International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 06 | June -2017 www.irjet.net p-ISSN: 2395-0072

Auditing and Resisting Key Exposure on Cloud Storage

Akshata M. Bhand, D. A. Meshram

Student, ME (IT) , RMD Sinhgad School of Engineering,Pune,
Assistant Professor, ME (IT), RMD Sinhgad School of Engineering, Pune

---

Abstract— *In the cloud computing auditing is an important service to maintain the integrity. Existing examining conventions are altogether in light of the supposition that the Client's mystery key for examining is totally secured. Such supposition may not generally be held, due to the likely frail suspicion that all is well and good and additionally low security settings at the customer. In a large portion of the current evaluating conventions would unavoidably get to be distinctly not able to work when a mystery key for evaluating is uncovered. It is explored on the best way to decrease the harm of the customer's key disclosure in distributed storage evaluating, and give the main helpful illustration to this new issue setting. Formalized the definition and the security model of inspecting convention with key-presentation strength and propose such a convention. Used and built up a novel authenticator development to bolster the forward security and the property of piece less undeniable nature utilizing the current plan. The security verification and the execution investigation appear that the anticipated convention is secured and efficient.*

Key words — Data storage, cloud storage auditing, homomorphic linear authenticator, cloud computation, keyexposure resistance.

1 INTRODUCTION

Cloud storage auditing is used to verify the integrity of the data stored in public cloud, which is one of the important security techniques in cloud storage. In recent years, auditing protocols for cloud storage have attracted much attention and have been researched intensively [1]. These protocols focus on several different aspects of auditing, and how to achieve high bandwidth and computation efficiency is one of the essential concerns [3]. For that purpose, the Homomorphism Linear Authenticator (HLA) 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.

Many cloud storage auditing protocols like have been proposed based on this technique [1]-[8]. The privacy protection of data is also an important aspect of cloud storage auditing. In order to reduce the computational burden of the client, a third-party auditor (TPA) is introduced to help the client to periodically check the integrity of the data in cloud. However, it is possible for the TPA to get the client‟s data after it executes the auditing protocol multiple times [3]. Auditing protocols are designed to ensure the privacy of the client‟s data in cloud. Another aspect having been addressed in cloud storage auditing is how to support data dynamic operations [9].

Key exposure could happen due to several reasons:

1) Key management- Key management is a process which is done by the client. In case any fault occurs and if the client is using a cheap software-based key management, then key exposure is possible.

2) Internet based security attacks- Suppose if a client downloads any data or file and if that it contains malicious program, then it may infect the system. This allows the hackers to easily access any confidential data [4].

3) Trading with hackers- It can happen that cloud also earns incentives by trading with the concerned hackers. In this process, the cloud can get the client's data and forge the authenticator by regenerating false data or by hiding data loss. Thus, dealing with key exposure is a vital issue in cloud storage and various methodologies were adopted

2 RELATED WORKS

1. Enabling Cloud Storage Auditing With Key-Exposure Resistance

Authors- Jia Yu, Kui Ren, Cong Wang and Vijay Varadharajan
In this paper manage the customer's key introduction in distributed storage examining. Creator propose another worldview called evaluating convention with key-introduction strength. In such a convention, the uprightness of the information beforehand put away in cloud can at present be confirmed regardless of the possibility that the customer's present mystery key for distributed storage evaluating is uncovered. Formalize the definition and the security model of evaluating convention with key-presentation versatility, and afterward propose the primary handy arrangement. The security confirmation and the asymptotic execution assessment demonstrate that the proposed convention is secure and proficient [1].

2.“ Enhancing Data Security In Cloud Storage Auditing With Key Abstraction”

Authors- Priyadharshni, and Geo Jenefer. G
in this paper two essential answers for the key-presentation issue of distributed storage evaluating is talked about and actualized. The first is an innocent arrangement, which in truth can't in a general sense take care of this issue. The second is a marginally better arrangement, which can tackle this issue however has a substantial overhead. They are both unfeasible when connected in practical settings. And after that center convention that is a great deal more productive than both of the essential arrangements [2] .

3.“ An Efficient Cloud Storage Batch Auditing Without Key Exposure Resistance Using Public Verifier”

Authors- T Yawaikha, R Meyanand
Paper presents think about on the most proficient method to manage the customer's key without uncovering into the cloud. The evaluating performed by open verifier reviews the information as well as checks the honesty of the information in cloud. The idea of client repudiation permits to renounce the invalid key enlisted. Formalize the definition and the security model of reviewing convention without key-introduction versatility, and after that propose and confirm the principal down to earth arrangement [3]

4.“ Survey Paper On Cloud Storage Auditing With Exposure Resistance”

Authors- Sneha Singha, S. D. Satav
As this total paper portrays the diverse approachs on empowering distributed storage evaluating with key presentation strength, yet none of the systems is by all accounts idealize. Along these lines, this study paper as a bit proposes a technique for a viable key presentation resistance where we embrace the deduplication system of information. Besides, it will check the duplicacy of information and dispense with the excess one utilizing MD5 hashing calculation. After people in general and private keys are created, it utilizes tile bitmap technique wherein it will check the past and the present adaptations of the information to facilitate the inspector's workload and to make the framework more effective [4]

5.“ Efficient provable data possession for hybrid clouds”

Authors- Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau
This paper tended to the development of PDP plan for cross breed mists. In light of homomorphic undeniable reactions and hash record chain of command, Author proposed an agreeable PDP plan to bolster dynamic versatility on different stockpiling servers. Tests demonstrated that our plans require a little, steady measure of overhead [5] .

