Asian Journal of Science and Applied Technology (AJSAT)
Vol.3.No.1 2015 pp 6-9
available at: www.goniv.com
Paper Received :08-03-2015
Paper Accepted:20-03-2015
Paper Reviewed by: 1. R. Venkatakrishnan 2. R. Marimuthu
Editor : Prof. P.Muthukumar

PRIVACY - PRESERVING MULTIPLE AUDITING FOR DISTRIBUTED DA1TA IN THE CLOUD STORAGE

[1]M.DEVIKA, [2]A.ANTONIDOSS, M.E (PH.D.),
[1]M.E CSE (Student), [2]Asst. Professor
Tagore Engineering College
Chennai, India
[1]devima1310@gmail.com

ABSTRACT

In the cloud data service, the data are not only stored in single location, but also distributed among multiple users. There exists integrity skepticism of data due to human error and hardware/software failure. Many mechanisms have been proposed to allow not only data owner but also public verifier to perform the integrity checking. A public verifier is able to verify the correctness of distributed data. In this paper, we propose a novel privacy preserving mechanism that supports multiple auditing on data distributed in cloud. We exploit homomorphic ring signature to audit the correctness of distributed data. We also ensure cloud data freshness, based on random auditing mechanism. With our mechanism we can able to audit the data without downloading the entire file. The identity of the signer on each chunk in distributed data is kept private from public verifier. Multiple auditing is also able to perform multiple tasking simultaneously on each chunk of data. Our experimental results helps to secure and efficient dynamic operations on cloud data.

Index Terms –Multiple auditing, Cloud computing, Distributed data, Ring signature

I.INTRODUCTION

Cloud computing is type of computing, offers users efficient and scalable data storage services with a much lower marginal cost than traditional approaches. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes a standard feature in most cloud storage offerings, including Dropbox, icloud and Google Drive. In cloud computing, the word cloud is used as a metaphor for "the Internet,", so the phrase cloud computing means a type of Internet-based computing. The fundamental services provided by cloud computing is the data storage. The data that are stored in cloud is not only stored in single location, but also distributed among multiple users. There exists integrity skepticism of data due to human error and hardware/software failure.

To make the matter even worse, the cloud service providers may be unwilling to inform users about these errors to avoid losing profits. In order to maintain the correctness of the data, the entire data is retrieved from the cloud and then verified using signatures or hash values of the entire data. The efficiency of using this approach on cloud data is in doubt because the size of the entire data is large. Downloading the entire data to verify the data integrity will cost more and waste of computation time too.

Many mechanisms have been proposed to allow not only a data user but also a public verifier to perform integrity checking without downloading the entire file from the cloud, which is referred to as multiple auditing. In this mechanism, data is divided into number of chunks, where each chunks is independently signed by the owner. A public verifier will randomly check the correctness of the data. This leads to skepticism that owner data is retrieved during the integrity checking.

In this paper, the privacy issues on the distributed data is solved using ring signatures to construct homomorphic authenticators, so that public verifier is able to verify the integrity of distributed data without downloading the whole data from the cloud. The identity of the signer on each chunk in the distributed data is kept private from the public verifier. We also extend our mechanism to batch auditing, which can perform multiple auditing simultaneously.

II.PROBLEM STATEMENT

A .System Model

The system model in this paper involves four members: the cloud server, the data owner, the group users and a public verifier. The data owner creates distributed data and shares it in the cloud. Both the data owner and the group users are members of the group. Every member of the group is allowed to access and modify the distributed data. The distributed data and its signature are stored in the cloud server. A public verifier, who do auditing mechanism services or the data owner outside the group can able to utilize the distributed data, is able to publicly verify the integrity of distributed data stored in the cloud server.

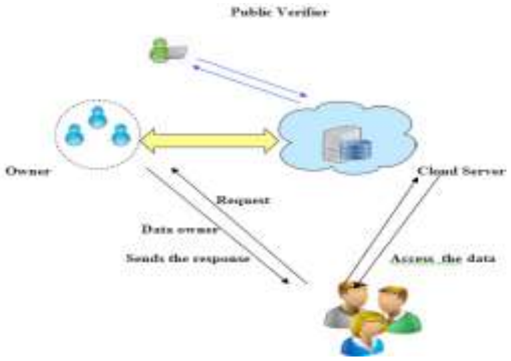

Figure 1. System model

When a public verifier checks the integrity of distributed data, it first sends an auditing challenge to the cloud server. Then, the cloud server responds to the public verifier with an auditing proof of the distributed data. Then, the public verifier checks for the freshness of the data by verifying the proof. The process of public auditing is a challenge–and–responsive protocol between a public verifier and the cloud server.

