I J C T A, 9(15), 2016, pp. 7057-7063

Privacy Preserving Secure Auditing for Shared Data in the Cloud

Priyanka Kundu* and A.M.J. Muhukumaran**

ABSTRACT

The cloud computing technology which develops at the last decade and outsourcing data to cloud service for storage for an attractive trend, which will give benefit in sparing efforts on heavy data maintenance and management. The outsource cloud storage is not complete trustworthy, which raises security concerns on how to realize data to avoid duplication in the cloud while achieving integrity auditing. Here we study the problem of integrity analysis and secure to de-duplication of cloud data. Which aiming at achieving both data integrity and to de-duplication in the cloud, we propose two secure systems, namely Sec-Cloud and Sec-Cloud+. Sec-Cloud introduces an auditing entity with the maintenance of a Map Reduce cloud; that helps to clients for generating data tags before uploading and audit the integrity of data which have been store in the cloud. In previous work, the computation by the user in Sec-Cloud is reducing during the file uploading and auditing phases. The Sec-Cloud+ is designed to motivate by the fact that customers always want to encrypt their data before uploading and also enables the integrity analysis for secure to avoid duplication of encrypted data.

Keywords: Role-based access control, Role-based encryption, Provable data possession, Certificate Authority, Attributes Authorities, Attribute-based encryption, Advanced Encryption Standard.

1. INTRODUCTION

Cloud service use to manage an enterprise-class infrastructure. This infrastructure offers a secure and reliable environment for users, at a lower marginal cost due to the sharing nature of resources. The routine for users to use cloud storage services to share the data with others in a group, as the data become a standard feature in most cloud storage for offerings and including Google Docs. In the cloud storage the data integrity is described to skepticism and security, as data stored in an authorized cloud can be easily lost or corrupted, due to the hardware failures and the human error. To protect the integrity of cloud data is best to perform public auditing by introducing a third party auditor who offers its analyzing service with more computation and communication abilities than regular users. The first provable data possession mechanism to perform public auditing is designed to check the correctness of data stored in the entrusted server, without retrieving the entire data. It is create to construct a public auditing mechanism for cloud data, so the private data belonging to a personal user is close for the third party auditor. We believe that to sharing data among multiple users is perhaps one of the most engaging features which motivate cloud storage. A unique problem introduced in public auditing for shared information in the cloud. To share data may indicate that a particular user in the group or a block is allocating data which is a more valuable target than others. After performing several auditing tasks, some private and sensitive information may reveal to the third party auditor. On one hand, in the shared file most of the blocks are signed by the auditor who may indicate to Alice is an important role in this group, such as a group leader. On the other hand, the eighth block is frequently modified by different users. This block may contain high-value data, such as a final bit in an auction.

* Department of Computer science and engineering SRM University Chennai, India, *Email: priyankakunduvel@gmail.com*

** Department of Computer science and engineering SRM University Chennai, India, *Email: muthu.a@ktr.srmuniv.ac.in*

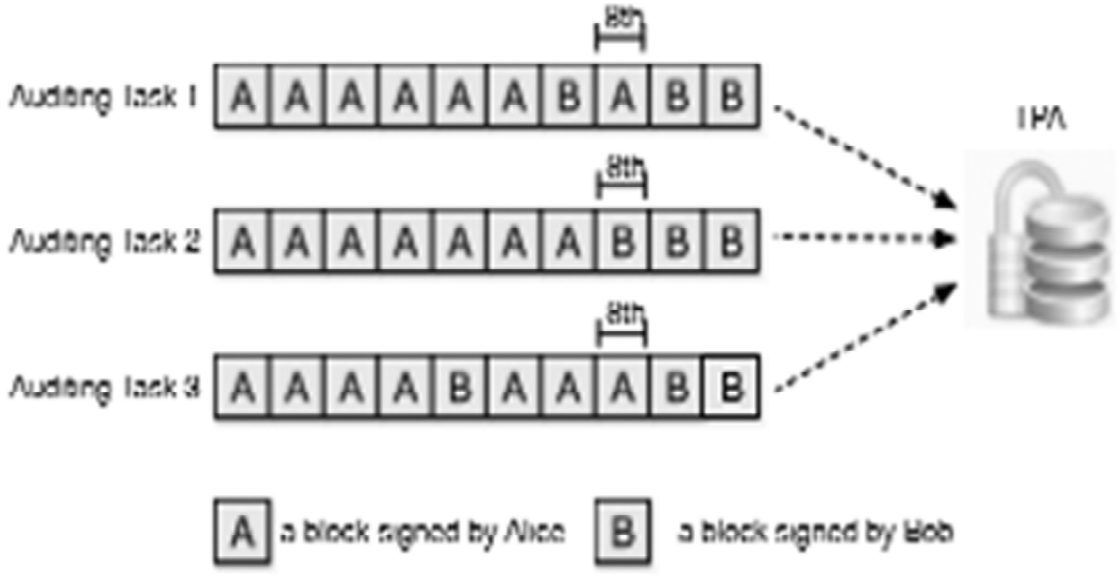

Figure 1: The TPA audits the integrity of data with existing mechanisms.

