International Journal of Current Trends in Engineering & Technology
ISSN: 2395-3152
Volume: 03, Issue: 06 (NOVEMBER - DECEMBER, 2017)

Secure Auditing On Cloud Data with Outsourcing of Secret Keys

R. G. Suresh Kumar[1], Prof. Dr. T. Nalini2, Dhanam. M [3]
Department Of Computer Science and Engineering
[1]*Vels, University, India*
[2]Bharath University,Chennai,Tamilnadu,India
[3]*Rajiv Gandhi College of Engineering & Technology, Tamilnadu, India*

Abstract:— *Cloud computing, the fastest growing technology, performs auditing with the help of the TPA (Third Party Authorizer).They can only provide the encrypted key whenever the client needs to upload file in the cloud which provides auditing. This can reduce the work load from client side. Security of data in the file is not preserved well. In the proposed work, along with encrypted key for auditing the user login BLS signature is used to verify the integrity of data. Virtual Machines are used for Auditing purpose. This will ensure the integrity while using cloud data. The privacy of user's data is also preserved from virtual machines because while verifying integrity virtual machine does not have any knowledge about user's data. Compared with single user auditing, the performance will be better in batch auditing. Virtual Machines can perform batch auditing here. And divide and conquer strategy is used to detect corrupt blocks. Multithreading is used to increase performance.*

Keywords:—*Data storage, cloud storage auditing, Data Integrity, BLS Signature, cloud computation, Privacy-Preserving.*

I. INTRODUCTION

CLOUD storage auditing is used to verify the data integrity stored in public cloud, which is one of the important security techniques in cloud storage. In sharing of data in cloud storage, auditing protocols have fascinated much attention and have been researched intensively [1–16]. These protocols focus on several different aspects of auditing, and how to achieve high bandwidth and computation efficiency is one of the essential concerns. For that purpose, the Boney Lynn Shacham (BLS) signature technique that supports block less verification is explored to reduce the overheads of computation and communication in auditing protocols, which allows the auditor to verify the integrity of the data in cloud without retrieving the whole data.

In today technology outsourcing computation has attracted much by the business. It has been considered in many applications including scientific computations [1], linear algebraic computations [2], linear programming computations [3] and modular exponentiation computations [4], etc. Cloud computing can also offers users the unlimited storage resource which is considered as one of the most important services of cloud computing. Although cloud storage provides great benefit to users, it brings new security challenging problems.

One important security problem is how to efficiently check the integrity of the data stored in cloud. In recent years, many auditing protocols for cloud storage have been proposed to deal with this problem. To overcome this problem many auditing protocols have been proposed. These protocols focus on different aspects of cloud storage auditing such as the high efficiency [5]–[17], the privacy protection of data [18], the privacy protection of identities [19],dynamic data operations [13], [15], [16], [20], the data sharing [21], [22], etc. Lately, another important problem arises known to be key exposure in cloud storage auditing.

The problem itself is non-trivial by nature. Once the client's secret key for storage auditing is exposed to cloud, the cloud is able to easily hide the data loss incidents for maintaining its reputation, even discard the client's data rarely accessed for saving the storage space. Yu *et al.* [23] constructed a cloud storage auditing protocol with key-exposure resilience by updating the user's secret keys periodically. In this way, the damage of key exposure in cloud storage auditing can be reduced. But it also brings in new local burdens for the client because the client has to execute the key update algorithm in each time period to make his secret key move forward. For some clients with limited computation resources, they might not like doing such extra computations by themselves in each time period. It would be obviously more attractive to make key updates as transparent as possible for the client, especially in frequent key update scenarios. In this paper, we consider achieving this goal by outsourcing key updates as well as verify data integrity.

To achieve this goal, Cloud computing has to fulfil some requisites. They are :(1) the virtual machine does not know about the client's secret keys for cloud storage auditing who performs outsourcing computation for key updates. If not, there arise security threats. Hence to avoid this only the encrypted key should be known to the virtual machine. (2) As the virtual machine knows the encrypted keys they should perform key updates only under the encrypted version. (3)It should be very efficient for the client to recover the real secret key from the encrypted version that is retrieved from the virtual machine, and then the client should be able to verify the validity of the encrypted secret key after the client retrieves it from the authorized party. The encrypted files are stored in the cloud and the keys of the encrypted files are stored in the Virtual Machine2.

Finally, to verify data integrity the comparison of keys stored in cloud and the one hold by the VM2 is performed. If both keys are matched, then data integrity can be achieved. The goal of this paper is to design a cloud storage auditing protocol that can satisfy above requirements to achieve the outsourcing of key updates and data integrity.

