INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN 2454-9924 Volume: 1 Issue: 1(2015)

User Activity Monitoring for Dynamic and Flexible Group Key Generation

R. Shanmuga Sundaram #1, T. Priya Rathika Devi *1

Mailam Engineering College, Mailam #1, *1

Shanmugasundaram2008@gmail.com

Abstract -In cloud user data can be easily shared and viewed across the shared medium. To make sure the shared data can be verified publicly, users in the group need to calculate signatures on all the blocks in shared data. Various blocks in shared data are generally determined by different users due to data modifications indicated by different users. In the previous methods, there is no Security in the Cloud is enforced. In the proposed method**,** Data Owner updates the information to the Remote Cloud Server for Data Access. Data owner appoints Members for Data Utility and Data updating. Members have to get permission for the Data updating from the Data Owner. Members will have their User Name, Key, and Group Key for Access. If Existing member is removed from that Group, Group Key is automatically changed and updated to all the Members of that Group. The modified work is Group Key can be changed in case of New Member is added in that Group or Existing Member is Resigned by themselves from the Group or Data Owner Terminates the Member or Cloud Terminates the Member in case of Misbehavior (DDOS Attack, Same Data Download), updated new key is sent to the corresponding users through Email.

Keywords: Data owner, Signatures, Cloud server, Permission, Group key

1Introduction

With data storage and sharing services (such as Drop box and Google Drive) provided by the cloud, people can easily work together as a group by sharing data with each other. More specifically, once a user creates shared data in the cloud, every user in the group is able to not only access and modify shared data, but also share the latest version of the shared data with the rest of the group. Although cloud providers promise a more secure and reliable environment to the users, the integrity of data in the cloud may still be compromised, due to the existence of hardware/ software failures and human errors. To protect the integrity of data in the cloud, a number of mechanisms have been proposed. In these mechanisms, a signature is attached to each block in data, and the integrity of data relies on the correctness of all the signatures. One of the most significant and common features of these mechanisms is to allow a public verifier to efficiently check data integrity in the cloud without downloading the entire data, referred to as public auditing (or denoted as Provable Data Possession). This public verifier could be a client who would like to utilize cloud data for particular purposes (e.g., search, computation, data mining, etc.) or a third-party auditor (TPA) who is able to provide verification services on data integrity to users. Most of the previous works focus on auditing the integrity of personal data. Different from these works, several recent works focus on how to preserve identity privacy from public verifiers when auditing the integrity of shared data. Unfortunately, none of the above mechanisms considers the efficiency of user revocation when auditing the correctness of shared data in the cloud. With shared data, once a user modifies a block, she also needs to compute a new signature for the modified block. Due to the modifications from different users, different blocks are signed by different users. For security reasons, when a user leaves the group or misbehaves, this user must be revoked from the group. As a result, this revoked user should no longer be able to access and modify shared data, and the signatures generated by this revoked user are no longer valid to the group. Therefore, although the content of shared data is not changed during user revocation, the blocks, which were previously signed by the revoked user, still need to be re-signed by an existing user in the group. As a result, the integrity of the entire data can still be verified with the public keys of existing users only. Since shared data is outsourced to the cloud and users no longer store it on local devices, a straightforward method to re-compute these signatures during user revocation is to ask an existing user (i.e., Alice) to first download the blocks previously signed by the revoked user (i.e., Bob), verify the correctness of these blocks.

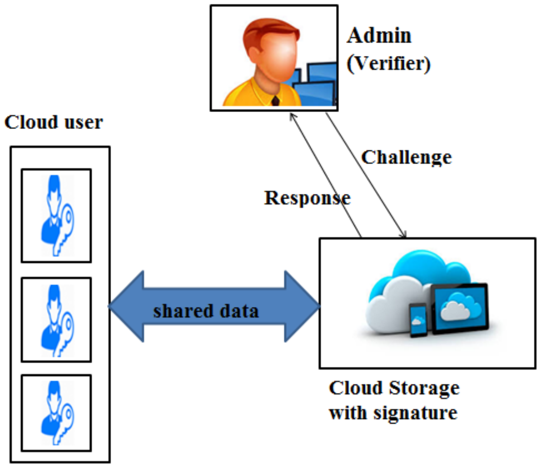
2 Related works

In the existing system, there is no Security in the Cloud is enforced. Here it provides some drawbacks they are; provide less security and poor data integrity and confidentiality

2.1 Proposed work

Data Owner updates the information to the Remote Cloud Server for Data Access. Data owner appoints Members of Data Utility and Data updating. Members have to get permission for the Data updating from the Data Owner. Members will have their User Name, Key, and Group Key for Access. Either If Existing member is removed from that Group, Group Key is automatically changed and updated to all the Members of that Group. Group Key can be changed in case of New Member is added in that Group also. Member can resign from the Group by themselves or Data Owner can terminate the Member or can be Cloud Terminates the Member in case of Misbehavior (DDOS Attack, Same Data Download). Finally the proposed system provides less efficiency and high performance.

