# Data Integrity Constraints in Cloud Computing

**V.Rakesh Goud, Dr. J. Srinivasa Rao [2]**

**[1] Student, NOVA COLLEGE OF ENGINEERING & TECHNOLOGY, Jupudi, Krishna (dt), Andhra Pradesh**

**[2] Professor, NOVA COLLEGE OF ENGINEERING & TECHNOLOGY, Jupudi, Krishna (dt), Andhra Pradesh**

**Abstract:** Now a days cloud is the main storage device for storing outsourced data. Data integrity is the main problem in storing of outsources data. In that security is the main replication on integrity of our sources data. Traditionally cloud auditing can be calculated by using Provable Data Possession (PDP) protocol to prevent fraudulence of power and leakage of verified data.For security they are using Deffie-Hellman assumption and refundable block box knowledge extractor. But Deffie-Hellman assumption doesn't give efficient data extraction on sharing data with third party auditor because it consists some probable actions in security. So, in this paper we are introducing an efficient security mechanism (Hashed Security using MD5) for auditing service in cloud computing. We are calculating the data storage process on cloud computing using auditing services of third party user. Our experimental results shows the efficient data auditing services using hash based priority issues.

**Index Terms: Cloud Computing, DaaS, DBM's, Auditing, Security.**

## 1. INTRODUCTION

Cloud computing delivers on-demand access to essential computing services providing benefits such as reduced maintenance, lower costs, global access, and others. One of its important and prominent services is Database as a Service (DaaS) which includes cloud Database Management Systems (DBMSs). Cloud DBMSs commonly adopt the key-value data model and are called not only SQL (NoSQL) DBMSs. These provide cloud suitable features like scalability, flexibility and robustness, but in order to provide these, features such as referential integrity are often sacrificed.

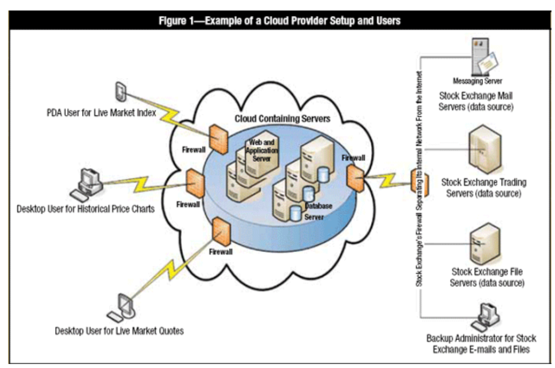


**Figure1: Auditing personalized data on cloud computing**

Data integrity refers to maintaining and assuring the accuracy and consistency of data over its entire life-cycle,[1] and is a critical aspect to the design, implementation and usage of any system which stores, processes or retrieves data. The term data integrity is broad in scope and may have widely different meanings depending on the specific context - even under the same general umbrella of computing. This article provides only a broad overview of some of the different types and concerns of data integrity.Any unintended changes to data as

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

the result of a storage, retrieval or processing operation, including malicious intent, unexpected hardware failure, and human error, is failure of data integrity. If the changes are the result of unauthorized access, it may also be a failure of data security. The cloud storage service (CSS) relieves the burden of storagemanagement and maintenance. However, if such an importantservice is vulnerable to attacks or failures, it would bring irretrievablelosses to users since their data or archives are stored intoan uncertain storage pool outside the enterprises. These securityrisks come from the following reasons: the cloud infrastructures aremuch more powerful and reliable than personal computing devices.

A party that can perform an independent examination of cloud service control with the intent to express an opinion thereon. Audits are performed to verify conformance to standards through a review of objective evidence. A cloud auditor can evaluate the services provided by a cloud provider such as security controls privacy impact and performance.

