International Journal of Computer Network and Security(IJCNS)
Vol 7. No.1 2015 Pp. 1-8

available at : www.ijcns.com
ISSN: 0975-8283

SECURE AND EFFICIENT PRIVACY-PRESERVING PUBLIC AUDITING SCHEME FOR CLOUD STORAGE

Dr.G.K.Kamalam[1] B.Neka[2] E.Jamunadevi[3]

[1]Assistant Professor(SLG),Department of IT,Kongu Engineering College,Perundurai,Erode,Tamilnadu,India

[2,3]IV BTech IT, Kongu Engineering College, Perundurai,Erode,Tamilnadu,India

ABSTRACT

With cloud storage services, it is commonplace for data to be not only stored in the cloud, but also shared across multiple users. However, public auditing for such shared data while preserving identity privacy remains to be an open challenge.Here secure and effective methods are needed to secure integrity and privacy data stored in cloud.This paper provides a privacy preserving public auditing scheme that supports public auditing and identity privacy on shared data stored in the cloud storage service for enhancing its security and efficiency .This paper has mainly concentrated on improving the security mechanism of ownCloud storage service.

Keywords: Secure and Efficient, Privacy-Preserving, Public Auditing Scheme,Cloud Storage

I. INTRODUCTION

Cloud service providers manage an enterprise-class infrastructure that offers a scalable, secure and reliable environment for users, at a much lower marginal cost due to the sharing nature of resources[1,8]. It is routine for users to use cloud storage services to share data with others in a team, as data sharing becomes a standard feature in most cloud storage offerings, including **Dropbox, Google Docs and ownCloud.** The integrity of data in cloud storage, however, is subject to skepticism and scrutiny, as data stored in an untrusted cloud can easily be lost or corrupted, due to hardware failures and human errors[2]. To protect the integrity of cloud data, it is best to perform public auditing by introducing a third party auditor (TPA), who offers its auditing service with more powerful computation and communication abilities than

regular users.

II. PROBLEM STATEMENT

A unique problem introduced during the process of public auditing for shared data in the cloud is how to preserve **identity privacy** from the TPA, because the identities of signers on shared data may indicate that a particular user in the group or a special block in shared data is a higher valuable target than others. However, no existing mechanism in the literature is able to perform public auditing on shared data in the cloud while still preserving identity privacy. In this paper, we propose a new privacy preserving public auditing mechanism for shared data in an untrusted cloud [2,3]. Here, we utilize ring signature so that the third party auditor is able to verify the integrity of shared data for a group of users without retrieving the entire data — while the identity of the signer on each block in shared data is kept private from the TPA[4,5].

III. PROPOSED SYSTEM MODEL

This paper involves three parties: the cloud server, the third party auditor (TPA) and users is shown in Figure 1. There are two types of users in a group: the original user and a number of group users. The original user and group users are both members of the group. Group members are allowed to access and modify shared data created by the original user based on access control polices. Shared data and its verification information (i.e. Mac code) are both stored in the cloud server. The third party auditor is able to verify the integrity of shared data in the cloud server on behalf of group members.

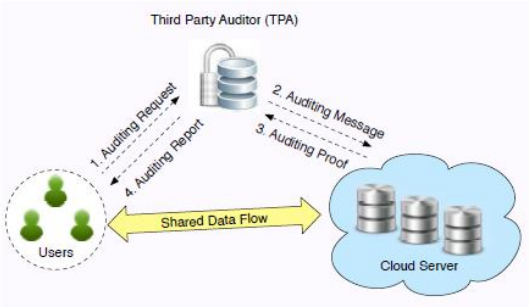


Fig 1: System model includes User, Cloud Server and TPA

In this paper, we only consider how to audit the integrity of shared data in the cloud with **static** groups. It means the group is pre-defined before shared data is created in the cloud and the membership of users in the group is not changed during data sharing. The original user is responsible for deciding who is able to share her data before outsourcing data to the cloud.

