A Survey on Data Integrity Techniques in Cloud Computing

Mahesh S.Giri
Research Scholar
Computer Science & Engineering
Technocrats Institute of Technology
Bhopal, India

Bhupesh Gaur, Ph.D
Head of Department
Computer Science & Engineering
Technocrats Institute of Technology
Bhopal, India

Deepak Tomar
Assistant Professor
Computer Science & Engineering
Technocrats Institute of Technology
Bhopal, India

ABSTRACT
Cloud computing which is envisioned as the next generation architecture of IT Enterprise comes into focus when someone thinks about what IT always needs. It is way to increase capacity or add capabilities without investing in infrastructure as well as licensing cost on new software. Besides of this advantage there is one major problem that needs to face while keeping sensitive data in cloud, Assurance of data integrity that is data remain as it is on server for long time. Client cannot physically access the data from the cloud server directly, without client's knowledge, Cloud Service Provider (CSP) can alter or delete data which are either unused by client from a long a time or takes large memory space. Hence, there is need of checking the data periodically for its integrity, checking data for correction is called data integrity. To overcome data integrity problem, many techniques are proposed under different systems and security models. This paper will focus on some of the integrity proving techniques in detail along with their limitations.

Keywords
Survey, Cloud Computing, Data Integrity,

1. INTRODUCTION
Cloud computing is a network where user can use services provided by CSP on pay per use bases. It is a research area where will get the benefits of its advantages-on demand service, location rapid resource elasticity ,independent resource pooling, and pay and use based policy. It is derived from Grid computing but still it make its own unique identity in IT industry and provides three service model SaaS, PaaS and IaaS In this paper section 1.1 explains about cloud computing, section 1.2 deals with challenges in cloud commuting section 1.3 discuss about data integrity. The sections of this paper is organized as follows: section two describes the proving techniques currently used to ensure data integrity, section three focuses comparative study of the surveyed papers, and the paper is concluded finally in section five.

2. INTRODUCTION TO CLOUD COMPUTING AND DATA INTEGRITY

2.1 Cloud Computing
According to Hewitt, C. cloud computing is defined as a next generation computing model for enabling convenient, efficient, on-demand network access to a shared pool of configurable computing resources[1]. The growing need of Technology in every field has lead to the evolution of cloud computing for highly efficient usage of IT resources. Cloud Storage is an important service of cloud computing, as it allows data owners to move their data remotely. More and more data owners start choosing to host their data in the cloud.

2.2 Challenges in Cloud Computing
As cloud provides many advantages but as every coin has 2 side, and cloud computing is no exception, it also has certain challenges. Every day, a fresh news item, latest publication, blog entry, highlights the cloud computing's challenges and issues. In each technology there are some security issues that affect the usage and the behavior below some of these concerns in the cloud: [2]

- Access: When there is an unauthorized access to the data, the ability of altering on the client data arise.
- Availability: The data must be available all the time for the clients without having problems that affect the storage and lead to the client data lose.
- Network Load: The over load capacity on the cloud may drop the system out according to the high amount of data between the computers and the servers.
- Integrity: The data correctness, legality and security is the most fields that influence on the cloud and have major lay on the service provider.
- Data Location: The client does not know the actual place that the data saved or centered in because it distributed over many places that led to confusion.

One of the important concerns in the cloud computing that need to be addressed is to assure the customer of the integrity, accordingly in the next section will discuss about data integrity.

2.3 Data Integrity
Integrity, in terms of data security, is nothing but the guarantee that data can only be accessed or modified by those authorized to do so, in simple word it is process of verifying data. Data Integrity is very important among the other cloud challenges. As data integrity gives the guarantee that data is of high quality, correct, unmodified. After storing data to the cloud, user depends on the cloud to provide more reliable services to them and hopes that their data and applications are in secured manner. But that hope may fail sometimes the user's data may be altered or deleted. Sometimes, the cloud service providers may be dishonest and they may discard the data which has not been accessed or rarely accessed to save the storage space or keep fewer replicas than promised [3]. Moreover, the cloud service providers may choose to hide

data loss and claim that the data are still correctly stored in the Cloud. As a result, data owners need to be convinced that their data are correctly stored in the Cloud. So, one of the biggest concerns with cloud data storage is that of data integrity verification at untrusted servers. In order to solve the problem of data integrity checking, many researchers have proposed different systems and security models.