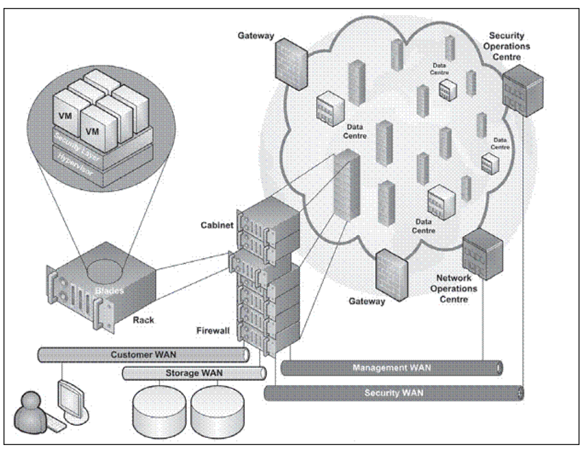


**Figure 2: Security Implications on cloud computing**

Traditionalcryptographic technologies for data integrity andavailability, based on hash functions and signature schemes, cannot work on theoutsourced data without a local copy of data. In addition, it is nota practical solution for data validation by downloading them dueto the expensive transaction, especially for large-size files. Moreover,the solutions to audit the correctness of the data in a cloudEnvironment can be formidable and expensive for the cloud users. In this paper we propose to extend data security considerations on hash based mechanism. These mechanisms are efficiently accessing the services with auditing present in cloud computing.

## 2.RELATED WORK

There has been a considerable amount of work done onuntrusted outsourced storage. The most direct way to enforcethe integrity control is to employ cryptographic hash function.Yumerefendi and Chase proposed a solution for authenticated network storage, using a hash tree (called as Merkle tree) as the underlying datastructure. However their processing of updates is computationally expensive described and implemented a methodfor efficiently and securely accessing a read-only file system thathas been distributed to many providers. This architecture is a solutionfor efficiently authenticating operations on an outsourced file system.
To check the integrity of stored data without download, some researchers have proposed two basic approaches called provable data possession (PDP) and proofs of irretrievability (POR) first proposed the PDP model for ensuring possession of files on untrusted storagesand provided an RSA-based scheme for the static case that achieve so (1) communication costs. They also proposed a publicly verifiableversion, which allows anyone, not just the owner, to challenge theservers for data possession. This property greatly extends applicationareas of PDP protocol due to the separation of data owners andthe authorized users.

## 3. EXISTING SYSTEM

A cloud storage provider only needsto add a corresponding algorithm module to implement this auditservice. Since the audit process could be considered as an interactive Protocol implementation between TPA and this module, sucha module is usually designed as a server daemon to respond auditrequests of TPA through cloud interfaces. This daemon is just a simplelightweight service due to the reason that it does not need totransfer the verified data to the TPA (audit-without-downloadingproperty). Hence, this daemon can be easily appended into variouscloud computing environments.

### 3.1 Construction of interactive audit scheme

A Cryptographic interactive audit scheme (also called as interactive PDP, IPDP) to support our audit system in clouds. This scheme is constructed on the standard model of interactive proof system, which can ensure the confidentiality of secret data (zero-knowledge property) and the deceivability of invalid tags (soundness property).We set up our systems using bilinear pairings proposed be two multiplicative groups using elliptic curve conventions with large prime order p. The function ebe a computable bilinear map e : G×G→GT with following properties: for any G, H∈G and all a, b∈Zp, we have (1) Bilinearity:e([a]G, [b]H) = e(G, H)ab. (2) Non-degeneracy: e(G, H) / = 1 unless G
or H = 1. (3) Computability: e(G, H) is efficiently computable.

## 4. PROPOSED APPROACH

We present our construction of audit scheme in Fig. 3. Thisscheme involves three algorithms: key generation, tag generation, and verification protocol. In the key generation algorithm, eachclient is assigned a secret key sk, which can be used to generatethe tags of many files, and a public key pk, which be used to verifythe integrity of stored files.

**KeyGen**($1^\kappa$)**:** Let $\mathbb{S} = (p, \mathbb{G}, \mathbb{G}_T, e)$ be a bilinear map group system with randomly selected generators $g, h \in_R \mathbb{G}$, where $\mathbb{G}$, $\mathbb{G}_T$ are two group of large prime order $p$, $|p| = O(\kappa)$. Generate a collision-resistant hash function $H_k(\cdot)$ and chooses a random $\alpha, \beta \in_R \mathbb{Z}_p$ and computes $H_1 = h^\alpha$ and $H_2 = h^\beta \in \mathbb{G}$. Thus, the secret key is $sk = (\alpha, \beta)$ and the public key is $pk = (g, h, H_1, H_2)$.