When a user (either the original user or a group user) wishes to check the integrity of shared data, she first sends an auditing request to the TPA. After receiving the auditing request, the TPA generates an auditing message to the cloud server, and retrieves an auditing proof of shared data from the cloud server. Then the TPA verifies the correctness of the auditing proof. Finally, the TPA sends an auditing report to the user based on the result of the verification.

A. Threat Model

Integrity Threats

Two kinds of threats related to the integrity of shared data are possible. First, an adversary may try to corrupt the integrity of shared data and prevent users from using data correctly. Second, the cloud service provider may

inadvertently corrupt (or even remove) data in its storage due to hardware failures and human errors[5,6].

Privacy Threats

The identity of the signer on each block in shared data is private and confidential to the group. During the process of auditing, a **semi-trusted** TPA, who is only responsible for auditing the integrity of shared data, may try to reveal the identity of the signer on each block in shared data based on verification information. Once the TPA reveals the identity of the signer on each block, it can easily distinguish a high-value target (a particular user in the group or a special block in shared data).

B. Design Objectives

To enable the TPA efficiently and securely verify shared data for a group of users, its required to achieve following properties: (1) **Public Auditing**: The third party auditor is able to publicly verify the integrity of shared data for a group of users without retrieving the entire data. (2) **Correctness**: The third party auditor is able to correctly detect whether there is any corrupted block in shared data. (3) **Unforgeability**: Only a user in the group can generate valid verification information on shared data. (4) **Identity Privacy**: During auditing, the TPA cannot distinguish the identity of the signer on each block in shared data.

IV. PRELIMINARIES

C. Ring Signatures

The concept of ring signatures is that with ring signatures, a verifier is convinced that a signature is computed using one of group members' private keys, but the verifier is not able to determine which one. This property can be used to preserve the identity of the signer from a verifier [4].

D. Homomorphic Authenticable Ring Signature

The ring signatures generated by HARS is able not only to preserve identity privacy but also to support blockless verification [9,10,11]. HARS contains three algorithms: **KeyGen**, **RingSign** and **RingVerify**. In **KeyGen**, each user in the group generates her public key and private key. In **RingSign**, a user in the group is able to sign a block with her private key and all the group members' public keys. A verifier is allowed to check whether a given block is signed by a group member in **RingVerify**.

V. PRIVACY PRESERVING PUBLIC AUDITING FOR SHARED DATA IN CLOUD

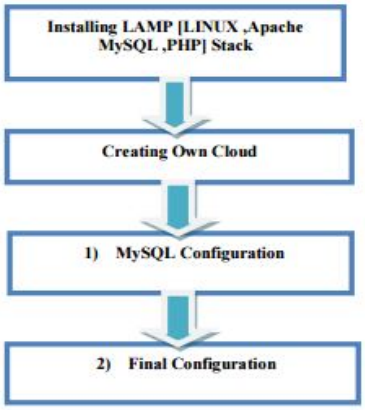
Using HARS and its properties, we now construct privacypreserving public auditing mechanism for shared data in the cloud. Here, the TPA can verify the integrity of shared data for a group of users.Meanwhile, the identity of the signer on each block in shared data is kept private from the TPA during the auditing [7].
It includes five algorithms: **KeyGen**,**SigGen**, **Modify**, **ProofGen** and **ProofVerify**.

E. ownCloud

ownCloud is a software system for what is commonly termed "file hosting". As such,

ownCloud is very similar to the widely used Dropbox, with the primary difference being that ownCloud is free and open-source, and thereby allowing anyone to install and operate it without charge on a private server, with no limits on storage space or the number of connected clients.

Steps in developing ownCloud



Overall operation of Auditing

F. Generation of Key and Signature for user file

In **Key-Gen**, users generate their own public/private key pairs.In **SigGen**, a user (either the original user or a groupuser) is able to compute ring signatures on blocks in shared data. Each user in the group is able to perform an insert, delete or update operation on a block, and compute the new ring signature on this new block in **Modify**.