3. CURRENT DATA INTEGRITY PROVING TECHNIQUES THEIR CHALLENGES

In Cloud computing the issue of data integrity is still carried out by many researchers. There is lot of research still going on in this field to provide secure and efficient data integrity in cloud computing. Researchers have given many solutions to focus on resolving the issues of data integrity.

This section will try to focus on few such techniques .This paper provide survey on the different techniques of data integrity and there limitation. The basic schemes for data integrity in cloud are Provable Data Possession (PDP) and Proof of retrievability (PoR). The following section describes the privacy techniques for data integrity.

3.1 Provable Data Possession (PDP)

Provable Data possession (PDP) is a technique for assuring data integrity over remote servers. In PDP A client that has stored data at an unfaithful server can verify that the server possesses the original data without retrieving it. Ateniese et al. are the first to consider public audit ability in their defined "provable data possession" model for ensuring possession of files on untrusted storages. [4]

Principal of PDP:

The working principal of PDP is as shown in fig 1.It works in two stages. Set up stage and challenge stage.

Set up stage:

- The client generates pair of matching keys public & secrete key by using probabilistic key generation algorithm.
- Public key along with the file will be sent to the server for storage by client and he deletes the file from its local storage.

Challenge stage:

- The client challenges the server for a proof of possession for a subset of the blocks in the file.
- The client checks the response from the server.

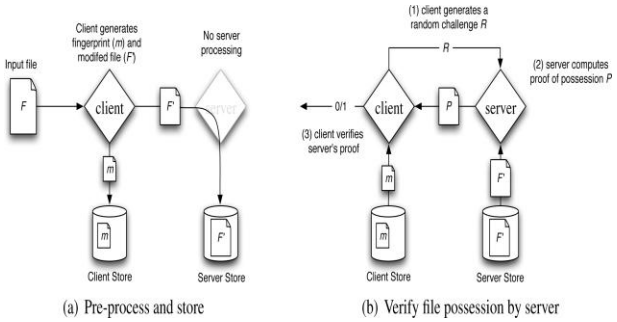

Fig: 1 Protocol for Provable Data Possession [4]

Advantages:

- The server does not actually have to access the file blocks, supporting Block less verification.
- Allows public verifiability.

Limitations:

- Lack of error-correcting codes to address concerns of corruption.
- Lack of privacy preservation.
- No dynamic support.
- Unbound no. of queries.

3.2 Basic PDP Scheme based on MAC

In paper [5] author proposed MAC based PDP to ensure data integrity of file F stored on cloud storage in very simple way .The data owner computes a Message Authentication Code (MAC) of the whole file with a set of secret keys and stores them locally before outsourcing it to CSP. It Keeps only the computed MAC on his local storage, sends the file to the CSP, and deletes the local copy of the file F. Whenever a verifier needs to check the Data integrity of file F, He/she sends a request to retrieve the file from CSP, reveals a secret key to the cloud server and asks to recompute the MAC of the whole file, and compares the re-computed MAC with the previously stored value.

Limitations:

- The number of verifications allowed is limited by the number of secret keys.
- The data owner has to retrieve the entire file of F from the server in order to compute new MACs, Which is not possible for large file.
- Public auditability is not supported as the private keys are required for verification.

3.3 Scalable PDP

Author in [4] proposed Scalable PDP which is an improved version of the original PDP. The main difference is Scalable PDP uses the symmetric encryption whereas original PDP uses public key to reduce computation overhead. Scalable PDP can have dynamic operation on remote data. Scalable PDP has all the challenges and answers are pre-computed and limited number of updates. Scalable PDP does not require bulk encryption. It relies on the symmetric-Key which is more efficient than public-Key encryption. So it does not offer public verifiability.

Limitations:

- A client can perform limited number of updates and challenges.
- It does not perform block insertions; only append-type insertions are possible.
- This scheme is problematic for large files as each update requires re-creating all the remaining challenges.

