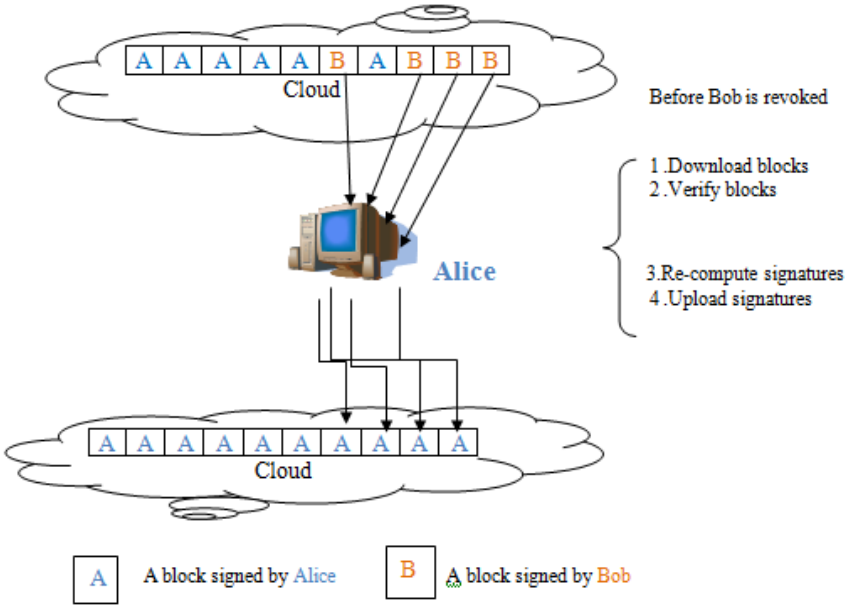

Fig. 1. Alice and Bob share data in the cloud. When Bob is revoked, Alice re-signs the blocks that were previously signed by Bob with her private key

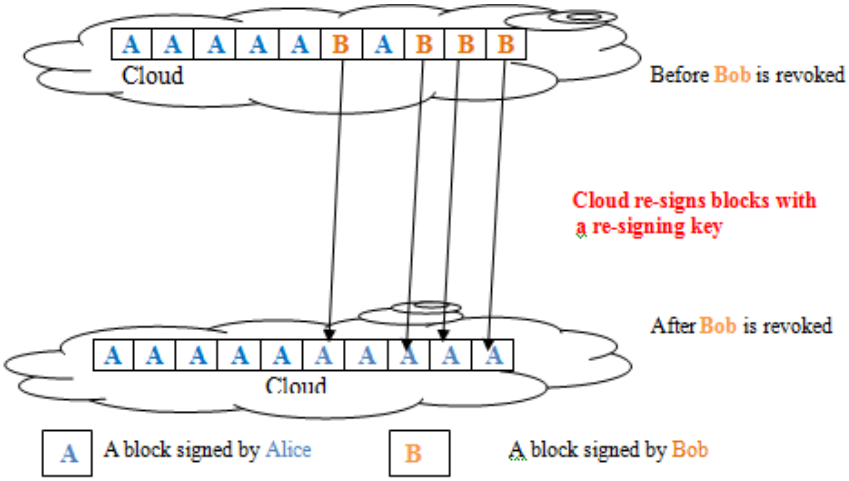

Fig. 2. When Bob is revoked, the cloud re-signs the blocks that were previously signed by Bob with a resigning key.

II. PRELIMINARIES

II-A. Bilinear Maps:

Let G1, G2 and G*T* be three multiplicative cyclic groups of prime order *p*, *g*1 and *g*2 be the generators of G1 and G2. ψ is a computable isomorphism from *G*2 to *G*1, with $\psi(g2) = g1$. The map $e : G1 \times G2 \rightarrow GT$ is said to be an admissible bilinear pairing if the following conditions hold true.

(1) *e* is bilinear, i.e. $e(g1^a, g2^b) = e(g1, g2)^{ab}$ for all $a, b \in Zp$.

(2) *e* is non-degenerate, i.e. $e(g1, g2) \neq 1G_T$

(3) *e* is efficiently computable.

II-B. Security Assumptions:

The security of our mechanism is based on the following security assumptions.

1. Computational Diffie-Hellman (CDH) Problem:

   Let $a, b \in z_p^*$, given $g, g^a, g^b \epsilon G_1$ as input, output $g^{ab} \epsilon G_1$.

2. Homomorphic Authenticators:

   Homomorphic authenticators, also called homomorphic verifiable tags, allow a public verifier to check the integrity of data stored in the cloud without downloading the entire data. They have been widely used as building blocks in the previous public auditing mechanisms.

3. Proxy Re-signatures:

   Proxy re-signatures, first proposed by Blaze *et al.*allow a semi-trusted proxy to act as a translator of signatures between two users, for example, Alice and Bob. More specifically, the proxy is able to convert a signature of Alice into a signature of Bob on the same block. Meanwhile, the proxy is not able to learn any private keys of the two users, which means it cannot sign any block on behalf of either Alice or Bob. In this paper, to improve the efficiency of user revocation, we propose to let the cloud to act as the proxy and convert signatures for users during user revocation.