Public auditing mechanism for shared data in an authorized the cloud. We utilize ring signatures to construct homomorphism authenticator so that the third party auditor can verify the integrity of shared data. For the identity of the signer on each block in shared data is kept private from the third party auditor. Also, to continue the support data privacy and dynamic operations on data during public auditing.

2. PROBLEM STATEMENT

Existing system access control model is a Role-based access control (RBAC), which provides flexible controls and also, management, has two mappings, users to roles and these roles to privileges on data objects. In the existing system a role-based encryption (RBE) scheme that integrates the cryptographic techniques with role-based access control. Our role-based encryption allows role-based access control policies to be enforcing for the encrypted data stored in public clouds. The role-based access control is a method for regulating access to computer or network resources based on the roles of the individual users. Here, access is the ability of the individual user to perform a specified task, such as the view, create and modify a file. Roles are use to define according to the job competency, authority, and responsibility within the enterprise. In our role-based encryption scheme, the owner of the data use to encrypts the data with appropriate roles. This role-based access control policy can decrypt and view the data. The role grants for permissions to the users who qualify the aspect and can also revoke the permissions from existing users.

Role names are still associate with users, but the consideration that roles are collections of permissions is no longer in the case. The Security loses because of the user's aspect to set determines the maximum set of available permissions, supporting to the principle of least privilege and allowing review of user permissions. Dynamically changing attributes of user role, such as time of day and location.

2.1. Integrity Auditing

The provable data possession (PDP) was defined by Ateniese et al. for assuring the cloud servers possess to target ûles without retrieving or downloading the whole data. Essentially, provable data possession is a probabilistic proof protocol by sampling a random set of blocks and asking the servers to prove which they exactly possess these blocks, and the verifier only maintaining a small amount of metadata can perform an integrity checking. After Ateniese et al.'s proposal several works concerned on how to realize provable data possession on the dynamic scenario: Ateniese et al. proposed a Dynamic provable data possession schema without insertion operation; Erway et al. improved the Ateniese et al.'s work and also support the insert

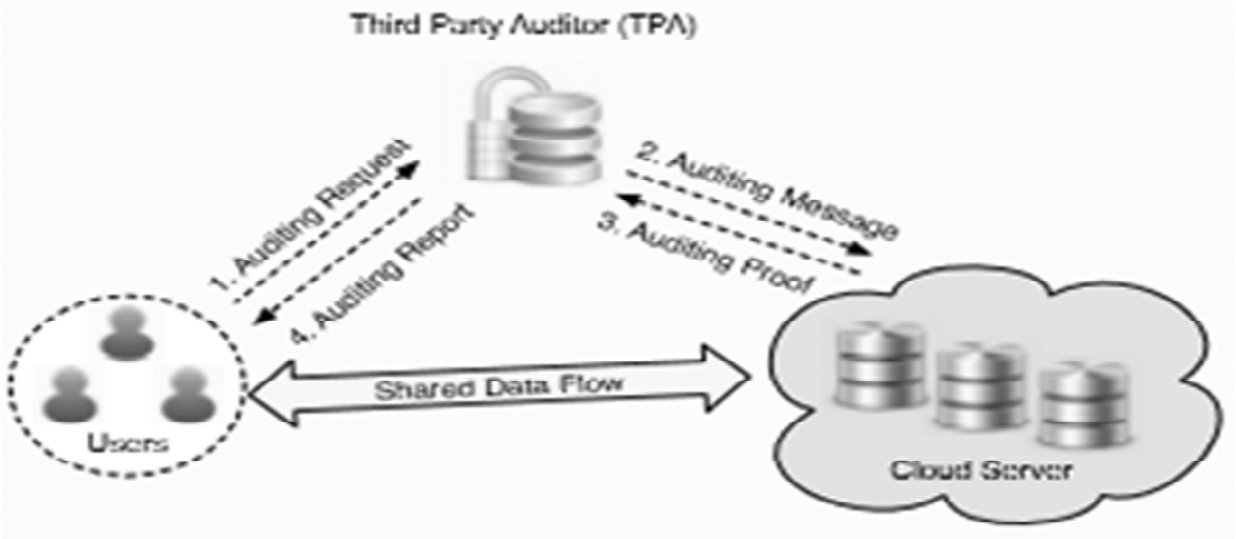

Figure 2: Our system model includes the cloud server, the third party auditor and users.