**TagGen**($sk, F$)**:** Splits the file $F$ into $n \times s$ sectors $F = \{m_{i,j}\} \in \mathbb{Z}_p^{n\times s}$. Chooses $s$ random $\tau_1, \cdots, \tau_s \in \mathbb{Z}_p$ as the secret of this file and computes $u_i = g^{\tau_i} \in \mathbb{G}$ for $i \in [1, s]$ and $\xi^{(1)} = H_\xi(\text{“}Fn\text{”})$, where $\xi = \sum_{i=1}^{s} \tau_i$ and $Fn$ is the file name. Builds an index table $\chi = \{\chi_i\}_{i=1}^{n}$, then calculates its tag as

$$\sigma_i \quad \leftarrow \quad (\xi_i^{(2)})^\alpha \cdot g^{\sum_{j=1}^{s} \tau_j \cdot m_{i,j} \cdot \beta} \in \mathbb{G}.$$

where $\xi_i^{(2)} = H_{\xi^{(1)}}(\chi_i)$ and $i \in [1, n]$. Finally, sets $u = (\xi^{(1)}, u_1, \cdots, u_s)$ and outputs $\zeta = (\tau_1, \cdots, \tau_s)$, $\psi = (u, \chi)$ to TTP, and $\sigma = (\sigma_1, \cdots, \sigma_n)$ to CSP.

**Proof**($CSP, TPA$)**:** This is a 3-move protocol between CSP and TPA with the common input $(pk, \psi)$, as follows:

- **Commitment**($CSP \rightarrow TPA$): $CSP$ chooses a random $\gamma \in \mathbb{Z}_p$ and $s$ random $\lambda_j \in_R \mathbb{Z}_p$ for $j \in [1, s]$, and sends its commitment $C = (H_1', \pi)$ to $TPA$, where $H_1' = H_1^\gamma$ and $\pi \leftarrow e(\prod_{j=1}^{s} u_j^{\lambda_j}, H_2)$;
- **Challenge**($CSP \leftarrow TPA$): $TPA$ chooses a random challenge set $I$ of $t$ indexes along with $t$ random coefficients $v_i \in \mathbb{Z}_p$. Let $Q$ be the set $\{(i, v_i)\}_{i\in I}$ of challenge index coefficient pairs. $TPA$ sends $Q$ to $CSP$;
- **Response**($CSP \rightarrow TPA$): $CSP$ calculates the response $\theta, \mu$ as

$$\begin{cases} \sigma' & \leftarrow \quad \prod_{(i,v_i)\in Q} \sigma_i^{\gamma \cdot v_i}, \\ \mu_j & \leftarrow \quad \lambda_j + \gamma \cdot \sum_{(i,v_i)\in Q} v_i \cdot m_{i,j}, \end{cases}$$

where $\mu = \{\mu_j\}_{j\in[1,s]}$. $P$ sends $\theta = (\sigma', \mu)$ to $V$;

**Verification**: $TPA$ can check that the response was correctly formed by checking that

$$\pi \cdot e(\sigma', h) \stackrel{?}{=} e(\prod_{(i,v_i)\in Q} (\xi_i^{(2)})^{v_i}, H_1') \cdot e(\prod_{j=1}^{s} u_j^{\mu_j}, H_2).$$

**Figure 3: Proposed interactive audit protocol.**

In tag generation algorithm, each processed file F will produce apublic verification parameter = (u, _), where u = (_(1), u1, . . ., us),_ ={_i}i∈[1,n] is a hash index table. The hash value _(1) = H_(“ Fn ”) canbe considered as the signature of the secret _1, . . ., _s and u1, . . .,us denotes the “encryption” of these secrets. The structure of hashindex table should be designed according to applications. For example,for a static, archival file, we can define briefly _i = Bi, where Biis the sequence number of block; for a dynamic file, we can also define _i = (Bi||Vi||Ri), where Bi is the sequence number of block, Riis the version number of updates for this block, and Ri is a random Integer to avoid collision.

## 5. EXPERIMENTAL RESULTS

To validate the efficiency of our approach, we have implementeda prototype of an audit system based on our proposed solution. Thissystem have been developed in an experimental cloud computingsystem