The main contributions are as follows: (1) we proposed a model called cloud storage auditing with verifiable outsourcing of key updates and checking data integrity. In this new paradigm, keys are updated by a virtual machine, which holds an encrypted secret key of the client for cloud storage auditing and updates it under the encrypted state in each time period. If the client wants to upload files to cloud, he can do by decrypting the encrypted secret key. Also, the client can verify the validity of the encrypted secret key. (2) This is the first cloud storage auditing protocol with verifiable outsourcing of key updates. In our design, the Virtual Machines play the role of the authorized party who not only updates the key but also checks the integrity of the files in cloud. The CSP knows only an encrypted version of the data. In the detailed, the files are divided into blocks and calculate aggregate authenticator for each block. Encrypted files are stored in the cloud and corresponding encrypted keys are stored in the VM2. By matching both the keys, VM can verify its data integrity. (3) We formalize the definition and the security model of the cloud storage auditing protocol with verifiable outsourcing of key updates as well as data integrity and checking the correctness of the data. We also prove the security of our protocol in the formalized model and its performance by firm implementation.

II. RELATED WORK

How to effectively outsource Time-consuming computations has become a hot topic in the research of the theoretical computer science in the recent two decades. Outsourcing computation has been considered in many application domains. Chaum and Pedersen [24] firstly proposed the notion of wallet databases with observers, in which hardware was used to help the client perform some expensive computations. The method for secure outsourcing of some scientific computations was proposed by Atallah *et al*. [1]. Chevallier-Mames *et al*. [25] designed the first effective algorithm for secure delegation of elliptic curve pairings based on an untrusted server. The first outsourcing algorithm for modular exponentiations was proposed by Hohenberger and Lysyanskaya [26], which was based on the methods of precomputation and server-aided computation. Atallah and Li [27] proposed a secure outsourcing algorithm to complete sequence comparisons. Chen *et al*. [4] proposed new algorithms for secure outsourcing of modular exponentiations. Benjamin and Atallah [2] researched on how to securely outsource the computation for linear algebra.

Atallah and Frikken [28] gave further improvement based on the weak secret hiding assumption. Wang *et al*. [3] presented an efficient method for secure outsourcing of linear programming computation. Chen *et al*. [29] proposed an outsourcing algorithm for attribute-based signatures computations. Zhang *et al*. [30] proposed an efficient method for outsourcing a class of homomorphic functions.

Cloud Storage Auditing: How to check the integrity of the data stored in cloud is a hot topic in cloud security. The notion of "provable data possession" (PDP) was firstly proposed by

Ateniese *et al*. [5] to ensure data possession at untrusted servers. The notion of "proof of retrievability" (PoR) was proposed by Juels *et al.* [6] to ensure both possession and retrievability of data at untrusted servers. Wang *et al*. [18] proposed a public privacy-preserving auditing protocol.

They used the random masking technique to make the protocol achieve privacy preserving property. Proxy provable data possession protocol was proposed in [17]. The auditing protocols supporting dynamic data operations were also proposed in [13] and [20]. Yang and Jia [16] proposed an auditing protocol supporting both the dynamic property and the privacy preserving property.

The privacy preserving of the user's identity for shared data auditing was considered in [19]. The problem of user revocation in shared data auditing was considered in [21]. Yuan and Yu [22] proposed a public auditing protocol for data sharing with multiuser modification. Sookhak *et al*. [31] proposed a public cloud auditing protocol for securing big data storage based on algebraic signature. Guan *et al*. [32] proposed the first cloud storage auditing protocol based on indistinguishability obfuscation, which is especially useful for low-power cloud users. Yang *et al*. [33] proposed a public auditing protocol for shared cloud data supporting both identity privacy and identity traceability.

All above auditing protocols are all built on the assumption that the secret key of the client is absolutely secure and would not be exposed.

In [23], the authors firstly considered the key exposure problem in cloud storage auditing and proposed a cloud storage auditing protocol with key-exposure resilience.

In that protocol, the secret keys for cloud storage auditing are updated periodically. As a result, any modifications in cloud's data, such as deleting or modifying the client's data previously stored in cloud, can all be detected, even if the cloud gets the client's current secret key for cloud storage auditing.

The disadvantage of such system was the client needs to update his secret key in each time period. When keys have to be updated frequently this will become a serious burden for client.

Provable Data Possession Model (Ateniese et al.)[5] At et al used RSA based homomorphic tags in his scheme Problems related to security. In this public auditability is achieved but disadvantage of this model is data exposed to external auditor.

Proof of Retrievability (Jules and Kaliski) [6] It uses Spot checking and error checking codes are used Number of audit challenges a user can perform is fixed priori and public auditability is not supported in this model.

Improved POR scheme (Shacham and Waters) [7] in this, Homomorphic authenticators are used. Public auditability is achieved but could not prevent the leakage of data blocks in the verification to external auditors.

