INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Enhanced Oruta Mechanism for Verifying Shared Data Integrity with Data Freshness and Traceability over the Cloud Data

N. Deivanayaki
Associate Professor, Department of Computer Science and Engineering, PET Engineering College, Vallioor, India
J. Amu Bebina
M.E. Student, Department of Computer Science and Engineering, PET Engineering College, Vallioor, India

Abstract:
Cloud computing is an on-demand computing which enables the user to store and share the data over the internet. While sharing, the cloud data is subject to scrutiny due to the modification introduced by the several users. In order to verify the data integrity several public auditing mechanisms is proposed to examine and analyze the system. But these mechanisms are failing to preserve the identity privacy which points out the private and sensitive information to the public verifiers and also we cannot assure that the cloud possesses the latest version of the cloud data. In this paper, to solve the above privacy and data freshness issues an enhanced ring signature mechanism is proposed by accomplishing the traceability over the cloud data. This paper results demonstrate the cloud data storage with more trustworthiness and more privacy and more security.

Keywords: *Identity privacy, public auditing, Data freshness, Traceability*

1. Introduction

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". One of these risks that can attack the cloud computing is the integrity of the data stored in the cloud [1]. When data is stored in clouds, the owner no longer has physical possession of the data's storage. But entrusted service providers have independent administrator authority over the data, and this creates potential security threats. Data integrity is also a security challenge in cloud computing.

The integrity of data is subject to doubt due to human errors and hardware or software failures. Therefore, the integrity of cloud data should be verified without any data utilization and without downloading the entire cloud. Traditionally, the data integrity is verified by retrieving the entire data from the cloud and then the correctness of signature is checked. However the efficiency of using this method on cloud data is in doubt [3].

The main reason is that normally the size of cloud data is large. Downloading the entire cloud data to verify data integrity will cost or even waste user's amounts of computation and communication resources. Besides many uses of cloud data do not necessarily need users to download the entire cloud data to local devices. It is because cloud providers, such as Amazon, can offer users computation services directly on large-scale data that already existed in the cloud.

Recently, many mechanisms [3], [4] have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing [2]. In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking [3].

Failing to preserve identity privacy on shared data during public auditing will reveal significant confidential information to public verifiers. To solve the above privacy issue on shared data, a novel privacy-preserving public auditing mechanism has been proposed. Here Ring signature [9] is exploited to construct homomorphic authenticators, so that a public verifier is able to verify the integrity of shared data without retrieving the entire data, while the identity of the signer on each block in shared data is kept private from the public verifier.

In this paper, to improve Data Privacy on shared data in cloud, we propose Traceability oruta (One ring signature to Rule Them All) mechanism to achieve traceability. The data freshness (the cloud possesses the latest version of shared data) is also proved while still preserving identity privacy. Achieving data freshness ensures that the retrieved data always reflects the most recent updates and prevents rollback attacks. Achieving data freshness is essential to protect against mis-configuration errors.

In this paper we assume that the data integrity in the cloud is well verified using the TPA and digital signature technique and we expect the system to reach a fine grade of data validity and quality.

2. Problem Statement

The system model in this paper involves cloud owner, Data owner, cloud user or group of users. There are two types of users in a group: the original user and a number of group users. Both the original user and group users are members of the group. Every member of the group is allowed to access and modify shared data. Shared data and its verification metadata (i.e., signatures) are both stored in the cloud server. A public verifier audits the shared data and it checks the integrity of shared data by the ring signature.

As illustrated in fig. 1, when the Data owner stores the data in the cloud, the cloud owner gives the public key and the private key to the Data owner. The original user in the group will form a ring signature using his private key and the other group members' public key. Due to this, a public verifier or TPA is convinced that a signature is computed using one of group members' private keys, but the verifier is not able to find which one. Due to this, the identity of the signer is preserved from a verifier during auditing. The TPA proves the data integrity by verifying the possession (the ownership of the data) of the auditing proof. Now there comes a problem when the private key is leaked other than the data owner. This leakage should be tracked.

