

ORUTA: Secured Shared Data In Cloud Using Ring Signature

¹K.Muthulakshmi, ²S.Ramakrishnan

¹PG Student, Department of Computer Applications, Sudharsan Engineering College, Pudukkottai, India.

²Assistant Professor, Department of Computer Applications, Sudharsan Engineering College, Pudukkottai, India.

¹kmuthulakshmi23@gmail.com, ²srk.accet@gmail.com

Abstract— With cloud information services, it is very common for information to be not only saved in the cloud, but also distributed across multiple users. Unfortunately, the reliability of cloud information is subject to uncertainty due to the lifestyle of hardware/software problems and human errors. Several systems have been designed to allow both information owners and public verifiers to effectively auditing cloud information integrity without accessing the whole information from the cloud server. However, public auditing on the reliability of distributed information with these existing mechanisms will certainly reveal private information—identity privacy—to public verifiers. This survey recommends a novel privacy-preserving procedure that facilitates public auditing on distributed information saved in the cloud. It manipulates ring signatures to estimate confirmation meta-data needed to audit the correctness of distributed information. With this procedure, the identification of the signer on each block in distributed information is kept private from public verifiers, who are able to effectively confirm distributed information reliability without retrieving the whole file. In addition, this procedure is able to perform several auditing projects at the same time instead of confirming them one by one. Our trial results illustrate the efficiency and usefulness of our procedure when auditing distributed information reliability.

Keywords— Public auditing, privacy-preserving, shared data, cloud computing

I. INTRODUCTION

Cloud companies offer customers effective and scalable data storage space alternatives with a much reduced marginal cost than traditional techniques [2]. It is schedule for Customers to leverage cloud storage space alternatives to work together with others in a team, as information discussing becomes a conventional function in most cloud storage space Promotions, such as Drop box, Cloud and Google Generate. The reliability of information in cloud storage space, however, is subject to uncertainty and analysis, as information saved in the cloud can easily be missing or damaged due to the unavoidable hardware/ software problems and individual mistakes. To create this matter even more intense, cloud companies may be reluctant to notify customers about these information mistakes to be able to maintain the popularity of their alternatives and prevent losing profits [3]. Therefore, the reliability of cloud information should be verified before any information usage, such as look for or computation over cloud information The traditional strategy for verifying information correctness is to recover the whole information from the cloud, and then verify data reliability by verifying the correctness of signatures (e.g. RSA) or hash principles (e.g., MD5) of the entire data. Certainly, this traditional strategy is able to successfully check the correctness of cloud information. The primary purpose is that the size of the cloud Information is large in common. Installing the whole cloud information to verify data reliability will price or even spend Customer's quantities of computation and interaction sources, Especially when information has been damaged in the cloud. Besides, many uses of cloud information (e.g., information Exploration and machine learning) do not actually need customers to obtain the entire cloud information to regional gadgets [2]. It is because cloud providers, such as Amazon, can offer customers computation services straight on large-scale information that already existed in the cloud. However, a new significant privacy issue introduced in the case of shared data with the use of existing mechanisms is the leakage of identity privacy to public verifiers [1]. To protect the confidential information, it is essential and critical to preserve identity privacy from public verifiers during public auditing. To solve the above privacy issue on shared data, we propose Oruta1, a novel privacy preserving public auditing mechanism. More specifically, we utilize ring signatures to construct homomorphic authenticators [10] in Oruta, 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 addition, extend this mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks. Meanwhile, Oruta is compatible with random masking [5]; **Oruta stands for “One Ring to Rule Them All”**. **Our contributions in this paper are multirole.** preserving identity privacy. Ensures that retrieved data always reflects the most recent updates and prevents rollback attacks. Achieving data freshness is essential to protect against mis-configuration errors

or rollbacks caused intentionally. We can develop an authenticated file system, that supports the migration of an enterprise-class distributes file system into the cloud efficiently, transparently and in a scalable manner. It's authenticated in the sense that enables an enterprise tenant to verify the freshness of retrieved data while performing the file system operations

A. KEY MANAGEMENT

In this paper, following are the cryptographic keys to protect data files stored on the cloud.