operation by introducing authenticated flip table. Nevertheless, this object is affected by the computational overhead for tag generation at the client. To fix this issue, Wang et al. proposed proxy provable data possession in public the cloud. Zhu et al. recommended the cooperative provable data possession in multi-cloud storage. Another line of work supporting integrity auditing is proof of retrievability (POR). Compared with provable data possession, confirmation of retrievability not merely assures the cloud server possess the target files, but also guarantee their full recovery. In clients to apply erasure codes and generates an authenticator for each block for verifiability and retrievability. To achieve the data dynamically, Wang et al. improved the proof of retrievability model for manipulating the classic Merkle hash tree construction for block tag authentication. Xu and Chang proposed to improve the verification of retrievability schema in with polynomial commitment for reducing communication cost. Stefanov et al. proposed a proof of retrievability protocol over authenticated file to system subject for frequent changes. Azraoui et al.combined the privacy-preserving word search algorithm with the insert to data segments of randomly generated short bit sequences and also developed by a new proof of retrievability protocol. Li et al. considered new cloud storage architecture with two independent cloud server for integrity auditing to reduce the computation load on the client side. Recently, Li et al. utilized the key disperse paradigm to fix the issue of the significant number of convergent keys in convergent encryption.

2.2. Secure Reduplications

To avoid duplication is a technique where the server stores only a single copy of each file, where many clients asked to store that file, such that the disk space of cloud servers, as well as the network bandwidths, are saved. However, the trivial client side to avoid duplication leads to the leakage of channel side information. For example, a server said to an applicant that it need not send the file reveals that some other client has the same file. To restrict the leakage of channel side information, Halevi et al. Introduced the proof of ownership protocol which lets a client prove efficiently to a server that that the applicant holds this file. The several proofs of ownership protocols based on the Merkle hash tree are proposed to enable secure client-side to avoid duplication. Pietro and Sorniotti proposed an efficient proof of ownership scheme by choosing the projection of the file onto some randomly selected bit positions as the register confirmation. Another line of work for secures to de-duplication focuses on the confidentiality de-duplicated data and considers avoiding duplication for encrypted data. Ng et al. firstly introduced the private data de-duplication as a complement of public data to avoid duplication protocols of Halevi et al. Convergent encryption is a promising cryptographic primitive for ensuring data privacy in to avoid duplication. Bellare et al. formalized this primitive as message-locked encryption and explored its application in space-efficient secure outsourced

storage. Abadi et al. further strengthened Bellare et al.'s security definitions by considering plaintext distributions that depend on the public parameters of the conception. Regarding the practical implementation of convergent encryption for securing to avoiding duplication, Keelveedhi et al. Designed the less system in which clients encrypt under file-based keys derived from a key server via an oblivious pseudo-random function protocol. Either integrity auditing or to avoid duplication, while in this paper, we attempt to solve both problems simultaneously. Also, it is a worthwhile noting that our work is also use to distinguished by which audits cloud data with to duplication because we also consider to a outsource the computation of tag generation, the auditing and to avoid duplicate encrypted the information in the proposed protocols.

3. METHODOLOGIES

3.1. System Initialization

The module consists of a Certificate Authority (CA) and Attributes Authorities (AAs), to arrange the proposed system. The certificate authority is a globally trusted certificate authority in the system. It sets up the system and accepts the registration of all the users. The attributes authority in the system is for each legal user in the system; the certificate authority assigns with a globally unique user identity to it and also generates a global public key for this user. The certificate authority is not involved in any attribute management and also the creation of secret keys which is an accomplice with attributes.

3.2. Secret Key Generation by Attributes authority

The secret key generation algorithm is run by each attribute authority. Every attribute is an accomplice with a single attribute authority, but each attribute authority can manage an arbitrary number of attributes. Each attribute authority has full control over the structure and semantics of its attributes. The attribute authority is responsible for generating an attribute public key for each attribute to manage and a secret key for each user for reflecting his/her attributes. Every user has a global identity in the system. A user may be entitled to a set of attributes which may come from multiple attribute authorities. The user will receive a secret key that associated with its attributes designated by the corresponding attribute authorities.

