International Journal of Science Engineering and Advance Technology

Privacy Preservation and traceability support using Public Auditing Mechanism

JayanthiAnusha [1], K.LakshmiPriya [2]
[1]PG Scholar, Pydah College of Engineering, Kakinada, AP, India,
[2]Assistant Professor, Pydah College of Engineering, Kakinada, AP, India.
E-mail: anusha.jayanthi24@gmail.com.

Abstract— as future enhancement, we enhance the Oruta system in two interesting problems we will continue the 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.However, public auditing on the integrity of shared data with these existing mechanisms will inevitably reveal confidential information to identity the privacy to public verifiers. Since Oruta is based on the ring signatures, where the identity of the signer is unconditionally protected, the current design of ours does not support traceability. To the best of our knowledge, designing an efficient public auditing mechanism with the capabilities of preserving identity privacy and supporting traceability is still open. Another problem for our future work is how to prove data freshness (prove the cloud possesses the latest version of shared data) while still preserving identity privacy. In addition, our mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. Our experimental results demonstrate the effectiveness and efficiency of our mechanism when auditing shared data integrity.

Key words: Public auditing, privacy-preserving, cloud computing, shared data, TPA (third party auditor).

I. Introduction

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers. Many mechanisms 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. 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. A public verifier could be a data user (e.g., researcher) who would like to utilize the owner's data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking services. Moving a step forward, Wang et al. designed an advanced auditing mechanism .so that during public auditing on cloud data, the content of private data belonging to a personal user is not disclosed to any public verifiers. Unfortunately, current public auditing solutions mentioned above only focus on personal data in the cloud.

Protect these confidential information is essential and critical to preserve identity privacy from public verifiers during public auditing. In this paper, 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 homomorphism 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. Meanwhile, Oruta is compatible with random masking, which has been utilized in WWRL and can preserve data privacy from public verifiers. Moreover, we also leverage index hash tables from a previous public auditing solution to support dynamic data. A high-level comparison among Oruta and existing mechanisms is presented.

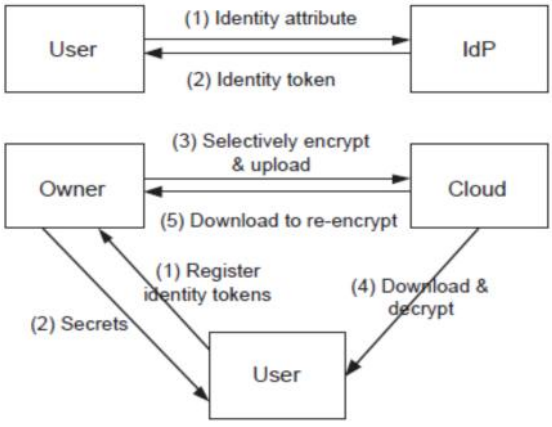


FIG 1:Public auditing model

II. RELATED WORK

With cloud data services, it is commonplace for data to be not only stored in the cloud, but also shared across multiple users. Unfortunately, the integrity of cloud data is subject to skepticism due to the existence of hardware/software failures and human errors. Several mechanisms have been designed to allow both data owners and public verifiers to efficiently audit cloud data integrity without retrieving the entire data from the cloud server. We exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, our mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. Our experimental results demonstrate the effectiveness and efficiency of our mechanism when auditing shared data integrity. In this talk, I will first discuss a number of pressing security challenges in Cloud Computing, including data service outsourcing security and secure computation outsourcing. Then, I will focus on data storage security in Cloud Computing. As one of the primitive services, cloud storage allows data owners to outsource their data to cloud for its appealing benefits. However, the fact that owners no longer have physical possession of the outsourced data raises big security concerns on the storage correctness. Hence, enabling secure storage auditing in the cloud environment with new approaches becomes imperative and challenging. In this talk, I will present our recent research efforts towards storage outsourcing security in cloud computing and describe both our technical approaches and security & performance evaluations. Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the possibly large size of outsourced data makes the data integrity protection in Cloud Computing a very challenging and potentially formidable task, especially for users with constrained computing resources and capabilities. Thus, enabling public audit ability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed.