3 System Model



We assume the cloud itself is semi- trusted, which means it follows protocols and does not pollute data integrity actively as a malicious adversary, but it may lie to verifiers about the incorrectness of shared data in order to save the reputation of its data services and avoid losing money on its data services. In addition, we also assume there is no collusion between the cloud and any user during the design of our mechanism. Generally, the incorrectness of share data under the above semi-trusted model can be introduced by hardware/software failures or human errors happened in the cloud. Considering these factors, users do not fully trust the cloud with the integrity of shared data. To protect the integrity of shared data, each block in shared data is attached with a signature, which is computed by one of the users in the group. Specifically, when shared data is initially created by the original user in the cloud, all the signatures on shared data are computed by the original user. After that, once a user modifies a block, this user also needs to sign the modified block with his/her own private key. By sharing data among a group of users, different blocks may be signed by different users due to modifications from different users. When a user in the group leaves or misbehaves, the group needs to revoke this user. Generally, as the creator of shared data, the original user acts as the group manager and is able to revoke users on behalf of the group. Once a user is revoked, the signatures computed by this revoked User become invalid to the group, and the blocks that were previously signed by this revoked user should be re-signed by an existing user's private key, so that the correctness of the entire data can still be verified with the public keys of existing users only.

4 Methodologies

Network Construction

This module is developed in order to create a dynamic network. In a network, nodes are interconnected with a particular group and the information can be shared among them. For the successful data transfer the network must be properly controlled and handled. Every node is interconnected & this forms a network.

4.1 Server in Cloud

Here the server will have the entire details about all the group information. It distributes the data to client in a particular group. Server is responsible for maintaining all the group information. If any user will removed from a particular group means it will instruct to the group member to change the group id and send the SMS to all group members.

4.2 Cloud User Status

Users can be moved from one group to another group and he will also participated in more than one group. Depending upon the user status they can share their information with the group member. All the user status information is to be maintained here. If a new user login or the existing user logged in or logged out all that information about a user must maintained for authenticate.

4.3 Generation of Group Key

In this module we have to create group key as well as the individual key then share the key between the group members via SMS.

Any changes occurs in the group then change the group key and then send that key to the other entire group member. This process is done whenever any changes made in that group.

4.4 Cloud Data Access

If a user wants to access any information about any user then he will give his individual key as well as the group key. If he want to access the information about the user, but the user is not belongs to their group is not possible. He can only access the user's information within their group only. Without knowledge of the other group key it is not possible to access the information.

5 PANDA

Based on the new proxy re-signature scheme and its properties in the previous section, we now present Panda—a public auditing mechanism for shared data with efficient user revocation. In our mechanism, the original user acts as the group manager, who is able to revoke users from the group when it is necessary. Meanwhile, we allow the cloud to perform as the semi-trusted proxy and translate signatures for users in the group with re-signing keys. As emphasized in recent work, for security reasons, it is necessary for the cloud service providers to storage data and keys separately on different servers inside the cloud in practice. Therefore, in our mechanism, we assume the cloud has a server to store shared data, and has another server to manage re-signing keys. To ensure the privacy of cloud shared data at the same time, additional mechanisms, such as, can be utilized. The details of preserving data privacy are out of scope of this paper. The main focus of this paper is to audit the integrity of cloud shared data.

6 Literature Review

[2] Describes about, with data services in the cloud, users can easily modify and share data as a group. To ensure data integrity can be audited publicly, users need to compute signatures on all the blocks in shared data. Different blocks are signed by different users due to data modifications performed by different users. For security reasons, once a user is revoked from the group, the blocks, which were previously signed by this revoked user, must be re-signed by an existing user. The straightforward method, which allows an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient

due to the large size of shared data in the cloud. In this paper, we propose a novel public auditing mechanism for the integrity of shared data with efficient user revocation in mind. By utilizing proxy re-signatures, we allow the cloud to re-sign blocks on behalf of existing users during user revocation, so that existing users do not need to download and re-sign blocks by themselves. In addition, a public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud, even if some part of shared data has been re-signed by the cloud. Experimental results show that our mechanism can significantly improve the efficiency of user revocation.