3.4 Dynamic PDP

A author in [6] proposed Dynamic PDP which is a collection of seven polynomial-time algorithms (KeyGen DPDP, PrepareUpdate DPDP, PerformUpdate DPDP, VerifyUpdate DPDP, GenChallenge DPDP ,Prove DPDP,Verify DPDP) .it

supports full dynamic operations like insert, update, modify, delete etc. Here in this technique uses rank-based authenticated directories and along with a skip list for inserting and deleting functions .It has DPDP some computational complexity, it is still efficient. For example, for verifying the proof for 500MB file, DPDP only produces 208KB proof data and 15ms computational overhead. This technique offers fully dynamic operation like modification, deletion, insertion etc. as it supports fully dynamic operation there is relatively higher computational, communication, and storage overhead. All the challenges and answers are dynamically generated.

Limitations:

- It has some computational complexity.
- Not suitable for thin client.
- DPDP does not include provisions for robustness.

3.5 Basic Proof of Retrievability (PoR):

Proofs of Retrievability (POR) is a cryptographic method for remotely verifying the integrity of files stored in the cloud, without keeping a copy of the user's original files in local storage. In a scheme, user backups his data file together with some authentication data to a potentially dishonest cloud storage server. User can check the data for its integrity stored with CSP using the authentication key, without retrieving back the data file from cloud.(7)

Principal of POR

A POR works on two phase first is setup phase and another is sequence of verification phases.

Setup phase:

In the setup phase, user preprocesses his data file using his private key to generate some authentication code. Then he sends the data file together with authentication code to the cloud storage server, and removes them from his local disk. Consequently, in the end of setup phase user has his private key in her local disk, and CSP has both the data file and the corresponding authentication code.

Sequence of verification phases:

In each sequence of verification phase, user generates a random challenge query and CSP is supposed to produce a short response or proof upon the received challenge query, based on user's data file and the corresponding authentication information. In the end of a verification phase, user will verify CSP's response using his private key and decide to accept or reject this response coming from CSP.

Limitations:

It only works with static data sets.

- It supports only a limited number of queries as a challenge since it deals with a finite number of check blocks.
- A POR does not provide in prevention to the file stored on CSP.

3.6 POR based on keyed hash function hk(F)

A keyed hash function is very simple and easily implementable .It provides the strong proof of integrity. In this method the user, pre-computes the cryptographic hash of F using hk(F) before outsourcing the data file F in the cloud storage, and stores secret key K along with computed hash. The user releases the secret key K to the CSP to check the integrity of the file F and asks it to compute and return the value of hk(F). If the user want to check the integrity of the file F for multiple times he has store multiple hash values for different keys.

Limitations:

- Verifier need to store key for each of checks it wants to perform as well as the hash value of the data file F with each hash key.
- It requires higher resource costs for the implementation as every time hashing has to perform on entire file.

Computation of the hash value for large data files can be computationally burdensome for thin clients.

3.7 Proof of Retrievability for large files

Authors of the paper in [8] technique "Proof of Retrievability" for large files using "sentinels". In this method, only a single key can be used irrespective of the size of the file or the number of files the user needs to access only a small portion of the file F. This small portion of the file F is in fact independent of the length of F.

In this method special sentinels blocks, which are hidden among other blocks in the data file F are randomly embeds among the data blocks. To check the integrity of the data file F, the user challenges the CSP during the verification phase by specifying the positions of a collection of sentinels and asks the CSP to return the associated sentinel values. If the CSP has modified or deleted some portion of F, then it possible that the position of sentinels also changed. Therefore it is unlikely to respond correctly to the CSP. The encryption is performed on whole modified file to indistinguish the sentinels from the data blocks, and stored in the CSP.

Limitations:

- This technique put the computational overhead for large files as encryption is to be performed on whole file.
- This method put storage overhead on the server, because of newly inserted sentinels and partly due to the error correcting codes that are inserted.
- To check the integrity of file user need to download whole file which increases of input/output and transmission cost across the network.
- This method works only with static data.