To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met: 1) TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user; 2) The third party auditing process should bring in no new vulnerabilities towards user data privacy. In this paper, we utilize and uniquely combine the public key based homomorphism authenticator with random masking to achieve the privacy-preserving public cloud data auditing system, which meets all above requirements. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient.

III.DESIGN OBJECTIVES

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2.It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

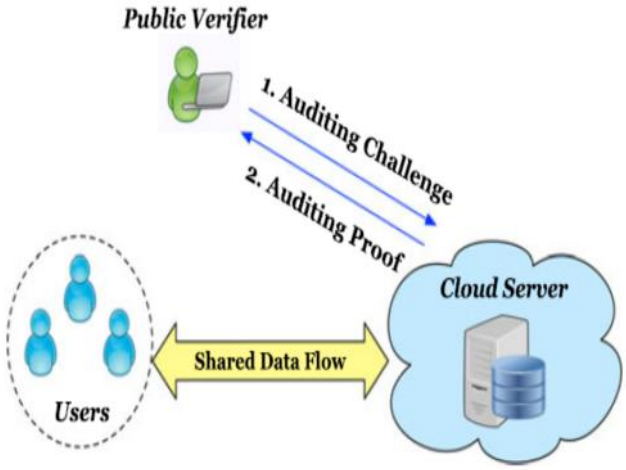


Fig:2. System architecture

SYSTEM PREMERILIERS

Cloud server

- ✓ In the first module, we design our system with Cloud Server, where the datas are stored globally. Our mechanism, Oruta, should be designed to achieve following properties:
- ✓ (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.

Group of users

- ✓ 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.
- ✓ Owner Registration: In this module an owner has to upload its files in a cloud server, he/she should register first. Then only he/she can be able to do it. For that he needs to fill the details in the registration form. These details are maintained in a database.
- ✓ Owner Login: In this module, owner have to login, they should login by giving their email id and password.
- ✓ User Registration: In this module if a user wants to access the data which is stored in a cloud, he/she should register their details first. These details are maintained in a Database.
- ✓ User Login: If the user is an authorized user, he/she can download the file by using file id which has been stored by data owner when it was uploading.

Public verifier

- ✓ 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

Auditing Module

- ✓ In this module, if a third party auditor TPA (maintainer of clouds) should register first. This system allows only cloud service providers. After third party auditor gets logged in, He/ She can see how many data owners have uploaded their files into the cloud. Here we are providing TPA for maintaining clouds.
- ✓ 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. Another interesting problem is how to audit the integrity of shared data in the cloud with *dynamic groups* — a new user can be added into the group and an existing group member can be revoked during data sharing — while still preserving identity privacy.

IV.PROPOSED SYSTEM

- ☸ In this paper, 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.
- ☸ Meanwhile, Oruta is compatible with random masking, which has been utilized in WWRL and can preserve data privacy from public verifiers. Moreover, we also leverage index hash tables from a previous public auditing solution to support dynamic data. A high-level comparison among Oruta and existing mechanisms is presented.

USER REGISTRATION

If a user wants to access the data which isstored in a cloud, he/she should register theirfirst. These details are maintained in a Database.

USER LOGIN

If the user is an authorized user, he/she candownload the file which has been stored by dataowner. The user can modify the data by threedifferent types of operations they are
i. insert
ii. Delete
iii. update

1. INSERT

The user m joins to theshared data. He/she inserts a new block computes the new cloudserver. For the rest of the block, the identifiers ofthese blocks are not changed.

Identifier of the inserted block m j and then uploads to the total number of theblock in the shared data increases to n+1.

Index	Block
1	M_1
2	M_2
.	.
.	.
N	M_n

Insert

Index	Block
1	M_1
2	M_2
3	M_3
.	.
.	.
n-1	M_n

Table 1 inserting a new data block

2. DELETE

The user deletes block mj, its identifier idjfrom the cloud server. The identifier and content ofother blocks in shared data are remaining the same.The total number of blocks in the shared datadecreases to n-1. For deletion operation the user don'twant to compute new identifier.