Our mechanism should be designed to achieve the following properties: (1) Public Auditing: A public verifier is able to publicly verify the integrity of shared data without retrieving or downloading the entire data from the cloud. (2) Correctness: A public verifier is able to correctly verify shared data integrity. (3) Traceability: Tracking the fake user from accessing the data from the cloud. (4) Identity Privacy: A public verifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing. (5) Data freshness: Data freshness is essential to protect against mis-configuration errors.

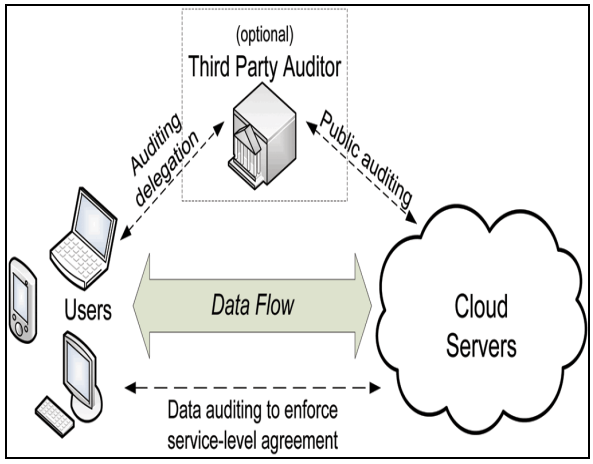

Figure 1: The system model includes the cloud owner, data owner and cloud users.

3. Architecture Design

In this paper, we are going to propose traceability Oruta to achieve traceability (tracking the fake user, tracking the data history).Data freshness is also proved by using Traceability Oruta.

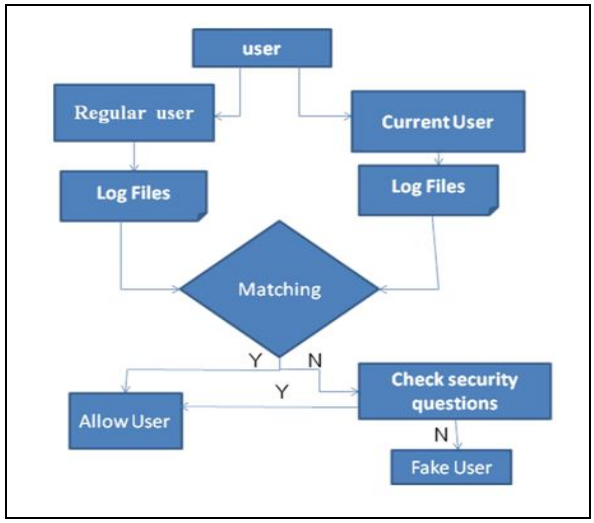

Figure 2: Tracking the fake user

The above figure explains about tracking the fake user. The verifier saves all the attributes and details of the regular user in the log files. The verifier maintains the log files. When the user login; the verifier checks the log files with the existing log files. If the matching is yes, it allow the user and if the matching is no, it checks by asking some security questions. If the answer is correct, then it will allow the user and if the answer is wrong, it is considered as fake user and it blocks that user from accessing the data from the cloud.

User1	Access1	Time	Ownership1	Public key1	Version Counter
User2	Access2	Time	Ownership2	Public key2	1….n
User3	Access3	Time	Ownership3	Public key3	1….n
UserN	AccessN	Time	Ownership4	Public key4	1….n

Figure 3: Tracking the data history

The Fig. 3 shows that the traceability of the data stored in the cloud. These data are used to denote which user accesses the data in which time and also specifies that the ownership of the data. It also maintains a version counter which denotes the latest updation of data stored in the cloud. So that it is used to retrieve the latest version of the cloud data. For each update the version counter is incremented by 1.

3.1. Ring Signatures

The ring signature is the type of digital signature which can be performed by any group member of users that each have keys. Therefore, a message signed with a ring signature is endorsed by someone in a particular group of people. The best characteristic of a ring signature is that it should be difficult to identify which of the group members' keys was used to produce the signature. In this, the signature is computed using one of the group member's private key, 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. The ring signatures concept is used to hide the identity of singer on each block so that the private and sensitive information of group does not seen by the TPA. To reduce time and long verification, we are extending the ring concept by homomorphic authenticable ring signatures. This homomorphic authenticable ring signature not only maintains the identity but also reduce long verification with supporting to blockless verification.