- **Public Key:** The Public key is a random generated binary key, generated and maintained by the Key manager itself. Particularly used for encryption/ decryption.
- **Private Key:** It is the combination of the username, password and two security question of user's choice. The private key is maintained by client itself. Used for encrypt /decrypt the file.
- **Access key:** It is associated with a policy. Private access key is maintained by the client. The access key is built on attribute based encryption. File access is of read or write.
- **Renew key:** Maintained by the client itself. Each has its own renew key. The renew key is used to renew the policy of each necessary file at easy method.
-

II. RELATED WORK

The authors [12] take a centralized technique where a single key distribution center (KDC) distributes secret keys and attributes to all the users. Unfortunately, a single KDC is not only a single data of failure but difficult to maintain because of the large number of users that are supported in a cloud environment. The receiver receiving the attributes and secret keys from the attribute authority and is able to decrypt the information if it has matching attributes. All the technique take a centralized approach and allow only one KDC, which is a single point of failure. A proposed a scheme in which there are several KDC authorities (coordinated by a trusted authority) which distribute attributes and secret keys of the users. However, the presence of one proxy and one KDC makes it less robust than decentralized approach. A new scheme given by Maji et al. takes a decentralized approach and provides authentication without disclosing the identity of the users.

Attribute-Based Signature Scheme:

- a) System Initialization
- b) User Registration
- c) KDC Setup
- d) Attribute Generation
- e) Sign
- f) Verify

III. PROPOSED SCHEME

To solve the above privacy issue on shared data, we propose Oruta, a novel privacy preserving public auditing mechanism. More specifically, we utilize ring signatures to construct homomorphic authenticators in Oruta, 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 addition, we further extend our mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks. The system style in this paper involves three parties: the reasoning server, a number of users and a community verifier. There are two kinds of customers in a group: the unique customer and a number of team customers. The original customer originally makes distributed information in the reasoning, and shares it with team customers. Both the unique customer and group users are associates of the team. Every participant of the group is permitted to accessibility and change distributed information. Shared data and its confirmation meta-data (i.e., signatures) are both stored in the reasoning server. When a community verifier desires to examine the reliability of shared information, it first delivers an audit task to the cloud server. After getting the audit task, the reasoning server reacts to the community verifier with an auditing evidence of the ownership of distributed information. Then, this community verifier assessments the correctness of the entire data by confirming the correctness of the audit evidence. Essentially, the procedure of community audit is a challenge and- response method between a community verifier and the cloud server

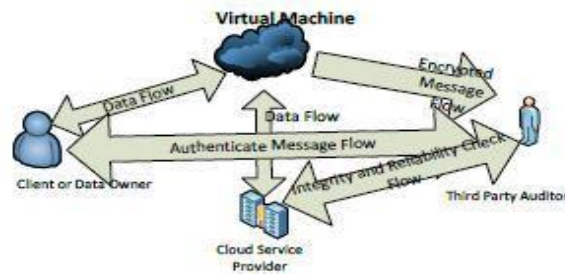


Figure 1 Architecture of Cloud server with CU and TPA.

1. **Service Request to TPA:** The user registers with the original identity and enrolls with the Third Party Authenticator (TPA). The user sends request to the Third Party Authenticator (TPA) for registration.
2. **TPA Policy Creation:** The TPA along with token provides the rules and regulation to be followed by Creator, Reader and Writer.
3. **User File Upload:** The file creator after getting proper authentication encrypts the file and uploads his files in the cloud.
4. **KDC Key Generation:** The Key Distribution Centers which are decentralized generate different keys to different types of users after getting tokens from users.
5. **Key Revocation:** Whenever there is misbehavior detected upon a user his key is revoked and that particular user can neither use or re-enter the cloud environment.
6. **Cloud Admin:** Cloud admin has the list of Key Distribution Centres (KDCs) and Third Party Authenticator (TPA). The cloud admin sets the norms to be followed by TPA and KDC. It monitors the key generation policies and informs abnormal behaviours

