*International Journal of Engineering & Technology*, 7 (4) (2018) 4791-4794

International Journal of Engineering & Technology

*Website: www.sciencepubco.com/index.php/IJET*
*doi: 10.14419/ijet. v7i4.23667*
***Research paper***

# Cloud data integrity checking methods: a survey

**Anju Susan George [1]*, A Shajin Nargunam [2]**

*[1] Research scholar, Department of CSE, Noorul Islam Centre for Higher Education, Thuckalay India*
*[2] Director – Academics, Department of CSE, Noorul Islam Centre for Higher Education, Thuckalay India*
**Corresponding author E-mail: anjususan_1980@yahoo.co.in*

**Abstract**

Cloud computing is an internet- based computing which provides different services to its users on demand. The users can keep their data in the cloud server without maintaining a native copy. The integrity of the data outsourced can be ensured using various data integrity checking methods. This paper discusses the pros and cons of various techniques for checking data integrity, along with future directions to researchers. The major concern of the data integrity checking methods is the computational load of auditors.

***Keywords***: *Auditing Protocols; Cloud Computing; Cloud Storage; Data Integrity; Third -Party Auditor;*

## 1. Introduction

Today, cloud computing is an inevitable part in academics and industry. It uses virtualized resources and runs on distributed network. The cloud computing services are classified into Software as a Service (SaaS) which provides software applications and its management, Platform as a Service (PaaS) which is used as an environment for developing applications and Infrastructure as a Service (IaaS) which allows storage and its management.
Cloud storage can be grouped into private cloud, public cloud, hybrid cloud and community cloud.
Cloud storage enables users to outsource their data without retaining a local copy. Cloud storage services have a large number of advantages including scalability, cheap cost, accessibility, high computing power, availability and high performance.
In spite of all the advantages of cloud storage, it has some major security concerns. One of the security concerns is the integrity of the data stored. The cloud servers should not be trusted fully. Therefore, it is necessary for cloud users to test the integrity of the data stowed in the cloud frequently. The data integrity verification can be done by a TPA (Third-Party Auditor), on behalf of the client. The TPA ensures that the cloud data can be audited effectively without a native copy. A variety of schemes were proposed by different scholars for integrity- checking.

The basic architecture of cloud data storage is illustrated in Figure 1.

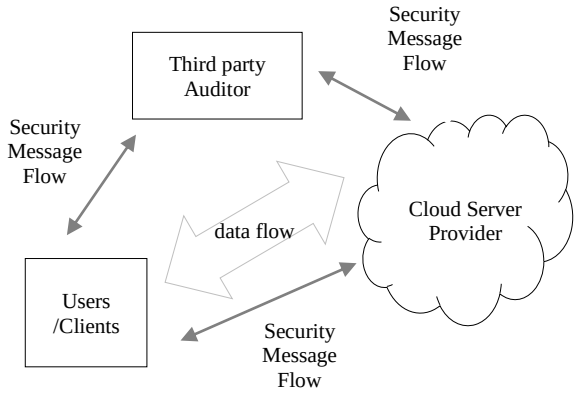


Fig 1 Basic Architecture of Cloud Data Storage

## 2. Related work

The notion of cloud data security, integrity and privacy has been presented in several literatures [11], [12], [18], [21], [22]. The concept of remote data integrity checking was introduced by Deswarte et al [17]. In this method HMAC was used to check files before storing the data into the cloud server. An efficient batch processing scheme was proposed by K Chida and G. Yamamoto [19] which is based on homomorphic hash function. The concept of PDP was proposed by Ateniese et al[16] offers public verifiability. POR is another concept proposed by Juels and Kaliski can retrieve the corrupted data [15] and many more.
Many scholars put forward various cloud data integrity auditing protocols such as dynamic auditing protocols [13], [14], [20], multiple copies auditing protocols [10], privacy preserving auditing protocols [7]- [9], identity- based auditing protocols [3] - [5], public verification protocols[6] and attribute – based auditing protocols[2]. A scheme which reduces the computational overhead

of third party auditor was proposed based on indistinguishability obfuscation [1]. This scheme supports batch verification and dynamic operations.
In this paper a conclusion of several advantages, disadvantages and the protocol sketch of each scheme are proposed. It is summarized in Table 1.