environment (called M-Cloud) of Peking University, whichis constructed within the framework of the IaaS to provide powerful
virtualization, distributed storage, and automated management.To verify the performance of our solution, we have simulated ouraudit service and storage service using two local IBM servers withtwo Intel Core 2 processors at 2.16 GHz and 500M RAM runningWindows Server 2003 and 64-bit Redhat Enterprise Linux Server5.3, respectively. These two servers were connected into the MCloudvia 250 MB/s of network bandwidth. The storage server was
Responsible for managing a 16TB storage array based on Hadoopdistributed file system (HDFS) 0.20 clusters with 8 workers nodes located in our laboratory. To develop the TPA's schedule algorithm and CSP's verificationdaemon, we have used the GMP and PBC libraries toimplement a cryptographic library. This C library contains approximately5200 lines of codes and has been tested on Windows and Linux platforms. The elliptic curve utilized in the experiment isa MNT curve, with base field size of 160 bits and the embeddingdegree 6. The security level is chosen to be 80 bits, which means|p| = 160.Firstly, we quantify the performance of our audit scheme under different parameters, such as file size sz, sampling ratio w, sectornumber per blocks, and so on. Our previous analysis showsthat the value of s should grow with the increase of sz in order toreduce computation and communication costs. Thus, our experiments were carried out as follows: the stored files were chosen from 10 KB to 10 MB, the sector numbers were changed from 20to 250 in terms of the file sizes, and the sampling ratios were also changed from 10% to 50%. The experimental results were showed in the left side of Fig. 4. These results dictate that the computation and communication costs (including I/O costs) grow with increase of file size and sampling ratio.

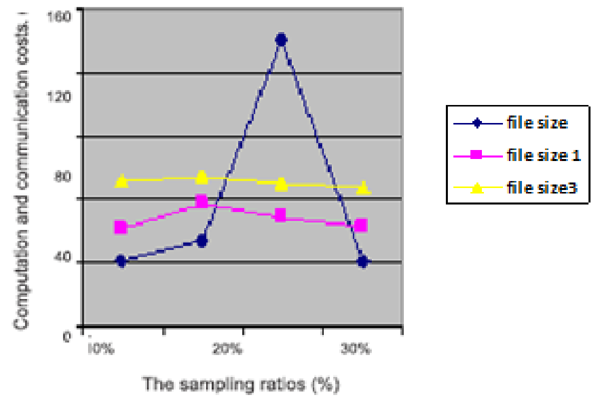


**Figure 4: Experimental results under different size, sampling ratio, second number.**

Unfortunately, these operations prevent any efficient extension toupdate data. Proposed an improvedversion of this protocol called Compact POR, which uses hemimorphicproperty to aggregate a proof into O(1) authenticator value andO(t) computation costs for t challenge blocks, but their solution isalso static and there exist leakages of data blocks in the verification process.

## 6. CONCLUSION

Profiting fromthe standard interactive proof system, we proposed an interactiveaudit protocol to implement the audit service based on athird party auditor. In this audit service, the third party auditor, known as an agent of data owners, can issue a periodicverification to monitor the change of outsourced data by providingan optimized schedule. To realize the audit model, weonly need to maintain the security of the third party auditorand deploy a lightweight daemon to execute the verification protocol.we are introducing an efficient security mechanism (Hashed Security using MD5) for auditing service in cloud computing. We are calculating the data storage process on cloud computing using auditing services of third party user. Our experimental results shows the efficient data auditing services using hash based priority issues.

## 7. REFERENCES

[1] http://en.wikipedia.org/wiki/Data_integrity.

[2] Ateniese, G., Burns, R.C., Curtmola, R., Herring, J., Kissner, L., Peterson, Z.N.J., Song, D.X., 2007. Provable data possession at untrusted stores. In: Proceedings of the2007 ACM Conference on Computer and Communications Security, CCS 2007,pp. 598–609.

[3] Ateniese, G., Pietro, R.D., Mancini, L.V., Tsudik, G., 2008. Scalable and efficient provabledata possession. In: Proceedings of the 4th International Conference onSecurity and Privacy in Communication Networks, SecureComm, pp. 1–10.

[4] Barreto, P.S.L.M., Galbraith, S.D., O'Eigeartaigh, C., Scott, M., 2007. Efficient pairingcomputation on supersingularabelian varieties. Des. Codes Cryptogr. 42 (3),239–271.

[5] Beuchat, J.-L., Brisebarre, N., Detrey, J., Okamoto, E., 2007. Arithmetic operators for pairing-based cryptography. In: Cryptographic Hardware and Embedded Systems– CHES 2007, 9th International Workshop, pp. 239–255.

[6] Yan Zhua,b,∗, HongxinHuc, Gail-JoonAhnc, Stephen S. Yauc," Efficient audit service outsourcing for data integrity in clouds", The Journal of Systems and Software 85 (2012) 1083– 1095, 0164-1212/$ – see front matter © 2011 Elsevier Inc. All rights reserved.doi:10.1016/j.jss.2011.12.024