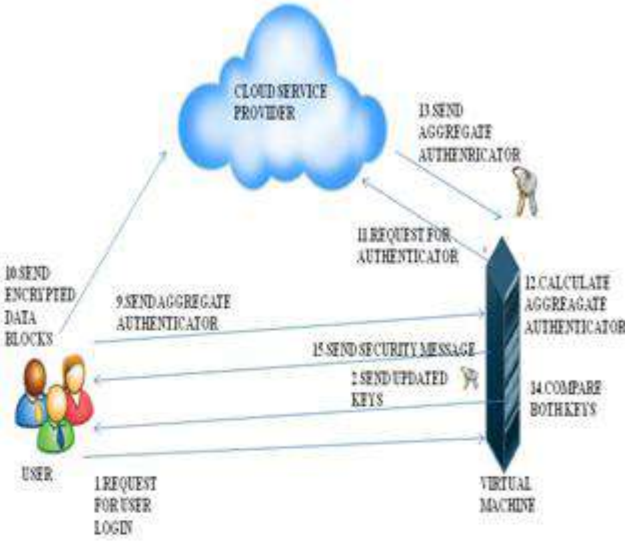
Privacy Preserving Audit (Shah et al.)[8] This model uses symmetric-keyed hash Brings online burden to users.

Dynamic PDP (Ateniese et al.)[9] This model uses symmetric key cryptography Bounded with number of audits.

III. OUR PROPOSED PROTOCOL

A. A. Architecture

Fig 1 shows architecture for cloud to check integrity of data. There are three main actors - User, CSP and VM.


CLOUD SERVICE PROVIDER
13.SEND AGGREGATE AUTHENRICATOR
11.REQUEST FOR AUTHENTICATOR
10.SEND ENCRYPTED DATA BLOCKS
9.SEND AGGREGATE AUTHENTICATOR
12.CALCULATE AGGREAGATE AUTHENTICATOR
15.SEND SECURITY MESSAGE
2.SEND UPDATED KEYS
14.COMPARE BOTH KEYS
USER
1.REQUEST FOR USER LOGIN
VIRTUAL MACHINE

3. User Generates Private and Public keys
4. User Divides File into blocks
5. Encrypt each file using secret keys
6. Calculate hash on each block using hmac MD5 and public key
7. Calculate encrypt hash using private key
8. Calculate aggregate authenticator

Fig 1 Architecture of Integrity Verification of the Cloud Data

Figure 1 shows Integrity verification architecture of cloud BLS signature is used for integrity verification purpose. There are 3 phases of BLS signature – key generation, signing and verification. [10] Key generation is the process of generating public and private keys. Signing means generation of proof. Cloud service provider is responsible for generating proof. Verification is the phase in which proof is verified. Verification phase is performed by third party auditor. There are following steps for integrity verification -

Key Generation - User generates private key x and public Key y using pseudo random function.

Signing - User encrypts the file blocks using AES with secret key. Then user divides file into blocks where each block has size 8kb. Calculate hash on each block using HMAC MD5 by using user's public key and finally calculate signature that is encrypt hash using user's private key which is called as authenticator. AES symmetric encryption algorithm is used for encrypting data blocks using secret key. Key size used for encryption is 128 bit. To calculate hash hmac MD5 is used. Hmac MD5 works on 512 bit blocks and it produces 128 bits of hash. User stores encrypted blocks on cloud and sends authenticator to VM.

VM sends Challenge to CSP – VM selects random blocks of file and for that block sends challenge to CSP to verify its integrity. CSP calculates new aggregate authenticator for random blocks. To calculate aggregate authenticator CSP uses encrypted file blocks. I.e. decryption doesn't take place at CSP side. So, more security is provided compared to earlier systems. CSP sends aggregate authenticator to VM.

Verification – Virtual Machine compares two authenticators one which is received from CSP and another which is received from user. Finally, the VM sends security message to user to indicate whether file is corrupted or not. In this, external auditor called Virtual Machines is used for integrity verification purpose. As VM is used for integrity verification, system achieves public auditability. CSP and user do not send data to VM. Our scheme enables an external auditor to audit user's outsourced data in the cloud without learning the data content data i.e. Virtual Machines does not have any knowledge about data. So, security is improved. Hence our scheme achieves privacy preservation as well as block less verification.

B. Batch Auditing

VM can handle multiple delegations simultaneously. Individual auditing is tedious and inefficient. For given different K delegations on K distinct files from K different users it is advantageous for VM to batch these multiple tasks together and audit at one time. Calculation of aggregation of all authenticators is useful. Suppose, there are K users in the system and each user k has data file F where Fk = m {k, 1} . . . m {k, n} to be outsourced to cloud server, where k ∈ {1. . . K}. for simplicity, we assume that each file Fk has same number of n blocks. Each user k runs signature generation and Computes authenticator σ for each block. Finally each user sends authenticator to VM and data file to CSP. To generate chal message VM selects random blocks of file F and forms batch of auditing requests from different users and sends it to CSP. CSP generates aggregate authenticator for request and send proof to VM, finally the VM verifies it.