The name of the ring signature comes from the ring like structure of the signature algorithm because the ring is formed by the private key of the data owner and the public key of the all the users or the group key.

The privacy-preserving public auditing using signatures consist of three algorithms as mentioned here: KeyGen, RingSign, and RingVerify. In KeyGen algorithm each user in the group generates their public key and private key. In RingSign algorithm user in the group is related to sign a block with her private key and all group members' public keys. In Ring verify algorithm the verifier is used to check whether the given block is signed by the group member. The ring signatures for public auditing consist of following steps for auditing:

1. Each user generates its public and private key.
2. A user in the group signs a block with her private key and all group members' public key. Pk1 is public key of the user; Sk1 is private key of the user; (Pk1 ……Pkd) is ,d number of users of data block m Є ZP
3. User randomly selects data block m Let id is identifier of data block m
4. User ui encrypts with all user's public key, so only private key of the group user's i Є [1, d] would be able to decrypt it. This ensures privacy of data.
5. To ensure auditing by third-party user (ui), where i Є [1, d] signs the data block using his private key.
6. TPA (Third-party auditor) using a Pk1…Pkd Where d is number of users in the group.

TPA calculates signature of data blocks but unaware of who sign it .Therefore calculates signature using each given public key (Pk1 ……Pkd) from this set. Gsign = signature set for (Pk1 ……Pkd) If Gsign = {sign1, sign2 …signd} matches with original sign then data block is intact. By using this scheme user can also do the data dynamic operation. As there is group of users which share their data to each other, they can do modification on data of CS.

3.2. Batch Auditing

In batch auditing multiple users can access CS (Cloud Server) simultaneously. The TPA may concurrently handle multiple auditing processes for multiple users. Multiple TPA is used for the auditing process. TPA batches all users task and audits it at one to time. The advantage of batch auditing is that it reduces the time for handling the multiple audits for multiple users. In Fig 4. the graph shows the comparison of individual auditing and the batch auditing. The comparison is done on the basis of auditing time required to perform number of tasks.

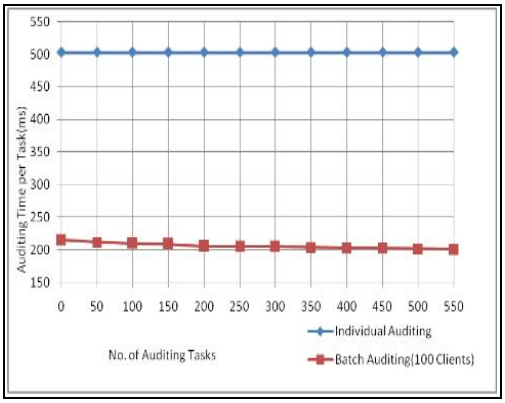

Figure 4: The comparison chart

3.3. Supports for Data Dynamics

In cloud computing, users can update their data continuously for various application purposes. So here privacy preserving public auditing supports for data dynamics in which user can do modifications on stored data. This data dynamics supports for update, delete, insert operations. The user can update the data by getting the permission from the data owner. If the data owner grants the permission to the data user then the user can do the modifications. Also that the Oruta mechanism not only supports the dynamic operations, it also supports the dynamic groups. In earlier mechanism the group is predefined (static group) before outsourcing the data. In oruta, a new user can be added and revoked from a group at anytime. If a new user can be added means a new ring signature is formed by adding his/her public key. Similarly if a user is revoked from the group means the recomputation is performed on the ring signature scheme.

3.4. Homomorphic Authenticators

Homomorphic authenticators are basic tools to construct public auditing mechanisms [1]. Besides unforgeability (i.e., only a user with a private key can generate valid signatures), should satisfy the following properties:

1. Blockless Verifiability
2. Non-malleabilty

Blockless verifiability allows a verifier to audit the correctness of data stored in the cloud server with a special block, which is a linear combination of all the blocks in data. If the integrity of the combined block is correct, then the verifier believes that the integrity of the entire data is correct.

Non-malleability denotes the members of the group only denote valid signatures which denote the unforgeability of the group members.