Index	Block
1	M_1
2	M_2
3	M_3
.	.
N	M_n

Delete

Index	Block
1	M_1
2	M_3
.	.
.	.
n-1	M_n

Table 2 deleting a data block

3. UPDATE

The user updates the j-th block in the shareddata with the new block m j. The new identifier ofthis block is updated and the identifier of the othershared data is not updated. The total number ofblocks in shared data id still n.

Index	Block
1	M_1
2	M_2
3	M_3
.	.
.	.
N	M_n

Update

Index	Block
1	M_1
2	M_2
3	M_3
.	.
.	.
n	M_n

Table 3 updating a data block

PARTY AUDITOR
THIRD PARTY AUDITOR REGISTRATION

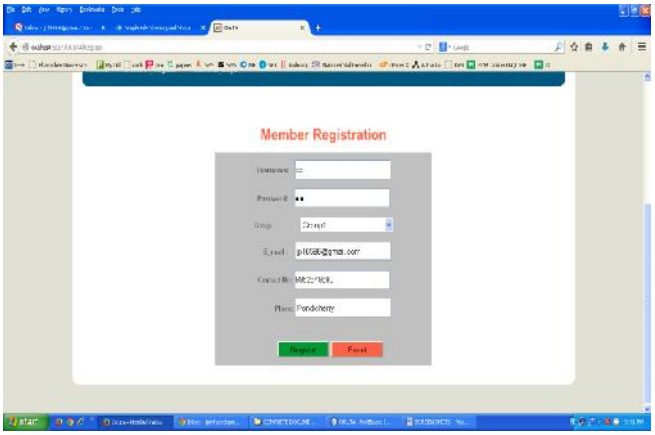
The third party auditor (maintainer of clouds)wants to do some cloud offer; they should registerfirst, to do auditing on the cloud data.

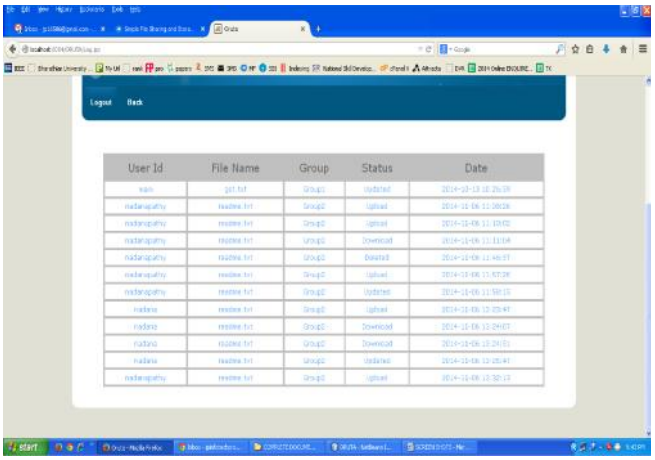
THIRD PARTY AUDITOR LOGIN

After third party auditor gets logged in, He/she can see how many data owners have uploadedtheir files into the cloud and who are the user send anauditing request. The third party auditor the audit thedata if any modifications are made then the publicverifier send a report to the data owner with theoriginal data and the modified data.

RESULTS

SCREEN SHOTS





VII. CONCLUSION

We utilize ring signatures to construct homomorphic authenticators, study for our future so that a public verifier is able to audit shared data integrity without retrieving the entire data, yet it cannot distinguish block. To improve the efficiency of verifying multiple auditing tasks, we further extend our mechanism who is the signer on each to support batch auditing. There are two interesting problems we will continue to work. One of them is traceability, which means the ability for the group manager to reveal the identity of the signer based on verification metadata in some special situations. Since Oruta is based on ring signatures, where the identity of the signer is unconditionally protected, the current design of ours does not support traceability.

VIII. REFERENCES

[1] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE Fifth Int'l Conf. Cloud Computing, pp. 295-302, 2012.

[2] 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," Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[3] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[4] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012.

[5] 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.

[6] B. Wang, M. Li, S.S. Chow, and H. Li, "Computing Encrypted Cloud Data Efficiently under Multiple Keys," Proc. IEEE Conf. Comm. and Network Security (CNS '13), pp. 90-99, 2013.

[7] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120-126,1978.

[8] The MD5 Message-Digest Algorithm (RFC1321). https://tools. ietf.org/html/rfc1321, 2014.

[9] 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.

[10] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.