C. Identification of Invalid Responses

Received authenticators are equal only when all the responses are valid and fail with high probability even if there is single invalid response. In many situations, response collection may contain invalid responses caused by malicious or data corruption. Ratio of invalid responses to valid could be quite small yet standard auditor rejects entire collection. To further sort out these responses in batch auditing we used divide and conquer approach. Simply, divide blocks into two halves and generate aggregate authenticator for both. This procedure repeats until we get corrupt block.

D. Identification of Invalid Responses using Multithreading

For identifying corrupt blocks we used multithreading. Numbers of blocks are divided in two halves. Then authenticator is calculated for both halves. For verifying these authenticators two threads are applied. Consider if authenticators of second half are not equal, then second half is again divided in two parts and again calculate authenticators and apply two threads which were applied previously. This procedure is repeated until we find corrupt block. In this we only require two threads for verification purpose.

E. Proposed Mathematics

TABLE I

SYMBOLS	*MEANING*
CSP	Cloud Service Provider
F_i	Set of files; i∈ {1 , 2, . . . , n}
F	File is divided in blocks mi; i∈ {1 , 2, . . . , n}
Mi	ith block
Σ	Signature
Hi	Hash on block

User
1. User divides file into blocks.
f = {m1, m2. . . mn}
2. Calculate hash on encrypted block
Hash (mi) → hi
3. Calculate digital signature
SigGen (mi) → σi
4. Calculate aggregate authenticator.
Aggre_auth (σi) → σ
⍰⍰VM
1. Send file to check its integrity (fi).
2. Verify Signature σ = σ’.
⍰⍰CSP
1. Calculate aggregate authenticator.
SigGen (mi) → σi'
Aggre_auth (σi') → σ’

IV. CONCLUSION

Virtual Machines are used to verify integrity of user’s data and privacy is also preserved as VM doesn’t have any knowledge about user’s data. Time is saved as VM is used for auditing purpose. Block less verification is achieved as data is not retrieved to check correctness of data. By using Virtual Machines for outsourcing of keys and data integrity, achieved public auditability and privacy of data is also preserved. Batch auditing performance is better than single user auditing as it requires less time than single user auditing. For divide and conquer multi-threading is used to find the corrupt block. When multithreading is used performance will be better and also it can be done quickly.

IV. FUTURE SCOPE

In the proposed work, we are using one CSP for storing user's data. In future we can extend our system by using multiple CSP for storing purpose. For example, if file is divided in three blocks, then these three blocks are stored on three different CSPs and integrity is verified of these three file blocks. As, shown above, on CSP-I block one, on CSP-II block two and on CSP-III block three is stored. In auditing VM retrieves blocks from different CSP's and performs auditing.

REFERENCES

[1]. Sonali Thosar, N. A. Mhetre, “Privacy Preserving Integrity Verification for Securing Cloud Storage”, International Journal of Engineering Research and Technology(IJERT), ISSN : 2278-0181, vol 3,Issue 1, JAN 2014.

[2]. Sonali Thosar, N. A. Mhetre, “An Integrated Approach to Public Audit ability for Secure Cloud Storage”, Third Post Graduate Symposium of Computer Engineering, cPGCON March 2014.

[3]. Cong Wang, Shermann S-M Chow, Qian Wang, Kui Ren and Wenjing Lou, “Privacy Preserving Public Auditability for Secure Cloud Storage”, IEEE Transactions on Cloud Computing, Year 2013.

[4]. K. Shirisha Reddy, Dr. M. Balaraju, “An Integrated Approach of Data Storage and Security in Cloud Computing”, International Journal of Application or Innovation in Engineering and Management, Volume 1, Issue 4, December 2012.

[5]. G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J. Peterson, and D.X. Song, “Provable Data Possession at Entrusted Stores”, Proc. 14th ACM Conf. Computer and Comm. Security (CCS’07), pp. 598-609, 2007.

[6]. A. Juels and B.S.K. Jr., “Pors: Proofs of Retrievability for Large Files”, Proc. 14th ACM Conf. Computer and Comm. Security (CCS ’07), pp. 584-597, 2007.

[7]. 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, and 2008.

[8]. Shuai Han, Jianchan Xing, “Ensuring Data Storage Security through Novel Third Party Auditor Scheme in Cloud Computing”, CCIS, proceedings of IEEE,2011.

[9]. Qian Wang, Cong Wang, Kui Ren, Wenjing Lou and Jin Li, “Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing”, IEEE Transactions on Parallel and Distributed Systems Distributed Systems, VOL. 22, NO. 5, MAY 2011.

[10]. D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the Weil Pairing”, J. Cryptology, vol. 17, no. 4, pp. 297–319, 2004.