Volume 4, Issue 4, April 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Audit Service for Data Integrity in Cloud

Pravin Rathod, Subhangi Sapkal
Government College of Engineering ,Aurangabad
India

Abstract: ***Cloud computing has been emerged solution to the rising storage costs of IT industry. With the high costs of data storage devices as well as the rapid rate at which data is being generated it proves costly for enterprises or individual users to frequently update their hardware. Apart from reduction in storage costs data outsourcing to the cloud also helps in reducing the maintenance. Cloud storage moves the user's data to large data centers, which are remotely located, on which user does not have any control. However, this unique feature of the cloud poses many new security challenges which need to be clearly understood and resolved. We provide a scheme which gives audit service of data integrity in the cloud which the data owner can employ to verify the correctness of his data in the cloud. This audit can be agreed upon by both the cloud and the data owner and can be incorporated in the Service level agreement (SLA).***

Keywords: ***Audit Service, TPA, Proof of Retrievability, data owner, Data integrity.***

I. INTRODUCTION

Cloud service provider provides data storage on the servers is raising trend among many organization and users owing to its economic advantages. Cloud provides service oriented applications in form of infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) over internet [1]. Many users store their data in clouds that can be accessed remotely over the Internet. Even though the advantages of cloud computing are immense, there are a lot of security issues. One important security problem is the integrity of stored data on cloud [2].

Data integrity is defined as the accuracy and consistency of stored data, in absence of any alteration to the data between two updates of a file or record. Cloud services should ensure data integrity and provide trust to the user privacy. Although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, it's lacking of offering strong assurance of data integrity [3].Cloud computing poses privacy concerns primarily, because the service provider at any point in time, may access the data that is on the cloud. The Cloud service provider could accidentally update or delete some data from the cloud server. Hence, the system must have some sort of mechanism to ensure the data integrity. In current scenario Cloud security model is based on the assumption that the user should trust the provider. This

This is typically governed by a Service Level Agreement (SLA) that in general defines mutual provider and user expectations and obligations. The traditional approach for checking data correctness is to retrieve entire file from server and then verify data integrity by checking the correctness by hash values of entire file. Downloading entire file from cloud to verify data integrity will increase cost or increase burden on communication resources or hardware of data owner, especially when data have been updated or deleted.

Ateniese and Juels [4] provide a method to efficiently verify the integrity of user's data stored in cloud storage server remotely, without retrieving users' data back and without keeping a local copy of user's data. Various settings are studied, including public verification [4], support of dynamic operation, multiple server multiple clients [5]. Public verifiable concept provide remote integrity check relaxes users from the computation and online burden for frequently integrity check, especially desirable when the user have small device (e.g. smart phone, PDA) or is not always connected to the Internet. Furthermore, several methods even support to verify multiple users' data together in batch [5].

In this paper we deal with the problem of implementing a protocol for obtaining a proof of data possession in the cloud sometimes referred to as Proof of retrievability (POR).This problem tries to obtain and verify a proof that the data that is stored by a user at a remote data storage in the cloud (called cloud storage archives) is not modified by the cloud and thereby the integrity of the data is assured. Such verifications are very helpful in peer-to-peer storage systems, network file systems, long term archives, web-service object stores, and database systems. Such verification systems prevent the cloud storage admin from misuse or modifying the data stored at it without the consent of the data owner by using frequent checks on the storage archives. Such verification must allow the data owner to efficiently, frequently, quickly and securely verify that the cloud archive is not cheating the owner [6].

It must be noted that the storage server might not be malicious; instead, it might be simply unreliable and lose or corrupt the owner data. But the data integrity schemes that are to be developed need to be equally applicable for malicious as well as unreliable cloud storage servers. Any such proofs of data possession schemes do not, by itself, protect the data from corruption by the archive. It just allows detection of tampering or deletion of a remotely located file at an unreliable cloud storage server.

While developing proofs for data possession at un trusted cloud storage servers we are often limited by the resources at the cloud server as well as at the owner. Given that the data sizes are large and are stored at remote location server, accessing the entire file can be expensive in I/O costs to the storage server. Also transmitting the file across the network to the client can consume heavy bandwidths. Since increase in storage capacity has far outpaced the growth in data access as well as network bandwidth, accessing and transmitting the entire archive even occasionally greatly limits the scalability of the network resources.