Generating a Key:

For Key Generation **MD5 operation** is employed. Below equation shows a single MD5 operation.

a = b + ((a + Process P (b, c, d) + M[i] + t[k]) <<< s)

Here :-a, b, c, d = are Chaining variablesProcess P=A non linear operationM[i] =For M[q x 16 + i], which is the **ith** 32-bit word in the **qth** 512-bit block of themessaget[k]=a constant<<<s =circular-left shift by s bits

Generating a ring signature:

Given the message m to be signed, a sequence of public keys P1, P2, . . . , Pr of all the ring members (each public key Pi specifies a trapdoor permutation gi), and a secret key Ss (which specifies the trapdoor information needed to compute g −1 s), the signer computes a ring signature as follows.

1. **Determine the symmetric key:** The signer first computes the symmetric key k as the hash of the message m to be signed: k = h(m) (a more complicated variant computes k as h(m, P1, . . . , Pr); however, the simpler construction is also secure.
2.
3. **Pick a random glue value:** Second, the signer picks an initialization (or "glue") value v uniformly at random from {0, 1} b .
4.
5. **Pick random xi's:** Third, the signer picks random xi for all the other ring members 1 ≤ i ≤ r, where i 6= s, uniformly and independently from {0, 1} b , and computes yi = gi(xi) .

6. **Solve for ys:** Fourth, the signer solves the following ring equation for ys: Ck,v(y1, y2, . . . , yr) = v . By assumption, given arbitrary values for the other inputs, there is a unique value for ys satisfying the equation, which can be computed efficiently.

7. **Invert the signer's trapdoor permutation:** Fifth, the signer uses his knowledge of his trapdoor in order to invert gs on ys, to obtain xs: xs = g −1 s (ys) . 6. Output the ring signature: The signature on the message m is defined to be the (2r + 1)-tuple: (P1, P2, . . . , Pr; v; x1, x2, . . . , xr) .

Encryption:
Files are encrypted using AES algorithm and uploaded to the cloud and TPA

G. Third Party Auditing and Implementation of proof Verify algorithm
The third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance.
ProofGen: To audit the integrity of shared data, a user first sends an auditing request to the TPA. After receiving an auditing request, the TPA generates an auditing message using encrypted signatue and sends an auditing message to the cloud server.
After receiving an auditing message , the cloud server generates a proof of possession of selected file with the public aggregate key **pak**.(mac code) and display it to the TPA
H. ProofVerification and Performance analysis
ProofVerify: TheTPA then generates the mac code for the selected file . If the code matches with the pak then the integrity is maintained or else file is corrupted
Computation Cost:
During auditing, the TPA first generates some random values to construct the auditing message, which only introduces a small cost in computation. Then, after receiving the

auditing message, the cloud server needs to compute a proof .

Communication Cost

The communication cost of Oruta is mainly introduced by two factors: the auditing message and the auditing proof. For each auditing message , the communication cost is c(|q| + |n|) bits, where |q| is the length of an element of Zq and |n| is the length of an index.

J. Performance of Auditing

Based on our proceeding analysis, the auditing performance under different detection probabilities is illustrated in Figure 2–5. As shown in Figure 2, the auditing time is linearly increasing with the size of the group. When c = 300, if there are two users sharing data in the cloud, the auditing time is only about 0.5 seconds. When the number of group member increases to 20, it takes about 2.5 seconds to finish the same auditing task. The communication cost of an auditing task under different parameters is presented in Figure 4 and 5. Compare to the size of entire shared data, the communication cost that the TPA consumes in an auditing task is very small. TPA needs to consume more computation and communication overhead to finish the auditing task. Specifically, when c = 300, it takes the TPA 1.32 seconds to audit the correctness of shared data, where the size of shared data is 2 GB; when c = 460, the TPA needs 1.94 seconds to verify the integrity of the same shared data.