- (1) **Public Auditing:** A public verifier is able to publicly verify the integrity of shared data without retrieving the entire data from the cloud.
- (2) **Correctness:** A public verifier is able to correctly verify shared data integrity.
- (3) **Unforgeability:** Only a user in the group can generate valid verification metadata (i.e., signatures) on shared data.
- (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 or rollbacks caused intentionally. We can develop an authenticated file system that supports the migration of an enterprise-class distributed file system into the cloud efficiently, transparently and in a scalable manner. It's authenticated in the sense that enables an enterprise tenant to verify the freshness of retrieved data while performing the file system operations

IV. SYSTEM MODEL

A. Architecture

A novel privacy-preserving public auditing mechanism. More specifically, we utilize ring signatures [21] to construct homomorphic authenticators [10] in Oruta, 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..

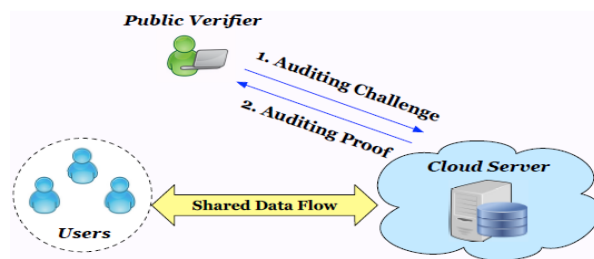


Figure 2. Our system model includes the cloud server, a group of users and public verifier.

As illustrated in Fig. 2, the system model in this paper involves three parties: the cloud server, a group of users and a public verifier. There are two types of users in a group: the original user and a number of group users. The original user initially creates shared data in the cloud, and shares it with 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, such as a thirdparty auditor providing expert data auditing services or a data user outside the group intending to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server. When a public verifier wishes to check the integrity of shared data, it first sends an auditing challenge to the cloud server. After receiving the auditing challenge, the cloud server responds to the public verifier with an auditing proof of the possession of shared data. Then, this public verifier checks the correctness of the entire data by verifying the correctness of the auditing proof. Essentially, the process of public auditing is a challenge-and-response protocol between a public verifier and the cloud server [9].

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a privacy-preserving public auditing with data freshness verification mechanism for shared data in the cloud. 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*. Each block has an associated version counter that is incremented every time the block is modified. This version number is bound to the file-block's MAC: To protect against cloud replay of stale file-blocks (rollback attacks), the counters themselves must be authenticated. The interesting problems we will continue to study for our future work. One of them is traceability, which means the ability for the group manager (i.e., the original user) to reveal the identity of the signer based on verification metadata in some special situations. Since the current design of ours does not support traceability.

References

- [1] Boyang Wang, Bao chun Li, and Hui Li, (2014), 'Oruta: Privacy-Preserving Public Auditing for Shared Data in the cloud', IEEE Trans.Computers, vol.2, no. 1, pp. 143-159.
- [2] R. Rivest, A. Shamir, and L. Adleman,(1978), 'A Method for Obtaining Digital Signatures and Public Key Cryptosystems', Comm. ACM, vol. 21, no. 2, pp.
- [3] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou,(2009), 'Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing', Proc. 14th European Conf.Research in Computer Security(ESORICS'09), pp. 355-370.
- [4] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau,(2011), 'Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds', Proc. ACM Symp. Applied Computing (SAC'11), pp. 1550-1557.
- [5] B. Wang, H. Li, and M. Li,(2013), 'Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics', Proc. IEEE Int'l Conf. Comm. (ICC'13), pp. 539 543.
- [6] B. Wang, S.S. Chow, M. Li, and H. Li,(2013), 'Storing Shared Data on the Cloud via Security Mediator', Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems(ICDCS'13),pp.124-133.