# 3. Cloud data integrity checking methods

Detailed submission guidelines can be found on the journal web pages. All authors are responsible for understanding these guidelines before submitting their manuscript.

### 3.1. Provable data possession (PDP)

Provable data possession is a probabilistic algorithm. It divides the whole file into data blocks for checking integrity instead of using the entire file. The client (data owner) generates a piece of metadata for each data block by pre-computing the file which can be stored locally. The client then transfers the file to the remote server along with the metadata and removes its local copy. The verifier generates a challenge. The remote server responds to challenges provided by the verifier and stores the file. The verifier verifies the proof provided by the server.
PDP adopts a spot checking technique. It supports both encrypted and plain text data. It offers public verifiability. The major drawback of PDP is that it can support only static data.

### 3.2. Proof of retrievability (POR)

POR is designed to handle large files. It ensures possession and retrievability. In this scheme, only a key will be stored by the client. The key is used to conceal the large file in order to get the encrypted file. In this protocol a set of arbitrary blocks called sentinels are inserted into the data file. The sentinels are embedded in such a way that, they are indistinguishable from the original data blocks.
The encrypted file along with the sentinels is transmitted to the remote server. The verifier uses challenge - response protocol. Here the prover is demanded to give back a specific subset of sentinels in the encrypted file. The cloud server or the prover has to specify the positions of a collection of sentinels. If the server has made minor changes to any part of the encrypted file, then it will not be possible for the server to produce a proper proof for the actual file. POR uses error-correcting codes.

### 3.3. Scalable PDP

This is an enhanced form of the original PDP. In order to reduce computation overhead, it uses symmetric key encryption in place of public key encryption. Scalable PDP supports only prefixed number of verifications and if this number to be increased, then the setup algorithm should be re-executed.

### 3.3. Dynamic PDP

Dynamic PDP is same as original PDP. The only difference is that dynamic PDP supports fully dynamic operations. All dynamic operations such as insert, delete, modify, etc can be used with dynamic PDP.

### 3.4. Multiple copies auditing protocols

This scheme is not under the control of a single cloud provider. The notion of multiple replicas in cloud was first proposed by Curtmola et al. [23] which proves the multiple replicas' integrity and needs only one tag. The limitation of this scheme was it does not support dynamic data updates.

Various schemes were proposed that support dynamic uploads by many scholars including Barsoum et al. [24]. Zhang et al. [10] proposed an MR-DPPD protocol which supports variable – sized file blocks and public auditing from RSA signature. This scheme makes use of MR-MHT (Multiple Replication Merkle Hash Tree) that improves the efficiency and also the integrity can be authenticated for all file replicas. Merkle hash tree is a binary tree which could verify the integrity of a set of elements efficiently.

### 3.5. High availability and integrity layer for cloud storage [HAIL]

In this method only, a small amount of data will be stored in local machine. The major portion of the data is distributed across multiple servers. The main function of this method is to provide redundancy. The processing can be done only on static data.

### 3.6. Privacy preserving auditing protocols

This protocol preserves the privacy of the data stored inside the cloud server. The third-party auditor can test the integrity of the data by maintaining the data privacy. During verification, the verifier learns no information regarding the data outsourced. Encryption can be done for this privacy preservation. Complex key management problem is the major issue of this protocol. Batch processing and data dynamics can also be supported by this protocol. Also, this protocol eliminates the computational overhead on the user side.