4. Shamir Secret Sharing:

   An (s, t)-Shamir Secret Sharing scheme($s \geq 2t - 1$), first proposed by Shamir, is able to divide a secret π into s pieces in such a way that this secret π can be easily recovered from any t pieces, while the knowledge of any $t - 1$ pieces reveals absolutely no information about this secret π.The essential idea of an (s, t)-Shamir Secret Sharing scheme is that, a number of t points uniquely defines at−1 degree polynomial.

III. PROPOSED SCHEME

III-A. Review of the system model

As illustrated in Fig. 3, the system model in this paper includes three entities: the cloud, the public verifier, and users (who share data as a group). The cloud offers data storage and sharing services to the group. The public verifier, such as a client who would like to utilize cloud data for particular purposes (e.g., search, computation, data mining, etc.) or a third-party auditor (TPA) who can provide verification services on data integrity aims to check the integrity of shared data via a challenge-and response protocol with the cloud. In the group, there is one original user and a number of group users. The original user is the original owner of data. This original user creates and shares data with other users in the group through the cloud. Both the original user and group users are able to access, download and modify shared data. Shared data is divided into a number of blocks. A user in the group can modify a block in shared data by performing an insert, delete or update operation on the block.

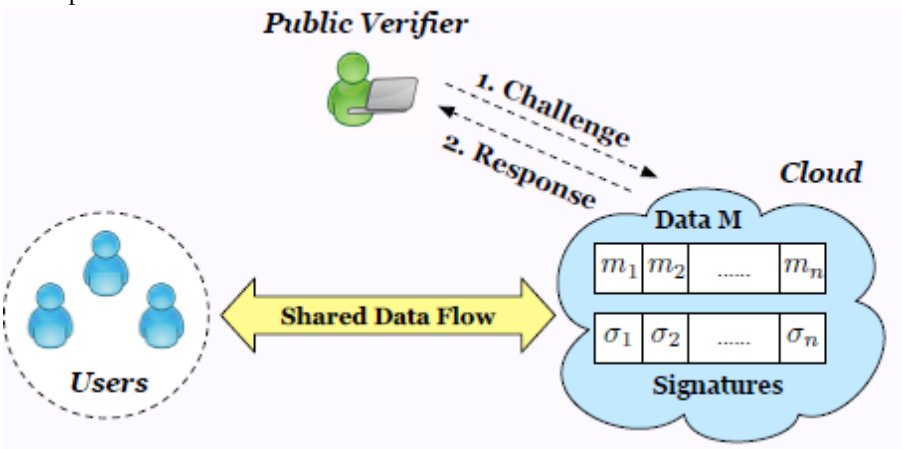

Fig. 3. The system model includes the cloud, the public verifier, and users

Once a user modifies a block, this user also needs to sign the modified block with his/her own private key. By sharing data among a group of users, different blocks may be signed by different users due to modifications from different users. When a user in the group leaves or misbehaves, the group needs to revoke this user. Generally, as the creator of shared data, the original user acts as the group manager and is able to revoke users on behalf of the group. Once a user is revoked, the signatures computed by this revoked user become invalid to the group, and the blocks that were previously signed by this revoked user should be re-signed by an existing user's private key, so that the correctness of the entire data can still be verified with the public keys of existing users only. By utilizing the idea of proxy re-signatures, once a user in the group is revoked, the cloud is able to resign the blocks, which were signed by the revoked user, with a re-signing key. As a result, the efficiency of user revocation can be significantly improved, and computation and communication resources of existing users can be easily saved. Meanwhile, the cloud, which is not in the same trusted domain with each user, is only able to convert a signature of the revoked user into a signature of an existing user on the same block, but it cannot sign arbitrary blocks on behalf of either the revoked user or an existing user.

III-B. Design Objectives

Our proposed mechanism should achieve the following properties:

(1) Correctness: The public verifier is able to correctly check the integrity of shared data.

(2)Efficient and Secure User Revocation: On one hand, once a user is revoked from the group, the blocks signed by the revoked user can be efficiently re-signed. On the other hand, only existing users in the group can generate valid signatures on shared data, and the revoked user can no longer compute valid signatures on shared data.

(3) Public Auditing: The public verifier can audit the integrity of shared data without retrieving the entire data from the cloud, even if some blocks in shared data have been re-signed by the cloud.

(4) Scalability: Cloud data can be efficiently shared among a large number of users, and the public verifier is able to handle a large number of auditing tasks simultaneously and efficiently.

IV. CONCLUSIONS

In this paper, we proposed a new mechanism for shared data with efficient user revocation in the cloud. When a user in the group is revoked, we allow the semi-trusted cloud to re-sign blocks that were signed by the revoked user with proxy re-signatures. These New proxy re-signatures of cloud can improve the efficiency and integrity of user revocation, and existing users in the group can save a significant amount of computation and communication resources during user revocation.

REFERENCES

[1] B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revoation in the Cloud," IEEE Transaction on service computing 2014.

[2] B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revoation in the Cloud," in the Proceedings of IEEE INFOCOM 2013, 2013, pp. 2904–2912

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz,A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, Apirl 2010.

[4] B. Wang, S. S. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," in Proceedings of IEEE ICDCS 2013, 2013.

[5] B. Wang, H. Li, and M. Li, "Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics," in the Proceedings of IEEE ICC 2013, 2013.

