www.ijecs.in

***International Journal Of Engineering And Computer Science ISSN:2319-7242***
***Volume 6 Issue 4 April 2017, Page No. 20876-20879***
***Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i4.13***

# Enhanced Public Auditing for Code-Based Multi-server Cloud

***Sonal N. Kotangale[1], Pushpa D. Chudhari[2], Prof. Shraddha D. Agnihotri[3]***

[1] Department of Computer Engineering,
MIET College, Bhandara
*kotangalesonal@gmail.com*
[2] Department of Computer Engineering,
MIET College, Bhandara
*pchudhari9823@gmail.com*
[3] Department of Computer Engineering,
MIET College, Bhandara
*shraddhadagnihotri@gmail.com*

**Abstract:** ***cloud computing is a storing and accessing data and programs over the Internet instead of computer's hardware. Cloud computing is gaining popularity because it provides various on demand services which is location independent. Cloud user can store their data on cloud server remotely. So the data storage is resided with third party cloud service provider. In such situation, maintaining the privacy of user's data from unauthorized users is not an easy task. Before storing the data on the cloud server, the data can be encrypted. Using privacy preserving public auditing, data hosting service also brings new security threats toward users data, thus making individuals or enterprisers still feel hesitant. Sometimes data owners lose ultimate control over the fate of their outsourced data so there is risk that availability and integrity of the data might be lost. To protect outsourced data in cloud storage against corruptions, adding fault tolerance to cloud storage together with data integrity checking and failure reparation. Due to lower repair bandwidth while providing fault tolerance Regenerating codes have gained popularity***

**Keywords:** Cloud, Privacy Peserving, Network third Party Auditor, Load balance.

## 1. Introduction

Cloud Computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the possibly large size of outsourced data makes the data integrity protection in Cloud Computing a very challenging and potentially formidable task, especially for users with constrained computing resources and capabilities. Thus, enabling public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met: 1) TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user; 2) The third party auditing process should bring in no new vulnerabilities towards user data privacy. In this paper, we utilize and uniquely combine the public key based homomorphic authenticator with random masking to achieve the privacy-preserving public cloud data auditing system, which meets all above requirements. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient.

Protect outsourced data in cloud storage against corruptions, adding fault tolerance to cloud storage together with data integrity checking and failure reparation. We elaborate on the correctness of verification with public auditing scheme and regenerating codes have gained Data integrity on cloud data storage.

Many mechanisms dealing with the integrity of outsourced data without a local copy have been proposed under different system and security models up to now

## 2. Related work

Due to lower repair bandwidth while providing fault tolerance of regenerating codes have gained popularity. By using existing methods for regenerating-coded data it only provide private auditing and it requires data owners to always stay online and for repairing also there is data owner is require which is sometimes impractical. By manipulating the classic Merkle Hash Tree construction for block tag authentication, it improves the existing proof of storage models and as well as achieve efficient data dynamics, multiple auditing tasks explore the technique of bilinear aggregate signature to extend the result into a multi-user setting with the help of TPA which can perform multiple auditing tasks simultaneously. The proposed schemes are highly efficient and provably secure as per Extensive security and performance analysis.

The most significant work among these studies are the PDP (provable data possession) model and POR (proof of retrievability) model, which were originally proposed for the single-server scenario].

### *2.1 MAC based Solution*

Used for checking integrity in cloud storage. Data owner maintains MACs for the data file to be outsourced

### *2.2 POP (Proof of Retrievability ) scheme*

A keyed hash function is used in POR scheme. Used for single server scenario only.

*2.3 PDP (provable data possession) model*

Verification for data stored in cloud. Do not solve error-correcting codes to address concerns of corruption

## 3. Comparative analysis

A table 1 shown below describes the comparative analysis of a literature review and various techniques

**Table 1:** COMPARATIVE ANALYSIS OF LITERATURE REVIEW

| Technique | Use | Demerit |
|---|---|---|
| 1. MAC based Solution | Used for checking integrity in cloud storage | Data owner maintains MACs for the data file to be outsourced |
| 2. POP (Proof of Retrievability ) scheme | A keyed hash function is used in POR scheme | Used for single server scenario only |
| 3. PDP (provable data possession) model | Verification for data stored in cloud | Do not solve error-correcting codes to address concerns of corruption |

.