B. Design Objectives

(1) Public Auditing: A public verifier is able to publicly verify the integrity of distributed data without retrieving the entire data from the cloud.

(2) Correctness: A public verifier is able to correctly verify distributed data integrity.

(3) Unforgeability: Only a user in the group can generate valid signatures on distributed data.

(4) Identity Privacy: A public verifier cannot differentiate the identity of the signer on each chunk in distributed data during the process of auditing.

(5) Data Freshness: Data Freshness is essential to protect against mis-configuration errors or rollbacks caused intentionally.

III.PROPOSED WORK

A. New Ring Signature Scheme Overview

The design of new Homomorphic Authenticable Ring Signatures (HARS) scheme, which is extended from a classic ring signature scheme. The ring signatures generated by HARS are not only able to preserve identity privacy but also able to support block less verifiability.

B. Construction of HARS

HARS contains three algorithms: KeyGen, RingSign and RingVerify. In KeyGen, each user in the group generates his/her public key and private key.

In RingSign, a user in the group is able to generate a signature on a block and its block identifier with his/her private key and all the group member's public keys. A block identifier is a string that can differentiate the corresponding block from others. A verifier is able to check whether a given block is signed by a group member in RingVerify.

C. Multiple Auditing Mechanism

1. Multiple Auditing

Now, we present the details of our multiple auditing mechanism. It includes five algorithms: KeyGen, SigGen, Modify,ProofGen and ProofVerify. In KeyGen, users generate their own public/private key pairs.

In SigGen, a user is able to compute ring signatures on chunks in the distributed data by using its own private key and the entire group member's public key. Each user in the group is able to perform an insert, delete or update operation on a chunk, and compute the new ring signature on this new block in Modify. ProofGen is operated by a public verifier and the cloud server together to interactively generate a proof of possession of the distributed data.

In ProofVerify, the public verifier audits the integrity of the distributed data by verifying the proof. Note that for the ease of understanding, we first assume the group is static, which means the group is predefined before distributed data is created in the cloud and the membership of the group is not changed during data sharing. Specifically, before the original user outsources the distributed data to the cloud, he/she decides all the group members.

2. Dynamic Groups

We now discuss the scenario of dynamic groups under our proposed mechanism. If a new user can be added in the group or an existing user can be revoked from the group, then this group is denoted as a dynamic group. To support dynamic groups while still allowing the public verifier to perform multiple auditing, all the ring signatures on the distributed data need to be re-computed with the signer's private key and all the current users' public keys when the membership of the group is changed.

IV.ANALYTICAL MODEL

A.Security Analysis

Now, we discuss security properties, including its correctness, unforgeability, identity privacy and data privacy.

Theorem: A public verifier is able to correctly audit the integrity of shared data.

Proof: According to the description of **ProofVerify,** a public verifier believes the integrity of shared data is correct. So, the correctness of our scheme can be proved by verifying the correctness.

Based on properties of bilinear maps and Theorem 1, the right-hand side (RHS) can be expanded as follows:

$$\begin{aligned}
\text{RHS} \vdash\ & \left(\prod_{i=1}^{d} e\Big(\prod_{j\in\mathcal{J}} \sigma_{j,i}^{y_j}, w_i\Big)\right)\cdot e\Big(\prod_{l=1}^{k}\lambda_l^{h(\lambda_l)}, g_2\Big)\\
=\ & \left(\prod_{j\in\mathcal{J}}\Big(\prod_{i=1}^{d} e(\sigma_{j,i}, w_i)^{y_j}\Big)\right)\cdot e\Big(\prod_{l=1}^{k}\eta_l^{\tau_l h(\lambda_l)}, g_2\Big)\\
=\ & \left(\prod_{j\in\mathcal{J}} e(\beta_j, g_2)^{y_j}\right)\cdot e\Big(\prod_{l=1}^{k}\eta_l^{\tau_l h(\lambda_l)}, g_2\Big)\\
=\ & e\Big(\prod_{j\in\mathcal{J}}\Big(H_1(id_j)\prod_{l=1}^{k}\eta_l^{m_{j,l}}\Big)^{y_j}, g_2\Big)\cdot e\Big(\prod_{l=1}^{k}\eta_l^{\tau_l h(\lambda_l)}, g_2\Big)\\
=\ & e\Big(\prod_{j\in\mathcal{J}} H_1(id_j)^{y_j}\cdot\prod_{l=1}^{k}\eta_l^{\sum_{j\in\mathcal{J}} m_{j,l} y_j}\cdot\prod_{l=1}^{k}\eta_l^{\tau_l h(\lambda_l)}, g_2\Big)\\
=\ & e\Big(\prod_{j\in\mathcal{J}} H_1(id_j)^{y_j}\cdot\prod_{l=1}^{k}\eta_l^{\mu_l}, g_2\Big).
\end{aligned}$$