[14] Describes about**,** in a proof-of-irretrievability system, a data storage center must prove to a varied that he is actually storing all of a client's data. The central challenge is to build systems that are both efficient and provably secure that is, it should be possible to extract the client's data from any prove that passes a verification check. In this paper, we give the first proof-of-irretrievability schemes with full proofs of security against arbitrary adversaries in the strongest model, that of Juels and Kaliski. Our scheme, built from BLS signatures and secure in the random oracle model, has the shortest query and response of any proof-of-irretrievability with public variability. Our second scheme, which builds elegantly on pseudorandom functions (PRFs) and is secure in the standard model, has the shortest response of any proof-of-irretrievability scheme with private variability (but a longer query). Both schemes rely on holomorphic properties to aggregate a proof into one small authenticator value. [10] Describes about, Cloud Computing has been envisioned as the next generation architecture of IT Enterprise. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, Cloud Computing moves the application software and databases to the large data centers, where the management of the data and services may not be fully trustworthy. This unique attribute, however, poses many new security challenges which have not been well understood. In this article, we focus on cloud data storage security, which has always been an important aspect of quality of service. To ensure the correctness of users' data in the cloud, we propose an effective and flexible distributed scheme with two salient features, opposing to its predecessors. By utilizing the

homomorphism token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s). Unlike most prior works, the new scheme further supports secure and efficient dynamic operations on data blocks, including: data update, delete and append. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks. [11] describes about, We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems. We present two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

7 Conclusion

From this, User Activity Monitoring for Dynamic and Flexible Group Key Generation has been implemented. We have proposed a new public auditing mechanism for shared data with efficient user revocation in the cloud. When a user in the group is revoked, we allow the semi-trusted cloud to re-sign blocks that were signed by the revoked user with proxy re-signatures. Experimental results show that the cloud can improve the efficiency of user revocation, and existing users in the group can save a significant amount of computation and communication resources during user revocation. In future, our system will enhance the efficiency also provides security in cloud.

8 References

[1] A. Shamir, “How to Share a Secret,” Comm. ACM, vol. 22, no. 11, pp. 612-613, Nov. 1979.

[2] Boyang Wang,Public Auditing for Shared Data with Efficient User Revocation in the Cloud

[3] 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.

[3] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS'13), pp. 124-133, July 2013.

[4] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE CLOUD, pp. 295-302, 2012.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th ACM/IEEE Int'l Workshop Quality of Service (IWQoS'09), pp. 1-9, 2009.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Jan. 2012.

[7] B. Wang, H. Li, and M. Li, "Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics," Proc. IEEE Int'l Conf. Comm. (ICC'13), pp. 1946-1950, June 2013.

[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] 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.

[10] Cong Wang, Ensuring Data Storage Security in Cloud Computing

[11]Giuseppe Ateniese, Provable Data Possession at Untrusted Stores

[12] 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.

[13] G. Ateniese and S. Hohenberger, "Proxy Re-signatures: New Definitions, Algorithms and Applications," Proc. 12th ACM Conf. Computer and Comm. Security (CCS'05), pp. 310-319, 2005.

[14]Hovav Shacham , Compact Proofs

[15] 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. 90107, 2008.

[16] H. Wang, "Proxy Provable Data Possession in Public Clouds," IEEE Trans. Services Computing, vol. 6, no. 4, pp. 551-559, Oct.Dec. 2013.

[13] H. Wang, "Proxy Provable Data Possession in Public Clouds," IEEE Trans. Services Computing, vol. 6, no. 4, pp. 551-559, Oct.Dec. 2013.

[17] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Kon-winski, 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.

[18] M. Blaze, G. Bleumer, and M. Strauss, “Divertible Protocols and Atomic Proxy Cryptography,” Proc. Int’l Conf. the Theory and Application of Cryptographic Techniques (EUROCRYPT’98), pp. 127-144, 1998.

[19] N. Cao, S. Yu, Z. Yang, W. Lou, and Y.T. Hou, “LT Codes-Based Secure and Reliable Cloud Storage Service,” Proc. IEEE INFOCOM, pp. 693-701, 2012.

[20] M. Li, N. Cao, S. Yu, and W. Lou, “FindU: Private-Preserving Personal Profile Matching in Mobile Social Networks,” Proc. IEEE INFOCOM, pp. 2435-2443, 2011.