## 4. Proposed Work

In our proposed system we address the problem of forwarding data to another user by storage servers directly under the command of the data owner. We consider the system model that consists of distributed storage servers and key servers. Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user. These key servers are highly protected by security mechanisms. In this section we introduced and define our architecture model and algorithm that are used in the privacy auditing there are some problems for designing public auditor in the regenerating-code-based cloud storage using privacy auditing, Using Third Party Auditor which provide more security and allows the user to know the information about data stored. in the cloud, there are some schemes which make data change even if data owner is not online

Considering that files are usually striped and redundantly stored across multi-servers or multi-clouds, explore integrity verification schemes suitable for such multi-servers or multi-clouds setting with different redundancy schemes, such as replication, erasure codes, and, more recently, regenerating codes.

The proposed architecture and method is described below:

### 4.1 Architecture

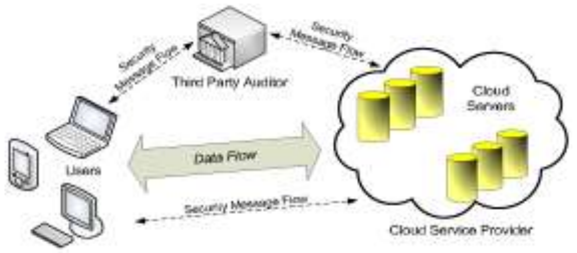

**Figure 1:** *The architecture of cloud data storage service*

Architectural model shown in Figure 1 describes the working of proposed system. cloud is a cluster of large number of computers operated and managed by a single authority. In this system first we are going to create a cloud servers. The system is divided into three parts where first part will be the data owner. Second part will be the centralized server called third party auditor; it will handle the load and perform the auditing in the data on cloud. Third part will be the client which will access services, store data on cloud servers. Here client will sent the request to server to get the service and server will send the reply to clients request, this is the normal flow of network. When number of clients will send the request at the same time to the same server then load balancer will check out the traffic on server side. If there will be more requests to the same server immediately it will transfer to the nearby server not having more traffic to reply with the same service.

### 4.2 Methodology for our auditing scheme:

Our auditing scheme consists of three procedures: Setup, Audit and Repair. Each procedure contains certain polynomial-time algorithms as follows:

*Setup:*

The data owner maintains this procedure to initialize the auditing scheme.

*KeyGen*(1*κ*) → (*pk, sk*): This polynomial-time algorithm is run by the data owner to initialize its public and secret parameters by taking a security parameter *κ* as input.

*Degelation*(*sk*) → (*x*): This algorithm represents the interaction between the data owner and proxy. The data owner delivers partial secret key *x* to the proxy through a secure approach.

*Sig And BlockGen* (*sk, F*) → (_,, *t*): This polynomial time algorithm is run by the data owner and takes the secret parameter *sk* and the original file *F* as input, and then outputs a coded block set , an authenticator set _ and a file tag *t*.

*Audit:*

The cloud servers and TPA interact with one another to take a random sample on the blocks and check the data intactness in this procedure.

*Challenge*(*Finfo*) → (*C*): This algorithm is performed by the TPA with the information of the file *Finfo* as input and a challenge *C* as output.

*ProofGen*(*C*,_,    ) → (*P*): This algorithm is run by each cloud server with input challenge *C*, coded block set  and authenticator set _, then it outputs a proof *P*.

*Verify*(*P, pk, C*) → (0, 1): This algorithm is run by TPA immediately after a proof is received. Taking the proof *P*, public parameter *pk* and the corresponding challenge *C* as input, it outputs 1 if the verification passed and 0 otherwise.

*Repair:*

In the absence of the data owner, the proxy interacts with the cloud servers during this procedure to repair the wrong server detected by the auditing process.

### 4.3 Working of algorithm

Suppose a given server set is S = {S0, S1, ….. Sn-1},

Wi (i=1,..,n) is the weight of each server i..

Ci (i=1,..,n) is the current connections.

ALL_CONNECTIONS is the sum of Ci (i=1,..,n),

the next network connection here will be send to the server j, in which

(Cj/ALL_CONNECTIONS)/Wj = min {

(Ci/ALL_CONNECTIONS)/Wi } ( i=1,..,n )

Since, here the ALL_CONNECTIONS is constant. Hence, in this case it is not needed to divide Ci by ALL_CONNECTIONS and it can be enhanced as

Cj/Wj = min { Ci/Wi } (i=1,..,n)

The scheduling gives an assurance that the server will not be scheduled when its weight is zero.