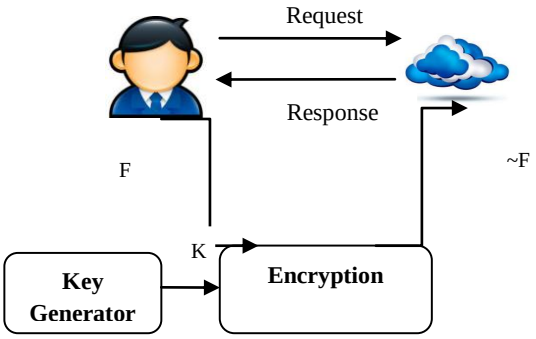

Fig.1. Schematic Diagram of Proof or Retrievability

The problem is created when owner of the data may be small device like PDA, smart phones which have limited computation, battery power and bandwidth. The system should be able to verify data integrity without access the entire file from cloud storage. It is used minimum usage of bandwidth or local computation

II. RELATED WORK

The Proof of retrievability (POR) scheme can be made using a keyed hash function h (F). In this scheme the verifier, before archiving the data file F in the cloud storage, pre-computes the cryptographic hash of F using h(F) and stores this hash as well as the secret key K. To check if the integrity of the file F is lost the verifier releases the secret key K to the cloud archive and asks it to compute and return the value of h (F). By storing multiple hash values for different keys the verifier can check for the integrity of the file F for multiple times, each one being an independent proof. This scheme is very simple and easy to implement but it requires high resource cost in the form bandwidth of network, Computational processor. At the owner or third part auditor involves storing many keys as number of checks it want to perform as well as hash value of data at every file [7]. A POR scheme consists of setup phase and a sequence of verification phases. In the setup phase, data owner preprocess data file using her private key to generate authentication information. Then that file sends to cloud storage server and removes that file from local storage. Owner has only secret key and cloud storage contain information and data file F. In verification phase, owner generate request and produce response from cloud server. In the end of verification phase, owner will verify cloud response using her private key and decide to accept or reject this response [8].

In this system involves the encryption of file F using secret key it becomes heavy data when the data is to be encrypted is large. This system is not suitable for small user with limited computational power (smart phone, low power device). In this scheme special blocks (called sentinels) are hidden among other blocks in the data file F. In the setup phase, the verifier randomly embeds these sentinels among the data blocks. During the verification phase, to check the integrity of the data file F, the verifier request to cloud archive by specifying the positions of a collection of sentinels and asking the cloud archive to return the associated sentinel values. If the cloud admin has modified or deleted a substantial portion of F, then with high probability it will also have suppressed a number of sentinels. It is therefore unlikely to respond correctly to the verifier. To make the sentinels indistinguishable from the data blocks, the whole modified file is encrypted and stored at the archive. The use of encryption here renders the sentinels indistinguishable from other file blocks. This scheme is best suited for storing encrypted files.

The cloud storage model considering here is consists of three main components as follows [9].

a) Cloud User or owner of Data: the user, who can be an individual or an organization originally storing their data in cloud and accessing the data.

b) Cloud Service Provider (CSP): the CSP, who manages cloud servers (CSs) and provides a paid storage space on its infrastructure to users as a service.

c) Third Party Auditor (TPA) or Verifier: the TPA or Verifier, who has expertise and capabilities that users may not have and verifies the integrity of outsourced data in cloud on behalf of users. Based on the audit result, the TPA could release an audit report to user.

III. PROPOSED SYSTEM

We present system which does not involve the encryption of whole file. We encrypt few random selected bits per data blocks in file thus reducing computation on owner side. In this system, owner does not to stored file on his disk. This system is more suitable for thin owner. In our data integrity protocol the third party auditor needs to store only a single cryptographic key-irrespective of the size of the data file F- and two functions which generate a random sequence. The third party does not store any data with it. The verifier before storing the file at the archive preprocesses the file and

appends some metadata to the file and stores at the archive. At the time of verification the verifier uses this metadata to verify the integrity of the data. It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted. It does not prevent the archive from modifying the data.

Audit service for data integrity in cloud based on selecting random bits in data files.

The data owner before storing its data file F at the client should process it and create suitable metadata or secret key which is used for verification data integrity in cloud storage. This process follows following steps

I) Setup phase

We initially preprocess the file and create metadata to append to file. Initial setup phase can follow steps.

a) Generation of metadata

let g be a function as fallows [6]

$g(i, j) \longrightarrow \{1 \ldots m\}$, $i \in \{1...n\}$, $j \in \{1..k\}$ -------- 1

Where k is number of bits per data block. The function g has each data block a set of k bit position within m bits that are in the data block. g (i,j) represents j^{th} bit in i^{th} data block. Audit service should be decide value of k. for each data block we get a set of k bits and for n blocks we get n*k bits. Let m_i represent k bits of Meta data for the i^{th} block [10].

b) Encrypting the metadata

Each Meta data of data blocks m_i is encrypted by suitable algorithm to create a new modified Meta data M_i. Let h be a function which generates a k bit integer α_i for each i. This function is a secret and is known only to the Audit service.

