Data Integrity Challenges in Cloud Computing

Sunita Sharma
BCIIT Kalka ji New Delhi, Faridabad, India
sunitasharmacse@gmail.com

Abstract: - *Cloud, jargon in computing, is a way to increase capacity and add capabilities without investing in infrastructure. It also saves licensing/renewal cost for new software. This evolution and ease of use comes with a number of data integrity and security issues. Data integrity concerned with correctness of data and security concerned with preventing unauthentic data access. Data integrity most critical concern of cloud storage as it assures data remain as it is on server for long time. Client cannot access the data from the cloud server directly, without CSP's (Cloud Service Provider) knowledge. Cloud Service Provider can modify/delete data, which are either unused by client from a long a time or takes large memory space. There is decisive need of reconciliation of data periodically in cloud, for maintaining integrity. Reconciliation of data for correctness is referred as data integrity. To overcome data integrity challenges, multiple techniques are proposed under different systems and security models. All these have one or many challenges. Here, in this paper, will focus on challenges in data integrity techniques in comparative manner.*

Keywords: Cloud Computing Challenges, Data Integrity.

1. Introduction

Cloud computing in the terminology in which data keeping job is outsourced to service provider and user need not to worry about any data storing constraints like how and where data will be stored, security and integrity of data etc. External service provider is called CSP (Cloud Service Provider) and it is CSP, who insures user about all data storage related challenges. Storage space is provided over a network by CSP on pay per use bases. Cloud computing evolved out from GRID COMPUTING but it has made its own unique identity, so early. Essence of cloud computing is not only because of large scale industries but also small scale industries played vital role in it. It is because data generation is far outpacing data storage and it is very costly affair for small firms to purchase new hardware whenever additional data storage required. So it is favorable to outsource storage task to CSP. Storage outsourcing helps in reducing the costs of storage, maintenance and personnel. It also assures a reliable storage of important data by keeping multiple copies of the data thereby reducing the chance of losing data by hardware failures. Despite of all advantages of cloud computing, it has many interesting security concerns which need to be extensively investigated and resolved, so that cloud computing can become reliable solution to the problem of avoiding local storage of data.

2. Cloud computing challenges and data security

2.1 Cloud Computing Challenges

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]. As cloud provides many advantages, like other side of the coin, it also has certain challenges. Every technology have its own challenges so as cloud, some of these are as follows:

S. N.	Challenges	Description
1.	**Access**	If there is an unauthorized access to the data, the ability of altering data on the client side arises. Due to this we will be going to store corrupted data in cloud storage.
2.	**Availability**	The data must be available all the time for the clients (viz. High Availability) without having problems that affect the storage and lead to the client data loses.
3.	**Data Integrity**	System may collapse, when high amount of data shared between the computers and the servers, overloads the network.
4.	**Network Load**	Data correctness, legality and security most influencing on the cloud and have major lay on the service provider. Altogether these are termed as integrity.
5.	**Data Location**	The client does not know the actual place where data is stored or centered because it is distributed over many nodes. So it can lead to confusion about the actual data storage location.

Out of all challenges described above, data integrity is most critical to achieve. This paper describes challenges in achieving integrity using various methods.

2.2 Data Integrity

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 entrusted servers. In order to solve the problem of data integrity checking, many researchers have proposed different systems and security models. In terms of a database data integrity [8] refers to the process of ensuring that a database remains an accurate reflection of the universe of discourse it is modeling or representing. In other words there is a close correspondence between the facts stored in the database and the real world it models.

3. Current data integrity proving techniques

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. Working principle of PDP is:

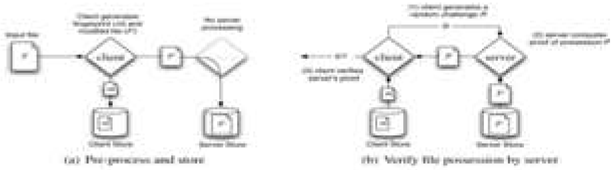


Fig: Principle of PDP [4]

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. 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. Challenges in PDP: Lack of error-correcting codes to address concerns of corruption. Lack of privacy preservation. No dynamic support.

3.2 Basic PDP Scheme based on MAC

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. Challenges in PDP based on MAC: 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 audit ability is not supported as the private keys are required for verification.