3.3. Data Encryption by Owner

To encrypt the data m with content keys by using symmetric encryption methods, and then they encrypt the content keys by running the encryption algorithm. Each owner first divides the data into several components according to the logic granularity and encrypts each data composing with different content keys by using symmetric encryption techniques. This symmetric encryption is a fast encryption algorithm. The owner defines the access policies over attributes and also encrypts the content keys under the policies.

4. RELATED WORK

The present system achieves a goal by exploiting and uniquely combining techniques of attribute-based encryption (ABE) and the Advanced Encryption Standard (AES).So the recent system shows secure ABE-based hybrid cloud storage architecture which allows an organization to store data securely in a public cloud that related to the organization structure in a private cloud. In the ABE, data are associated with the attributes for each of which a public key component is defined. The encrypted associates the set of attributes to the message by encrypting it with the corresponding public key elements based on proposed ABE scheme; develop a secure cloud data storage architecture using a hybrid cloud infrastructure. This hybrid cloud architecture is a combination of private cloud and a public cloud, where the private cloud is used to store in the organization sensitive structure information such as the role hierarchy and the user membership information, and the public cloud is used to store the actual data that is in the encrypted form.

ABE is a rule-based approach to access control that can be newer and simpler to implement and also accommodates real-time environmental states as access control parameters.ABE attributes can change dynamically, determining a user's potential permission to set will also require some rules with all possible attribute values. The proposed scheme enables the data owner to delegate most of the computation-intensive tasks to the cloud server without disclosing the data contents or user access privilege information.

4.1. System Model

In our work in this paper involves three parties that are the cloud server, the third party auditor (TPA) and the users. There are two types of users in a group: one is the original user, and another one is some group users. The original user and group users are both members of the organization. The group members are allowed to access and modify the shared data created by the original user based on access control policy. Shared data and its verification information are use to store in the cloud server. The third party auditor can verify the integrity of shared data in the cloud server on behalf of group members.

In this paper, we only consider that how to audit the integrity of shared data in the cloud with static groups. It means that the group is pre-defined before shared data is use to create in the cloud and the membership of users in the organization does not change during the data sharing. The original user is responsible for deciding that can share her data before outsourcing the data to the cloud. Another problem is audit the integrity of shared data in the cloud with dynamic groups, a new user can be add in the group, and an existing group member can be remove during data sharing. We will leave this problem to our future work. At the time of checking the integrity of shared data, she first sends an auditing request to the TPA and after receiving the verification request; the TPA generates an auditing message to the cloud server and retrieves an investigation proof of shared data from the cloud server. Then the TPA gets results of the auditing proof. Finally, the TPA sends an auditing report to the user based on the result of verification.

4.1.1. Design Objectives

The TPA efficiently and securely verifies shared data for users, Orta should be form by following the properties: (1) Public Auditing: The third party auditor can verify the integrity of shared data for a group of a user without any retrieving the entire data. (2) Correctness: It is the third party auditor who can detect correctly, (3) Enforceability: Only a user in the group can generate valid information on shared data. (4) Identity Privacy: During the auditing, the TPA cannot distinguish the identity of the signer on each of the blocks in shared data.

4.1.2. SEC-CLOUD+

We specify that our proposed Sec-Cloud system has achieved both integrity auditing and file to avoid duplication. However, it cannot prevent the cloud server from the knowing the content of files having been a store. The functionalities of integrity auditing and secure to avoid duplication are introduced on plain files. In this section, we propose Sec-Cloud+, which allows for integrity check and to avoiding duplication of encrypted information.

4.2. System Models

Compared with Sec-Cloud, our proposed Sec-Cloud+ is involved an additional trusted entity, namely key server, which is responsible for assigning the clients with a secret key for encrypting files. Our work is distinguished by the previous work by allowing for integrity auditing on encrypted data. Sec-Cloud+ follows the same three protocols as with Sec-Cloud. The only difference is the file uploading protocol in Sec-Cloud+ involves an additional phase for communication between cloud client and the main server. Then the client communicates with the key server. So that can get the convergent key for encrypting the uploading

file before the phase 2 in Sec-Cloud. Unlike Sec-Cloud, another design goal of information confidentiality is desired in the Sec-Cloud+ as follows. Data Confidentiality. The design goal of information affection requires preventing the cloud servers from accessing the content of files which needs to be resistant to "dictionary attack."That is, even the adversaries have pre-knowledge to the "dictionary" which includes all the possible files, and that they still cannot recover the target file.