### 3.7. Identity – based auditing protocols

The complexity of certificate management can be reduced with ID – based auditing protocol in Public-key Infrastructure (PKI). The key management policies such as generation of certificate, storage of certificate, updating of certificate and revocation of certificate are expensive and time-consuming.
In this method, a Key Generation Centre (KGC), will generate private keys for the users based on their identities. The TPA or anyone knowing the identity of the user, is capable of checking the integrity of the data stored in the cloud on behalf of the user. Thus, public verifiability is supported by this protocol.

### 3.7. Attribute – based auditing protocols

The attribute-based system consists of two entities: a key generation centre (KGC) and a user. With the help of a pre-defined attribute set, the Key Generation Centre will generate the matching secret key for a user. After collecting secret key from KGC, the user can produce a signature based on these attributes. The users are allowed to select their attributes while uploading files. Also, they support remote data integrity checking for the data outsourced. It supports multiple TPAs to check data integrity, thus avoiding single- point failure.

# 4. Advantages and limitations

The advantages and limitations of the above mentioned auditing protocols are given below in Table 1.

**Table1:** Advantages and Limitations of Data Integrity Checking Methods

| Sl. No. | Methods | Protocols Used | Advantages | Limitations |
|---|---|---|---|---|
| 1 | PDP | Client: GenKey<br>Client: TagGen<br>Verifier: Challenge<br>Server: ProofGen<br>Verifier: ProofCheck | - Encrypted data and plaintext data are supported<br>- Provides public verifiability | - Supports only static data<br>- Computationally costly |
| 2 | POR | Client: KeyGen<br>Client: Encode<br>Verifier: Challenge<br>Server: Response<br>Verifier: Verify | - Uses Error Correction code<br>- Improves storage reliability<br>- Can handle large files | - Supports only static data<br>- Computationally costly<br>- Only encrypted files can be uploaded to the server<br>- Extra storage space is needed to conceal sentinels |
| 3 | Scalable PDP | Client: GenKey<br>Client: TagGen<br>Server: Update<br>Client: Challenge<br>Server: Response<br>Client: Verify | - Reduce computational overhead<br>- No bulk encryption is required<br>- Dynamic operations are allowed on remote data | - Number of updates is limited<br>- Does not offer public verifiability<br>- Pre-computation is needed |
| 4 | Dynamic PDP | Client: KeyGen<br>Client: AddUpdate<br>Server: ExecuteUpdate<br>Client: ProveUpdate<br>Verifier: Challenge<br>Server: Response<br>Verifier: Verify | - Supports fully dynamic operation<br>- Supports provable updates to stored data | - High computational, communication and storage overhead |
| 5 | HAIL | Client: GenKey<br>Client: Encrypt<br>Server: Decrypt<br>Client: Challenge<br>Server: Response<br>Client: Verify<br>Client/Server: Redistribute | - Supports integrity checking in distributed storage<br>- Proof is compact in size | - Supports only static data |
| 6 | Multiple Copies Auditing Protocols | Client: KeyGen<br>Client: ReplicaGen<br>Client: TagGen<br>TPA: Challenge<br>Server: ProofGen<br>TPA: ProofVerify<br>Server: ExecUpdate<br>Client: VerifyUpdate | - Supports full dynamic updates<br>- Supports simultaneous access and updating of outsourced files<br>- Improves availability and reliability of critical data<br>- Soundness | - Communication overhead while updating and verifying integrity of multiple replicas.<br>- High computational overhead of TPA |
| 7 | Privacy Preserving Auditing Protocols | TA: GlobeSetup<br>Client: Setup<br>Client: TagGen<br>TPA: Challenge<br>Server: GenProof<br>TPA: CheckProof | - Zero-Knowledge privacy<br>- Supports batch auditing<br>- Supports data dynamics<br>- Efficient<br>- Performs lightweight computing | - TA's cannot be trusted fully<br>- High computational and storage overhead of TPA<br>- Complex key management |
| 8 | Identity – Based Auditing Protocols | KGC: Setup<br>KGC: Extract<br>Client: TagGen<br>TPA: Challenge<br>Server: ProofGen<br>TPA: ProofCheck | - Private, Public and Delegated Remote Data Integrity<br>- Flexible<br>- Efficient<br>- Soundness<br>- Perfect data privacy<br>- Reduces the complexity of PKI certificate management<br>- Timing results are persistent forSetup, Extract and TagGenprotocols<br>- Very Fast Extract and Setup algorithms<br>- File tag computation is a one-time task | - High computational overhead of TPA<br>- Unique ID must be chosen<br>- Client must prove his identity before KDC<br>- ID must be remembered<br>- TagGen algorithm is expensive<br>- Timing cost increases as the number of challenges increases for the steps Challenge, ProofGen and ProofCheck<br>- Tag generation time for a file increases with the increase of the file size linearly for on-line as well as off- line processes |