B. Batch Auditing

Sometimes, a public verifier may need to verify the correctness of multiple auditing tasks in a very short time. Directly verifying these multiple auditing tasks separately would be inefficient. By leveraging the properties of bilinear maps, we can further extend our mechanism to support batch auditing, which can verify the correctness of multiple auditing tasks simultaneously and improve the efficiency of public auditing.

V. RELATED WORK

Provable data possession (PDP), proposed by Ateniese et al., allows a verifier to check the freshness of a client's data saved at an untrusted server. By utilizing RSA-based homomorphic authenticators and sampling strategies, the verifier is able to publicly check the integrity of data without downloading the entire data. Unfortunately, their mechanism is only suitable for checking the integrity of personal data. Juels and Kaliski defined another similar model called Proofs of Retrievability (POR), which is also able to check the freshness of data on a server. The original documents is added with a set of randomly valued check blocks called sentinels.

The verifier challenges the server by specifying the positions of sentinels and asking the untrusted server to return the associated sentinel values. Shacham and Waters designed two improved schemes. The first scheme is built from BLS signatures, and the second one is based on false-random functions. To support effective data, Ateniese et al. presented an efficient PDP mechanism based on symmetric keys. This mechanism allows for update and delete operations on data; however, insert operations are not available in this method. Because it uses symmetric keys to check the integrity of data, it is not public verifiable and only provides a user with a limited number of verification requests. Wang et al. utilized Merkle Hash Tree and BLS signatures to support effective data in a public auditing method. Erway et al. introduced dynamic provable data possession (DPDP) by using authenticated dictionaries, which are based on rank information. Zhu et al. exploited the fragment structure to reduce the storage of signatures in their public auditing mechanism.

In addition, they also used index hash tables to provide dynamic operations on data. The public mechanism proposed by Wang et al. and its journal version are able to preserve users' confidential data from a public verifier by using random masking. In addition, to operate multiple auditing tasks from different users efficiently, they extended their mechanism to enable batch auditing by leveraging aggregate signatures.

Wang et al. leveraged homomorphic tokens to ensure the correctness of erasure codes-based distributed data on multiple servers. This method is able not only to support effective data, but also to identify aweless servers. To minimize communication overflow in the stage of data repair, Chen et al. also introduced a mechanism for checking the freshness of data under the multiple server scenario, where these documents are encrypted by network coding instead of using erasure codes. Compare to previous work this mechanism can avoid high decoding computation cost for data users and save computation resource for online data owners during data repair.

VI.CONCLUSION

In this paper, we propose Privacy Preserving Multiple Auditing for Distributed Data in the Cloud Storage. We utilize ring signatures to construct homomorphic authenticators, so that a public verifier is able to audit shared data integrity without retrieving the entire data, yet it cannot distinguish who is the signer on each block.

VII.FUTURE WORK

To improve the efficiency of verifying multiple auditing tasks, we further extend our mechanism to support batch auditing. There are two 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 our mechanism is based on 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 multiple auditing mechanisms 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.

REFERENCES

[1] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.

[2] B. Wang, B. Li, and H. Li, "Panda: Public Auditing for Shared Data with Efficient User

Revocation in the Cloud," IEEE Trans. Services Computing, 20 Dec. 2013.

[3] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1182-1191,June 2013.

[4] 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.

[5] 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.

[6] 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.

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

[8] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds," Proc. ACM Symp. Applied Computing (SAC'11), pp. 1550-1557, 2011.

[9] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," in Proceedings of ESORICS 2009. Springer-Verlag, 2009, pp. 355–370.

[10] Wang, Q. Wang, K. Ren, and W. Lou,"Ensuring Data Storage Security in Cloud *Computing,"Proceedings of ACM/IEEE IWQoS'09, 2009, pp.1-9*