3 PROPOSED SCHEME

At a high level, our setting of interest is an enterprise network, consisting of a group of affiliated clients (for example, employees of a company) who will use the SCSP and store data with deduplication technique. In deduplication can be frequently used in these settings for data backup and disaster recovery applications while greatly reducing storage space. Such systems are widespread and are often more suitable, in to user file backup and synchronization applications than richer storage abstractions [14]. There are three entities defined in our system, that is, users, private cloud and S-CSP in public cloud. The S-CSP performs deduplication by checking if the contents of two files are the same and stores only one of them. The access right to a file is defined based on a set of privileges. Cloud data storage service includes the user(U), who has the large data to be stored in cloud; the cloud server(CS), managed by cloud service provider(CSP) with significant storage; the third party auditor(TPA), trusted to access the CSP according to users request. When user store the data, the copy is sent to both the CSP and TPA. To verify the correctness of data stored in cloud, auditing process is done.

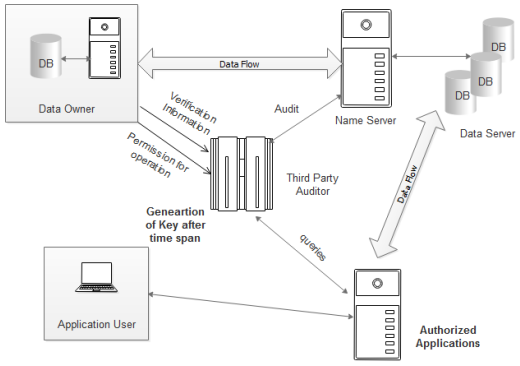

Fig -1: System Architecture

Here the auditing process is carried out TPA, it must efficiently audit without bringing any changes to the original data. For auditing, the data which is in TPA is used. Public auditability: Allow the TPA to verify the correctness of data without demanding the copy of data. Privacy preserving: To ensure that TPA cannot retrieve the data content during the auditing process. Lightweight: To allow TPA to perform auditing with minimum communication and computation overhead.

5. ALGORITHM USED

An auditing protocol with key-exposure resilience is composed by five algorithms (SysSetup,
KeyUpdate, AuthGen, ProofGen, ProofVerify), shown below:

5.1. SysSetup(1k, T) →(PK, SK0):

The system setup algorithm is a probabilistic algorithm which takes as input a security parameter *k* and the total number of time periods *T*, and generates a public key *PK* and the initial client‟s secret key *SK*0. This algorithm is run by the client.

5.2. KeyUpdate(PK, j, SK j) → (SK j+1):

The key update algorithm is a probabilistic algorithm which takes as input the public key *PK*, the current period *j* and a client‟s secret key *SK j* , and generates a new secret key *SK j*+1 for the next period *j* + 1. This algorithm is run by the client.

5.3. AuthGen(PK, j, SK j , F) → (_):

The authenticator generation algorithm is a probabilistic algorithm which takes as input the public key *PK*, the current period *j* , a client‟s secret key *SK j* and a file *F*, and generates the set of authenticators _ for *F* in time period *j*. This algorithm is also run by the client.

5.4. Proof Gen(PK, j,Chal, F,_) → (P):

The proof generation algorithm is a probabilistic algorithm which takes as input the public key *PK*, a time period *j* , a challenge *Chal*, a file *F* and the set of authenticators _, and generates a proof *P* which means the cloud possesses *F*. Here, *(j,Chal)* pair is issued by the auditor, and then used by the cloud. This algorithm is run by the cloud.

5.5.Proof Veri f y(PK, j,Chal, P) → (“True” or “False”):

The proof verifying algorithm is a deterministic algorithm which takes as input the public key *PK*, a time period *j* , a challenge *Chal* and a proof *P*, and returns “*True*” or “*False*”. Thisalgorithm is run by the client.

6. MATHEMATICAL MODEL

S is the system
S={I, O, F , K,T, Success, Failure }

Where,
I = Set of Input
I={I1, I2, I3}
Where,

I1=Login user ID
I2=Login password
I3=File

K=Key set of Secret key and Public key
K={(S_1,P_1), (S_2,P_3) ….., (S_i,P_i)}

O=Set of Outputs
O= {O1, O2, O3, O4}

Where,
O1=Authentication Message
O2=Encrypted File
O3= Attack Detection
O4= Periodic key
O5=Original Data file

T= Time Period for key generation

F=Set of Functions
F={F1, F2, F3, F4,F5}

Where,
F1=Authentication
O1←F1(I1, I2)
F2=Encryption
O2←F2(I3,K)
F3=Attack Detection
O3←F3(K)
F4= Periodic key Generation
O4←F4(O3,T)
F5= Decryption
O5←F5(O2,K)

7. RESULT TABLES

Following table shows result comes from the system performance, it shows that the the required to create a key with respect to file size of file size.