| | | | | |
|---|---|---|---|---|
| 9 | Attribute – Based Auditing Protocols | KGC: SetUp<br>KGC: Extract<br>Client: MetaDataGen<br>TPA: Challenge<br>Server: Response<br>TPA: Verify | - Avoids single – point failure<br>- Attribute selection during uploading only<br>- Flexible<br>- Soundness<br>- Preserves privacy of attributes<br>- Collusion resistance<br>- Reduces the complexity of PKI certificate management<br>- Constant time- consumption during online phase is independent of the block size<br>- The number of data blocks decrease as the block size increase, which leads to less calculation during verification | - The setup algorithm's time cost increases with the number of attributes<br>- The off-line phase time - cost increases with the increased number of data<br>- High computational cost in metadata generation<br>- When the challenged blocks increase the TPA server and the cloud server cost increases linearly |

## 5. Conclusion

In this paper, a survey of various cloud data integrity checking methods is presented. The merits and demerits of different methods are discussed. Based on the survey, it can be concluded that, the major problem of cloud integrity techniques is the high computational overhead of third party auditor and cloud server. One of the future research directions is to develop a secure multi-party computation scheme on a semi-trusted cloud setting for balancing the computational load.

## References


[1] Yuan Zhang et al., "Efficient Public Verification of Data Integrity for Cloud Storage Systems from Indistinguishability Obfuscation", *IEEE Transactions of Information Forensics and Security, vol.12, no.3, pp. 676 – 688, March 2017.*

[2] Yong Yu, Yannan Li, Bo Yang, Willey Susilo et al., "Attribute-based Cloud Data Integrity Auditing for Secure Outsourced Storage", *IEEE Transactions on Emerging Topics in Computing,vol. 14, no.8, pp. 1-13, October 2017.*

[3] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, G.Min. "Identity-Based Remote Data Integrity Checking with Perfect Data Privacy Preserving for Cloud Storage". *IEEE Trans. Information Forensics and Security vol.12, no.4, pp.767-778, April 2017.*

[4] H. Q.Wang. "Identity-Based Distributed Provable Data Possession Multicloud Storage". *IEEE Transactions on Services Computing vol. 8, no.2, pp.328-340, April 2015.*

[5] Huaqun Wang, Debiao He and Shaohua Tang, "Identity – Based Proxy Oriented Data Uploading and Remote Data Integrity Checking in Public Cloud", *IEEE Transactions of Information Forensics and Security, vol.11, no.6, pp. 1165 – 1176, June 2016.*

[6] Yong Yu, Jianbing Ni, Man Ho Au, Yi Mu, Boyang Wang and Hui Li, "Comments on a Public Auditing Mechanism for Shared Cloud Data Service", *IEEE Transactions on Services Computing, vol. 8, no.6, pp. 998 -999, November/ December 2015.*

[7] Y. Yu, M.H. Au, Y. Mu, S.H. Tang, J. Ren, W. Susilo and L.J. Dong. "Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage". *International Journal of Information Security., vol. 14, no. 4, pp.307-318, 2015.*