3.8 HAIL

Authors in [9] proposed HAIL, high-availability and integrity layer for cloud storage, in which HAIL allows the user to store their data on multiple servers so there is a redundancy of the data. Simple principal of this method is to ensure data integrity of file via data redundancy. HAIL uses message authentication codes (MACs), the pseudorandom function, and universal hash function to ensure integrity process. The proof is generated is by this method is independent of size of data and it is compact in size.

Limitations:

- Mobile adversaries are biggest threat which attack on HAIL, which may corrupt the file F.

- This technique is only applicable for the static data only.
- It requires more computation power.
- Not suitable for thin client.

3.9 POR Based on Selecting Random Bits in Data Blocks

In [10] author proposed a technique which involves the encryption of the few bits of data per data block instead of encrypting the whole file F thus reducing the computational burden on the clients. It's stands on the fact that high probability of security can be achieved by encrypting fewer bits instead of encrypting the whole data. The client storage computational overhead is also minimized as it does not store any data with it and it reduces bandwidth requirements. Hence this scheme suits well for thin client. In these techniques user needs to store only a single cryptographic key and two random sequence functions. The user does not store any data in its local machine. The user before storing the file at the CSP preprocesses the file and appends some Meta data to the file and stores at the CSP. At the time of verification the verifier uses this Meta data to verify the integrity of the data.

Limitations:

- This technique is only used for Static Data.
- No data prevention mechanism is used in this technique.
- No Data Prevention mechanism is implemented in this technique.

4. COMPARATIVE STUDY

This Comparative study provides a brief explanation of all the techniques that have been discuss so far in this paper.

Data Integrity Techniques	Method used for Data Integrity	Advantages	Limitations
Provable Data Possession	Key Generation Algorithm	I. This technique gives a strong proof of data integrity. II. Support Block less verification. III. Allows public verifiability.	I. Lack of error-correcting codes to address concerns of corruption. II. Lack of privacy preservation. III. No dynamic support IV. Unbound no. of queries.
PDP Scheme based on MAC	Message Authentication Code	I. Simple & Secure Technique. II. Gives strong proof Integrity of Data.	I. Limited number of verifications with limited number of secret keys. II. The data owner has to retrieve the entire file of F from the server in order to compute new MACs, Which is not possible for large file. III. Public auditability is not supported as the private keys are required for verification.
Scalable PDP	Cryptographic Hash function & symmetric key encryption	I. Does not require bulk encryption. II. Supports dynamic operations on outsourced data blocks.	I. Limited number of updates and challenges. II. Does not perform block insertions anywhere only append-type insertions are possible. III. Problematic for large files as each update requires re-creating all the remaining challenges.
Dynamic PDP	Rank-based authenticated skip list	I. Offers fully dynamic operation.	I. It has some computational complexity. II. Not suitable for thin client. III. DPDP does not include provisions for robustness.
Basic Proof of Retrievability	Encryption	I. Reduces the computational and storage overhead of the client as well as CSP.	I. It only works with static data sets. II. It supports only a limited number of queries as a challenge since it deals with a finite number of check

		II. It also minimizes the size of the proof of data integrity as reduces the network. Bandwidth.	blocks. III. A POR does not provide in prevention to the file stored on CSP.
POR based on keyed hash function hk	Key Hash Function	I. Simple and easily implementable.	I. More number of keys for each check. II. Requires high cost for computation. III. Puts the computational burden on client as well as server.
POR for large files	Sentinel-based scheme	I. Ensures both possession and retrievability of files on CSP.	I. Newly inserted sentinels and error correcting codes Put computational overhead. II. Increases input/output and transmission cost across the network. III. Works only with static data.
High Availability Integrity Layer (HAIL)	MAC, Pseudorandom function, Hash Function	I. Allow user to store data on multiple cloud.	I. This technique is only applicable for the static data only II. Not suitable for thin client
POR Based on Selecting Random Bits in Data Blocks	Generation of Meta Data	I. This technique is suitable for thin client. II. Put minimum storage overhead on client and CSP.	I. This technique only support for static data. II. No Data Prevention mechanism is implemented in this technique.