3.3 Scalable PDP

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. Challenges in Scalable PDP: 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

Dynamic PDP which is a collection of seven polynomial-time algorithms (KeyGen DPDP, PrepareUpdate DPDP, PerformUpdate DPDP, VerifyUpdate DPDP, GenChallengeDPDP ,ProveDPDP,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. Challenges in Dynamic PDP: It has some computational complexity. Not suitable for thin client. DPDP does not include provisions for robustness.

3.5 Basic Proof of Retrievability (PoR):

. The simplest Proof of retrievability (POR) scheme can be made using a keyed hash function hk (F). In this scheme the verifier, before archiving the data file F in the cloud storage, pre-computes the cryptographic hash of F using hk(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 hk (F). Challenges in Dynamic POR: 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.

3.5.1 Data placed on single server at cloud

Proof of retrievability for large files using 'sentinels'. The archive needs to access only a small portion of the file F. 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 challenges the prover (cloud archive) by specifying the positions of a collection of sentinels and asking the prover to return the associated sentinel values as shown in fig 2. Challenges in POR for large files: 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. This method works only with static data.

3.5.2 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. Challenges: 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.

3.5.3 HAIL

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. Challenges: 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.

3.5.4 POR Based on Selecting Random Bits in Data Blocks

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. Its stands on the fact that high probability of security can be achieved by encrypting fewer bits instead of encrypting the whole data. 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. Challenges: 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. Data integrity challenges

Comparative study of all Data integrity techniques is as:

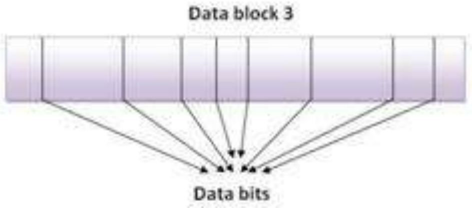
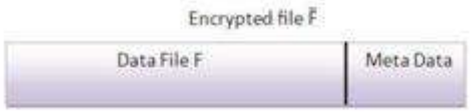


Fig. 3. A data block of the file F with random bits selected in it [1]



5. Conclusion

In the world of cloud computing the data integrity is most critical and burning issue. By considering the importance of data integrity, in this paper different existing techniques and their challenges 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. 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, like two sides of a coin. All papers were lack in proper data integrity mechanisms, supporting dynamic data operations, and by high resource and computation cost, which left open, requirement for future work on this topic.

Reference

[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]. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," in Proceedings of Secure Communication, 2008.

[5]. 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, 2007.

[8]. Yingyi Bu _ Bill Howe _ Magdalena Balazinska _ Michael D. Ernst "The HaLoop Approach to Large-Scale Iterative Data Analysis" VLDB 2010 paper "HaLoop: Efficient Iterative Data Processing on Large Clusters.

[9]. S. Vikram Phaneendra & E. Madhusudhan Reddy "Big Data- solutions for RDBMS problems-A Survey" In 12th IEEE/IFIP Network Operations & Management Symposium, (Osaka, Japan, Apr 19{23 2013).

[10]. Harshawardhan S. Bhosale1 , Prof. Devendra P. Gadekar2 "A Review Paper on Big Data and Hadoop" International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014 1 ISSN 2250-3153 www.ijsrp.org A

[11]. Rotsnarani Sethy , Mrutyunjaya Panda "Big Data Analysis using Hadoop: A Survey" international Journal of Advanced Research in Computer Science and Software Engineering Volume 5, Issue 7, July 2015

[12]. Albert Bifet "Mining Big Data in Real Time" Informatics 37 (2013) 15–20 DEC 2012.

[13]. R. Saraswathy, P. Priyadharshini, P. Sandeepa "HBase Cloud Research Architecture for Large Scale Image Processing" International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 12, December 2014

[14]. K. Arun, Dr. L. Jabasheela "Big Data: Review, Classification and Analysis Survey" International Journal of Innovative Research in Information Security (IJIRIS) ISSN: 2349-7017(O) Volume 1 Issue 3 (September 2014) ISSN: 2349-7009(P)

[15]. Bernice Purcell "The emergence of "big data" technology and analytics" Journal of Technology Research 2013. 1994 2/13/04