[8] .Jiangtao Li, Lei Zhang, Joseph K. Liu, Haifeng Qian and Zheming Dong, Privacy-Preserving Public Auditing Protocol for Low Performance End Devices in Cloud, *IEEE Transactions on Information Forensics and Security vol.11, no.11, pp:2572-2583, November 2016.*

[9] Cong Wang, S. M. Chow,Qian Wang, Kui Ren and Wenjing Lou, "Privacy – Preserving Public Auditing for Secure Cloud Storage", *IEEE Transactions on Computers, vol. 62, no. 2, pp. 362 – 375, February 2013.*

[10] Y. Zhang, J. Ni, X., Y. Wang, and Y. Yu. "Provable multiple replicationdata possession with full dynamics for secure cloud storage". *Concurrency and Computation: Practice and Experience, vol. 28, no.4, pp.1161- 1173, June 2016.*

[11] Kumar Parasuraman, P. Srinivasababu, S.Rajula., and T.Arumuga Maria Devi, "Secured Document Management through A Third Party Auditor Scheme in Cloud Computing", in *International Conference Electronics, Communication and Computational Engineering, pp.109 – 118, IEEE 2014.*

[12] Sophia Yakoubov, Vijay Gadepally, Nabil Schear, Emily Shen and ArkadyYerukhimovich, "A Survey of Cryptographic Approaches to Securing Big-Data Analytics in the Cloud", *High Performance Extreme Computing Conference 2014 IEEE.*

[13] Y. Yu, J.B. Ni, M. H. Au, H.Y. Liu, H.Wang and C.X. Xu. "Improved security of a dynamic remote data possession checking protocol for cloud storage". *Expert Syst. Appl.vol. 41, no. 17, pp.7789-7796, June 2014.*

[14] Erway C C, Kupcu A, Papamanthou C, et al. "Dynamic provable data possession". *ACM Transactions on Information and System Security (TISSEC), vol. 17, no. 4, April 2015.*

[15] Ari Juels and Burton S Kaliski, "Pors: proofs of retrievability for large files", *Proceedings of the 14th ACM conference on CCS, pp. 583 – 597, 2007.*

[16] G. Ateniese et al., "Provable Data Possession at Untrusted stores", in*Proc.CCS, pp.598 – 609, 2007.*

[17] Y. Deswarte, J.J Quisquarter and A Saidane, "Remote Integrity Checking", *Integrity and Internal Control in Information Systems VI. Springer US, pp 1-11,2004.*

[18] Yindong Chen, Liping Li, Ziran Chen, "An approach to Verifying Data Integrity for Cloud Storage", *13th International Conference on Computational Intelligence and Security, pp, 582-585, IEEE 2017.*

[19] K Chida, G. Yamamoto, "Batch Processing of Interactive Proofs", *Cryptographers Track at the RSA Conference, Springer Berlin Heidelberg, San Fransisco, USA, 2007, pp. 196-207.*

[20] Kan Yang and Xiaohua Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing", *IEEE Transactions on Parallel and Distributed Systems, vol. 24, no.9, pp. 1717-1726, September 2013.*

[21] Zhifeng Xiao and Yang Xiao, "Security and Privacy in Cloud Computing", *IEEE Communications Surveys & Tutorials, vol.15, no.2, pp. 843 – 859, 2013.*

[22] Sultan Aldossary and William Allen, "Data Security, Privacy, Availability and Integrity in Cloud Computing: Issues and Current Solutions", *International Journal of Advanced Computer Science and Applications, vol.7, no.4, pp. 485 –498, 2016.*

[23] Curtmola R, Khan O, Burns RC, G. Ateniese, "MR-PDP: Multiple – Replica Provable Data Possession", *Proceedings of ICDCS'08, Beijing, pp. 411-420, June 2008.*

[24] Barsoum AF and Hasan MA, "Integrity Verification of Multiple Data Copies Over Untrusted Cloud Servers*", Proceedings of CCGRID'12, Ottawa, pp. 829 – 834, May 2012.*