$h: i \longrightarrow \alpha_i$, $\alpha_i \epsilon \{ 0..2^n\}$ ---------- 2

For Meta data (m_i) of each data block the number α_i is to be added to get new k bit number M_i

$M_i = m_i + \alpha_i$ ---------- 3

In this system, we get a set of new Meta data bit blocks. Encryption method provided strong protection for data.

c) Appending metadata

II) Verification phase

Verifier want to verify the integrity of file F. it request to cloud and asks it to response. The request and response are compared and the verifier of third party auditor accepts or rejects integrity of file.

Every file contains number of data block with M size. From every data block k bit positions is selected for encrypted preprocess before storage on cloud storage. This process creates Meta data for every file and stored at owner as well as cloud storage.

The verifier V wishes to the store the file F with the archive. Let this file F consist of n file blocks. We initially preprocess the file and create metadata to be appended to the file. Let each of the n data blocks have m bits in them. A typical data file F which the client wishes to store in the cloud.

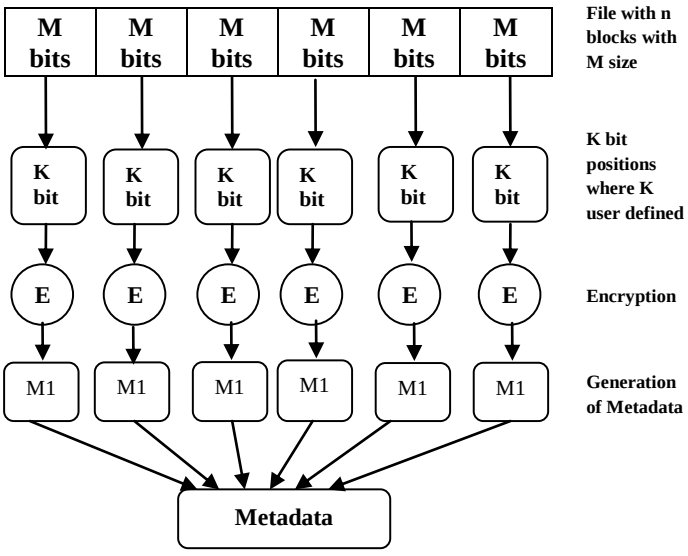

Fig.2. Random Selection bit for Encryption

Each of the Meta data from the data blocks m_i is encrypted by using a suitable algorithm to give a new modified Meta data M_i. Without loss of generality we show this process by using a simple XOR operation. The encryption method can be improvised to provide still stronger protection for verifier's data. All the Meta data bit blocks that are generated using the above procedure are to be concatenated together. This concatenated Meta data should be appended to the file F before storing it at the cloud server. The file F along with the appended Meta data F is archived with the cloud.

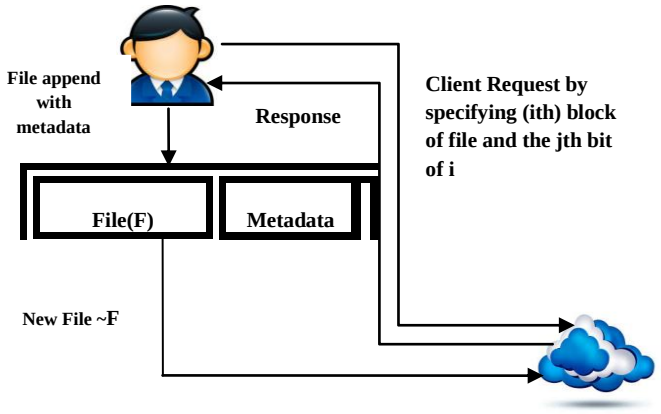

Fig.3. Verification phase in random selection Bit

III. IMPLEMENTATION

In our audit service system the verifier needs to store only single cryptographic key. Auditor does not store any file or data on his side. The verifier stored file on cloud storage before that preprocess file and attach meta data to file and store at cloud storage. Audit service for data integrity in cloud storage implemented model as shown in fig.4. In this model there are two methods as follows.

a) Direct Verification
b) Download verification

a) Direct Verification

This method, owner directly checks integrity of data with help of Secret key or metadata store at owner side. In direct verification method owner match download secret key from cloud storage that is to be compare with original secret. If both keys are matched with each other then data is safe or not modified by third party. If it is not matched then that data is to be modified or altered by third party.

b) Download verification

In this method third party auditor request to data owner for secret key to check integrity of file F. Data owner grant permission to third party auditor for allow download file F. After this response TPA download file F and check integrity of file F. If any modification or altered in file F then that changes reflects cloud admin as well as owner side. Data owner does not store any data on his side. Cloud admin send warning to data owner if any modification or altered data by third party auditor. Without any verification of metadata or secret key owner is to be understood data is to be modified.