Table -1: Time required to genearte key

Sr.No	Time(Sec)	File Size(KB)
1	100	7
2	160	8
3	200	11
4	310	12
5	400	13

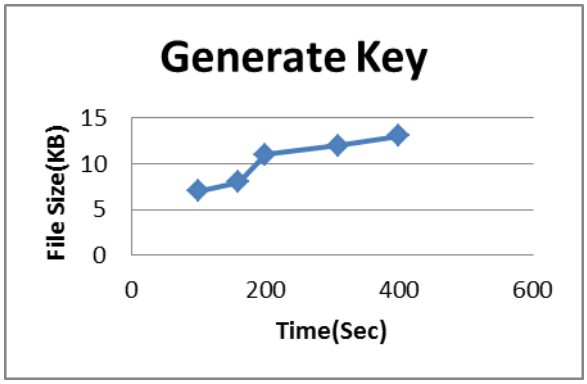

Chart -1: Result graph

As shown in figure 2 the graph represents time require to assign generate key. If file size is increasing rapidly then the time is also increased.

9. IMPLEMENTATION STATUS

Project is developed up to the base paper work, so remains just an proposed or contribution work. Development completed up to the 60% of total work.

10. S/W REQUIREMENT SPECIFICATION

Table - 2: Minimum Hardware Requirements

Hardware	Minimum Requirement
Processor	Pentium Dual Core 2.80 GHz or above
Primary Memory	1GB RAM or more
Secondary Memory	20 GB (minimum)

Table -3: Minimum Software Requirements

Software	Minimum Requirement
Front End (Prog. Lang.)	J2SDK1.5 Java and J2EE
Backend (DB)	My SQL
Development Tool (IDE)	Eclipse

11. POSSIBLE FUTURE WORKS

The cloud storage auditing with key exposure resilience protocol is used in paper .The user can upload their data in the cloud and they can protect their data by using the Third Party Auditor. The key update algorithm is used to protect the client‟s key from the unauthorized user. In paper, the data owner independently upload the data to the Cloud and it is difficult to monitor the data and checking the process in offline. Thus data owner stands in online for integrity checking. This can be achieved by introducing Proxy component to check for the integrity. This is an added advantage to the data owner that he need not stay online for integrity checking. The data owner provides a key to the proxy server using that key proxy is responsible for checking the data. This should be considering as the future work to overcome this drawback.

12. CONCLUSIONS

We examine on the best way to manage the client‟s enter introduction in distributed storage evaluating. We propose another worldview called reviewing convention with key-presentation flexibility. In such a convention, the uprightness of the information beforehand put away in cloud can at present be confirmed regardless of the possibility that the client‟s current mystery key for distributed storage inspecting is uncovered. We formalize the definition and the security model of reviewing convention with key-presentation versatility, and after that propose the principal down to earth arrangement. The security confirmation and the asymptotic execution assessment demonstrate that the proposed convention is secure and proficient.

13. REFERENCES

[1] Vijay varadhanajan , Jia Yu ,Kai Ren “Enabling Cloud Storage Auditing With Key Exposure Resistance” IEEE Transaction on information forensics and security,Vol 10.
[2] Priyadharshni, Geo Jenefer. G “ Enhancing Data Security In Cloud Storage Auditing With Key Abstraction” Vo.2,Issue 2,Oct 2015.
[3] T Yawaikha,R Meyanand, “ An Efficient Cloud Storage Batch Auditing Without Key Exposure Resistance Using Public Verifier” International conference on system 2016
[4] Sneha Singha”Survey Paper On Cloud Storage Auditing With Exposure Resistance” IJSR
[5] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, “Efficient provable data possession for hybrid clouds,” in Proc. 17th ACM Conf. Comput. Commun. Secur., 2010, pp. 756–758.
[6] K. Yang and X. Jia, “Data storage auditing service in cloud computing: Challenges, methods and opportunities,” World Wide Web, vol. 15, no. 4, pp. 409–428, 2012.
[7] H. Shacham and B. Waters, “Compact proofs of retrievability,” in Advances in Cryptology—ASIACRYPT. Berlin, Germany: Springer-Verlag, 2008, pp. 90–107.
[8] C. Wang, K. Ren, W. Lou, and J. Li, “Toward publicly auditable secure cloud data storage services,” IEEE Netw., vol. 24, no. 4, pp. 19–24, Jul./Aug. 2010.
[9] K. Yang and X. Jia, “An efficient and secure dynamic auditing protocol for data storage in cloud computing,” IEEE Trans. Parallel Distrib. Syst., mvol. 24, no. 9, pp. 1717–1726, Sep. 2013.
[10] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy preserving public auditing for secure cloud storage,” IEEE Trans. Comput., vol. 62, no. 2, pp. 362–375, Feb. 2013.