5. CONCLUSION

Aiming at achieving both data integrity and de-duplication in the cloud, we propose Sec-Cloud and Sec-Cloud+. Sec-Cloud introduces an auditing entity with the maintenance of a Map Reduce cloud, which helps the clients to generate data tags before uploading as well as audit the integrity of data having been store in the cloud. Also, Sec-Cloud enables secure to avoid duplication through introducing a Proof of Ownership protocol and preventing the leakage of channel side information in data to avoid duplication. Compared with previous work, the computation by the user in Sec-Cloud is reduced during the file uploading and auditing phases. Sec-Cloud+ is an advanced construction motivated by the fact which customers always want to encrypt their data before uploading, allows for integrity checking and secure de-duplication directly on encrypted data.

ACKNOWLEDGEMENT

The proposed method gives better results as compared to previous techniques. Integrity auditing and secure to avoid duplication is robust against Sec-Cloud and Sec-Cloud+ in a real world.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, R. Katz, A. Konwinski, G. Lee, D. Patterson, and M. Zaharia, "A view of cloud computing," vol. 53, no. 4, pp.50–58, 2010.

[2] J. Yuan and S. Yu, "The secure and the constant cost public cloud storage for auditing with reduplication," in IEEE Conference on Communications and Network Security, 2013, pp. 145–153.

[3] S. Halevi, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in Proceedings of the 18th ACM Conference on Computer and Communications Security. ACM, 2011, pp. 491–500.

[4] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Server aided encryption for deduplicated storage," in Proceedings of the 22Nd USENIX Conference on Security,2013, pp. 179–194.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, Z. Peterson, and D. Song, "Provable the data possession at trusted stores," in Proceedings of the 14th ACM Conference on Computer and Communications Security, USA: ACM, 2007.

[6] G.Ateniese, R.Burns, R.Curtmola, J.Herring, O.Khan, L.Kissner, Z.Peterson, and D. Song, "Remote data checking using provable data possession," ACM Trans. Inf. Syst. Secur., vol. 14, no. 1, pp. 12:1–12:34, 2011.

[7] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "The scalable and the provable data possession," in Proceedings of the 4th International Conference on Security and Communication Networks,'08. New York, NY, USA: ACM, 2008.

[8] C. Erway, A. Kupc¸ u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proceedings of the 16th ACM Conference on Computer and Communications Security, USA: ACM, 2009, pp. 213–222.

[9] F. Sebe, J. Domingo-Ferrer, A. Martinez-Ballester, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. on Knowl. And Data Eng., vol. 20, no. 8, pp. 1034–1038, 2008.

[10] H. Wang, "The Proxy provable data possession in public Clouds," IEEE Transactions on Services Computing, vol. 6, no. 4, pp. 551–559, 2013.

[11] Y. Zhu, G.J. Ahn, and M. Yu, "Cooperative Provable data possession for integrity verification in the cloud storage," IEEE Transaction on Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231–2244, 2012.

[12] H. Shacham and B. Waters, "Compact Proofs of a retrievability," in Proceed of the 14th International Conference on the Theory and Application of Cryptology and Information Security,08. Springer Berlin Heidelberg, 2008.

[13] M. Armbrust, A. Fox, 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, 2010.

[14] G. Ateniese, R. Burns, R. Curtmola, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Entrusted Stores," in Proc.ACM CCS, 2007, pp. 598–610.

[15] C. Wang, Q. Wang, and W. Lou, "Public Auditing for Data Storage Security in Cloud Computing," in Proc. IEEE INFOCOM, 2010, pp. 525–533.

[16] R. L. Rivest, A. Shamir, and Y. Truman, "How to Leak a Secret," in Proc. ASIACRYPT. Springer-Verlag, 2001, pp. 552–565.

[17] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," in Proc. EUROCRYPT. Springer-Verlag, 2003, pp. 416–432.

[18] H. Shacham and B. Waters, "Compact investigation of a retrievability," in Proc. ASIACRYPT. Springer-Verlag, 2008, pp. 90–107.

[19] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and the Data Dynamic for Storage Security in Cloud Computing," in Proc. European Symposium on Research in Computer Security. Springer-Verlag, 2009, pp. 355–370.

[20] Y. Zhu, H. Wang, Ahn, H. Hu, and S. S. Yau, "Dynamic Audit Services for the Integrity Verification of an Outsourced Storage in the Clouds," ACM Symposium On Applied *Computing*, 2011, pp. 1550–1557.