5. CONCLUSION

In the world of cloud computing the data integrity is most challenging and burning security issue. By considering the importance of data integrity, in this paper different existing paper techniques and their merits and demerits are explained. The analytical study briefly compares all this techniques. From this survey paper it is conclude that there is need to design efficient, dynamic secure data integrity technique which is still wide area of research.

6. FUTURE SCOPE

From the above comparative study it is clear that all these techniques which are surveyed in this paper have some advantages as well as some limitation. All those papers were lack in proper data integrity mechanisms, supporting dynamic data operations, and by high resource and computation cost. The technique POR Based on Selecting Random Bits in Data Blocks is best suited for thin client as well as this technique provides the strong proof of retrievability. The only drawback of this technique is it works only for static data and no data prevention mechanism. So expanding the scope of this paper will be the future work.

7. REFERENCES

[1] Hewitt, C. (2008) ”ORGs for scalable, robust, privacy friendly client Cloud Computing Environment in IEEE Proceedings Volume 12 Issue 5, September 2008.

[2] Chandran S. and Angepat M., “Cloud Computing: Analyzing the risks involved in cloud computing environments,” in Proceedings of Natural Sciences and Engineering, Sweden, 2010.

[3] Balachandra Reddy Kandukuri, Ramakrishna Paturi V, Dr. Atanu Rakshit, "Cloud Security Issues”, in Proceedings IEEE International Conference on Services Computing, September 2009.

[4] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” in Proceedings of SecureComm ’2008.

[5] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, “Auditing to keep online storage services honest,” in Proceedings of the 11th USENIX workshop on Hot topics in operating systems, 2007.

[6] Berkeley, CA, USA, 2007, pp. 1–6. C. Erway, A. K¨up¸c¨u, C. Papamanthou, and R. Tamassia. Dynamic provable data possession in Proceedings of the 16th ACM conference on Computer and communications security, CCS ’09, New York, NY, USA, 2009.

[7] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores,” in Proceedings of 14th ACM Conf. Computer and Comm. Security (CCS ’07), 2007.

[8] A. Juels and B.S. Kaliski Jr., “Pors: Proofs of Retrievability for Large Files,” in Proceedings of 14th ACM Conf. Computer and Comm. Security (CCS ’07), 2007.

[9] K.D. Bowers, A. Juels, and A. Oprea, HAIL: A high-availability and integrity layer for cloud storage, in Proceedings of 16th ACM conference on Computer and communications security, 2009.

[10] R. Sravan kumar and Saxena ,”Data integrity proofs in cloud storage” in Proceedings of IEEE 2011.

[11] Bo Chen and Reza Curtmola. "Robust Dynamic Provable Data Possession," in Proceedings of IEEE 2008.

[12] B. Priyadharshini and P. Parvathi, "Data Integrity in Cloud Storage", in Proceedings of IEEE 2012.

[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," in Proceedings of 17th Int'l Workshop Quality of Service 2009.

[14] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," in Proceedings of Fourth Int'l Conf.Security and Privacy in Comm. Networks 2008.

[15] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, and Jin Li "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" in Proceedings of IEEE Transactions on Parallel And Distributed Systems, VOL. 22, NO. 5, MAY 2011.

[16] K. Zeng, "Publicly verifiable remote data integrity," in Proceedings of ICICS, 2008.

[17] Kevin D. Bowers, Ari Juels, Alina Oprea, Proofs of Retrievability: Theory and Implementation, CCSW'09, in Proceedings of Journal of Systems and Software, May, 2012.

[18] E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," in Proceedings of IEEE Transactions, vol. 2, no. 2, 2006.

[19] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable DataPossession," in Proceedings of 28th IEEE ICDCS, 2008.

[20] Hovav Shacham1 and Brent Waters, Compact Proofs of Retrievability, in Proceedings of International Association for Cryptologic Research 2008.

[21] E. Stefanov, M. van Dijk, A. Oprea, and A. Juels, "Iris: A scalable cloud file system with efficient integrity checks," in Proceedings of IACR ePrint Cryptography Archive, Tech. 2011.