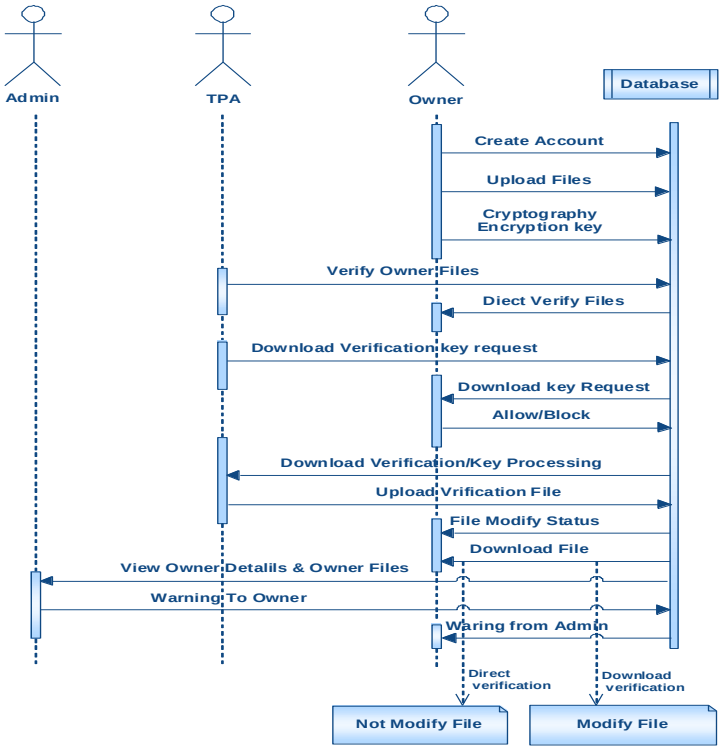

Fig.4. Audit service model for data integrity

V. EXPERIMENTAL RESULT

In this section, we analyze the performance of our audit system in terms of file type, file size and time required for POR and random selection bit.

Table I.
Comparison result for Time required for POR & Random selection Bit

Sr No	File Type	File Size(KB)	Time required for POR (ms)	Time required for Random selected Bit (ms)
1	JPG	581	568	35
2	Xls	625	583	2
3	Pdf	698	578	2
4	Pdf1	650	565	2
5	Doc	110	123	2
6	Doc1	99	85	1
7	Ppt	2471	2031	4

VI. CONCLUSION & FUTURE SCOPE

In this paper we have worked to facilitate the data owner in getting audit service of data integrity which he to store in cloud storage server with minimum costs & effort. Our system was developed to reduce the computational and storage overhead of the client as well as to minimize the computational overhead of the cloud storage server. We also minimized the size of the proof of data integrity so as to reduce the network bandwidth consumption. At the client we only store two functions, the bit generator function g, and the function h which is used for encrypting the data. Hence the storage at the client is very much minimal compared to all other schemes that were developed. Hence this system is suitable for thin clients or small device users like PDAs and smart phones. It should be noted that this system applies only to static storage of data. It cannot handle to case when the data need to be dynamically changed. Hence developing on this will be a future challenge.

REFERENCES:

[1] Satyakshma Rawat ,Richa Chowdhary , Dr. Abhay Bansal "Data Integrity of Cloud Data Storages (CDSs) in Cloud" International Journal of Advanced Research in Computer Science and Software Engineering Research Paper on Volume 3, Issue 3, March 2013

[2] Reenu Sara George, Sabitha S Survey on Data Integrity in Cloud Computing *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 2, January 2013*

[3] Venkatesa Kumar V, Poornima G "Ensuring Data Integrity in Cloud Computing" Journal of Computer Applications ISSN: 0974 –1925, Volume-5, Issue EICA2012-4, February 10, 2012

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring,L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In CCS '07: ACM conference on Computer and communications security, pages 598{609, 2007.

[5] Jia Xu "Auditing the Auditor: Secure Delegation of Auditing Operation over Cloud Storage" Copyright 200X ACM ...$5.00.

[6] Sravan Kumar R, Ashutosh Saxena "Data Integrity Proofs in Cloud Storage" 978-1-4244-8953-4/11/$26.00 c 2011 IEEE

[7] A. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2007, pp. 584–597.

[8] Jia Xu and Ee-Chien Chang "Towards Efficient Proofs of Retrievability in Cloud Storage" Cryptology ePrint Archive: Report 2011/362,15 OCT 2011

[9] Syam Kumar P, Subramanian R "An Efficient and Secure Protocol for Ensuring Data Storage Security in Cloud Computing" In International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1, November 2011.

[10] Ning Cao, Cong Wang, Ming Li, Kui Ren and Wenjing Lou, "Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data,"*IEEE Transactions on Parallel and Distributed Systems* (TPDS), Jan. 2014.