3.5. Threat Model

3.5.1. 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. Second, the cloud service provider may inadvertently corrupt (or even remove) data in its storage due to hardware failures and human errors.

3.5.2. 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 public verifier, who is only allowed to verify the correctness of shared data integrity, may try to reveal the identity of the signer on each block in shared data based on verification metadata.

4. Related Work

Provable data possession (PDP) [11], allows a verifier to check the correctness of a data stored at an untrusted server. By utilizing RSA-based homomorphic authenticators and sampling strategies, the verifier is able to publicly audit the integrity of data without retrieving the entire data, which is referred to as public auditing. Unfortunately, this mechanism is only suitable for auditing the integrity of personal data.

Proofs of Retrievability (POR), which is also able to check the correctness of data on an untrusted server. The public mechanism proposed by Wang et al. [4] and [12] are able to preserve users' confidential data from a public verifier by using random masking. Compared to previous works [10], [13], [7], this mechanism is able to improve data privacy by using traceability and the data freshness is also proved.

Shacham and Waters gave a new model for POR enabling verifiability of unlimited number of queries by user with reduced overhead. . Later Bowels and Juels gave a theoretical model for the operation of POR, but all these mechanisms proposed were weak from the

safety point because they all work for single server. Therefore Bowels in their further work gave a HAIL protocol extending the POR mechanism for multiple servers.
Priya Metri and Geeta Sarote [13] proposed a risk model to overcome the threat of integrity and provide data privacy in the cloud storage. It uses TPA (Third Party Auditor) and digital signature mechanism for the purpose of reliable data retrievable. The TPA being used notifies any illegal access attempting to make changes, avoiding the changes in data and maintaining the originality of data.

5. Conclusion

In this paper, we propose an enhanced Privacy Preserving with Data Freshness by accomplishing traceability over oruta. A new mechanism is adopted to achieve traceability called Traceability Oruta. Due to this, Data Privacy in cloud is improved. We utilize ring signatures, so that a public verifier is able to audit shared data integrity without retrieving the entire data. Data freshness is also proved. Freshness verification should be extremely efficient for existing file system operations and induce minimal latency. To ensure freshness, it is necessary to authenticate not just data blocks, but also their *versions.*

6. Acknowledgment

I would like to thank my guide Prof.N.Deivanayaki for assisting me in this paper work.

7. References

1. B. Wang, B. Li, and H. Li, "Oruta: Privacy- Preserving Public Auditing for Shared Data in The Proc. IEEE Fifth Int'l Conf. Cloud Computing, pp. 295- 302,
2. C. Wang, Q. Wang, K. Ren, and W, Lou,"Privacy- Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE INFOCOM, pp. 525-533, 2010.
3. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-610, 2007.
4. H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security:
5. C. Wang, S.S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013
6. B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revocation in the Cloud," Proc. IEEE INFOCOM, pp. 2904-2912, 2013
7. B. Wang, B. Li, and H. Li, "Panda: Public for Shared Data with Efficient User Revocation in the Cloud," IEEE Trans. Services Computing, 20 Dec. 2013, DOI:10.1109/TSC.2013.2295611
8. B. Wang, B. Li, and H. Li, "Knox: Privacy- Preserving Auditing for Shared Data with Large Groups in the Cloud," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS'12),pp. 507-525, June 2012.
9. R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01), pp. 552-565, 2001.
10. A. Juels and B.S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 584-597, 2007.
11. K.D.Bowers,A.Juels and A.oprea,"Proofs of Retrievability:Theory and Implementation,"Cryptology ePrint Archive,Report 2008/175,2008.
12. G.Ateniese,R.Burns,R.Curtmola,J.Herring,L.Kissner,Z.Peterson and D.Song, "Proovable data Possession at Untrusted Stores," Proceedings.Of CCS "07, pp. 598–609, 2007.
13. JiaXu and Ee-Chien Chang, "Towards efficient proofs of retrievibility in cloud storage".
14. G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession, "In Proceedings of SecureComm "08,pp.1-10,2008.
15. Q.Wang,C.Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiabilty and Data Dynamic for Storage Security in Cloud Computing,".