Below given the pseudo code for weighted least connection scheduling algorithm.

```
Begin
Step 1: for each j = 0 to N
Step 2: check if W(Sj) > 0
Step 3: for each i = j+1 to N
Step 4: check if (C(Sj)*W(Si) > C(Si)*W(Sj))
            Then
             j = i
            (  End of if )
Step 5: return Sj;
            ( End of Loop 2)
Step 6: return NULL;
            ( end of loop 1)
   End
```

**4.4 Regenerating Codes**

Regenerating codes are first introduced for distributed storage to reduce the repair bandwidth. Viewing cloud storage to be a collection of *n* storage servers, data file *F* is encoded and stored redundantly across these servers. Then *F* can be retrieved by connecting to any *k*-out-of-*n* servers, which is termed the MDS2-property. When data corruption at a server is detected, the client will contact $\ell$ healthy servers and download $\beta'$ bits from each server, thus regenerating the corrupted blocks without recovering the entire original file. The privacy protection of the owner's data can be easily achieved through integrating with the random proof blind technique or other technique . However, all these privacy-preservation methods introduce additional computation overhead to the auditor, who usually needs to audit for many clouds and a large number of data owners; thus, this could possibly make it create a performance bottleneck. Therefore, we prefer to present a novel method, which is more light-weight, to mitigate private data leakage to the auditor. Notice that in a regenerating-code-based cloud storage, data blocks stored at servers are coded as linear combinations of the original blocks Supposing that the curious TPA has recovered *m* coded blocks by elaborately performing *Challenge-Response* procedures and solving systems of linear equations. the TPA still requies to solve another group of *m* linearly independent equations to derive the *m* native blocks.

## 5. Expected Outcome

this system will provide a way for providing security to cloud storage by maintaining data integrity and privacy preserving. System uses encryption/decryption keys of user‘s data and stores it on remote server. Each storage server has an encrypted file system which encrypts the client‘s data and store. The system ensures that the client‘s data is stored only on trusted storage servers and it cannot be accessed by administrators or intruders. TPA can perform auditing tasks. Resulted encrypted method is secure and easy to use. Third party auditor can be a trusted third party to resolve the conflicts between the cloud service provider and the client. This paper provides cloud data security using third party auditor. TPA will provide the guarantee data privacy and The users data leakage will also be prevented. This system will be Used for multi-server cloud storage.

## 6. Conclusion

this system will provide a way for providing security to cloud storage by maintaining data integrity and privacy preserving. TPA will provide the guarantee data privacy and The users data leakage will also be prevented. This system will be Used for multi-server cloud storage.

ACKNOWLEDGMENT

The new work presented in this paper has done by Sonal Kotangale and Pushpa chudhari, a student of 8th semester computer science and engineering. The proposed work is ongoing under the guidance of Prof. Shraddha D Agnihotri. The proposed method will be implemented and tested using the software modules.

## .References

[1] Jachak K.B, Korde S.K, Ghorpade P.P and Gagare G.J “Homomorphic Authentication with Random Masking Technique Ensuring Privacy and Security in Cloud Computing”. (2012)

[2] A. Juels and B. S. Kaliski, Jr., “PORs: Proofs of retrievability for large files,” in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 584 597

[3] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, “MRPDP: Multiple-replica provable data possession,” in Proc. 28th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jun. 2008, pp. 411–420.

[4] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren, and Wenjing Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE TRANSACTIONS ON COMPUTERS, vol.62, pp. 362- 375, Ferbruary. 2013.

[5] Cong Wang, Qian Wang, Kui Ren, and Wenjing Lou, " Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing", in IEEE INFOCOM, 2010, pp. 1-15. 2009.

[6] F. sabahi Faculty of computer engineering Azad University Iran." Cloud Computing Security Threats and Responses".

[7] Yuchong Hu Student Member, IEEE, Lee, P.P.C. Student Member, IEEE; Shum, K.W, "Analysis and construction of functional regenerating codes with uncoded repair for distributed storage systems".

[8] Y. Hu, H. C. H. Chen, P. P. C. Lee, and Y. Tang, "NCCloud: Applying network coding for the storage repair in a cloud-of-clouds".