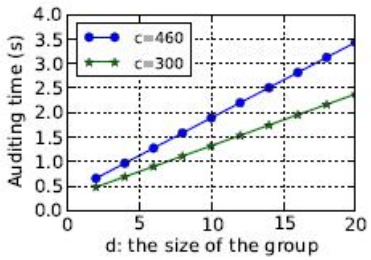


Fig 2: Impact of d on auditing time where k=100

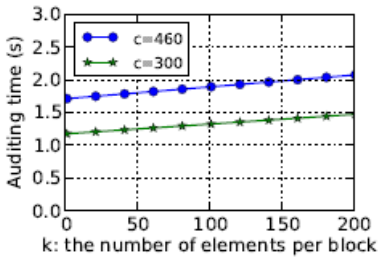


Fig 3: Impact of k on auditing time where d=10

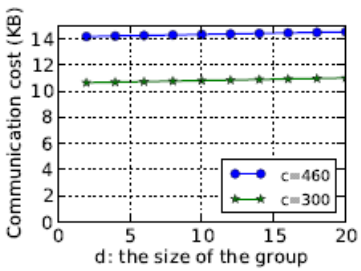


Fig 4: Impact of d- communication cost where k=100

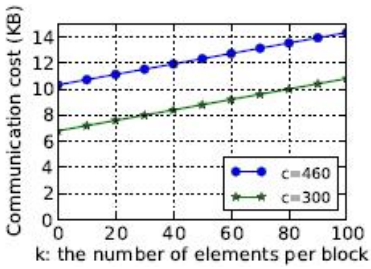


Fig 5: Impact of k- communication cost where d=10

VI. CONCLUSION

In this paper, we propose the privacy preserving public auditing mechanism for shared data in the cloud. We utilize ring signatures to construct homomorphic authenticators, so the TPA is able to audit the integrity of shared data, yet cannot distinguish who is the signer on each block, which can achieve identity privacy. An interesting problem in future work is how to efficiently audit the integrity of shared data with dynamic groups while still preserving the identity of the signer on each block from the third party auditor.

VI. REFERENCES

[1].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.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson,and D. Song, "Provable Data Possession at Untrusted Stores,"in Proc. ACM Conference on Computer and Communications Security(CCS), 2007, pp. 598–610.

[3] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-PreservingPublic Auditing for Data Storage Security in Cloud Computing,"in Proc. IEEE International Conference on Computer Communications(INFOCOM), 2010, pp. 525–533.

[4] R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret,"in Proc. International Conference on the Theory and Application ofCryptology and Information Security (ASIACRYPT). Springer-Verlag, 2001, pp. 552–565.

[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate andVerifiably Encrypted Signatures from Bilinear Maps," in Proc. In-ternational Conference on the Theory and Applications of CryptographicTechniques (EUROCRYPT). Springer-Verlag, 2003, pp. 416–432.

[6] H. Shacham and B. Waters, "Compact Proofs of Retrievability,"in Proc. International Conference on the Theory and Application ofCryptology and Information Security (ASIACRYPT). Springer-Verlag, 2008, pp. 90–107.

[7] Y. Zhu, H.Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S.Yau, "DynamicAudit Services for Integrity Verification of Outsourced Storage inClouds," in Proc. ACM Symposium on Applied Computing (SAC),2011, pp. 1550–1557.

[8] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable,and Fine-grained Data Access Control in Cloud Computing," inProc. IEEE International Conference on Computer Communications(INFOCOM), 2010, pp. 534–542.

[9] D. Boneh, B. Lynn, and H. Shacham, "Short Signature from theWeil Pairing," in Proc. International Conference on the Theory andApplication of Cryptology and Information Security (ASIACRYPT).Springer-Verlag, 2001, pp. 514–532.

[10] D. Boneh and D. M. Freeman, "Homomorphic Signatures forPolynomial Functions," in Proc. International Conference on the

[11]Theory and Applications of Cryptographic Techniques (EUROCRYPT).Springer-Verlag, 2011, pp. 149–168.