Data Integrity Verification in Cloud using Blockchain

Benedict J.N., Raghavi S.B., Selsiya C.L. and Sneha G.

Department of Computer Science and Engineering, Rajalakshmi Engineering College.

---***---

Abstract - In cloud environment, data integrity is required to secure user's data. The aim of the project is to ensure correctness of the data. This is done, when the user uploads the file in the cloud, the uploaded file gets encrypted using Base64 and MD5 algorithm and then it is split into random number of blocks using Dynamic Block Generation Algorithm and distributed among multiple servers in the cloud. When the blocks are created a block number and a unique signature will be generated for that particular block. The block details and the signature are stored in file allocation table (FAT) file system. The signature of these blocks will be periodically compared with the signature in the FAT file system. If an attacker corrupts the data in the cloud, the signature of that particular block is changed. This change is monitored by the Verifier Data Integrity Checking Algorithm and then the File recovery process is done automatically to retrieve the file.

Key Words**:** Data Integrity, Blockchain, Cloud storage, Data Verification

1.INTRODUCTION

Many organizations outsource their data, applications and business processes to the cloud, empowering them to achieve financial and technical benefits due to on-demand provisioning and pay-per-use pricing. However, organizations are still uncertain to adopt cloud services because of security, privacy, and reliability concerns regarding provisioned cloud services as well as doubts about trustworthiness of their cloud service provider [1]. Cloud Storage provides services to the users, in which the digital data is stored in logical pools to provide remote access over internet. This cloud storage reduces storage cost of the data but it has some critical security issues [15]. One such issues is Data integrity. Previously data were locally stored using physical hardware such as external hard drives, flash drive and CDs. There are chances for this physical hardware to fail or can be stolen. Therefore, to reduce this issue, the data is uploaded in the cloud. This uploaded data is put at high risk and hence data integrity is required to overcome the risk. Data integrity helps to maintain accuracy and consistency of the data.

In cloud computing, the resources are delivered as a service by a network. Data Integrity is a very crucial security issues [7]. User cannot control his own data. Now –a –days Cloud service has become a essential part of every system which lead to fastest evolving technology. Customers does not prefer to access their own data locally. Therefore, data integrity ensures that their data is stored and does not get lost. Public verification system needs auditor to perform verification periodically to detect data corruption [13]. The scheduled verification may be procrastinated due to network failure, errors in systems or request server cannot be procrastinated. Therefore, the auditors suffer from management of certification that involves problems with revocation of certificate, storage distribution and verification which are so expensive and difficult on practical world.

The section 2 describes about our proposed Dynamic Block Generation (DBG) framework. Section 3 describes the evaluation strategy adopted in our work. Section 4 describes related works and last Section 5 describe about conclusion and future scope.

2.DBG FRAMEWORK

In Multi cloud environment, remote data integrity checking is required to secure user's data. Authorized User will upload file to Cloud [11]. The uploaded file is split into multiple blocks using Dynamic Block Generation (DBG). Algorithm and stored in a multiple server in the cloud environment. File Allocation Table (FAT) File System has proper Indexing of the data and Metadata's for the different Chunks of the Cloud Storage. Here the auditor agrees to inspect logs, which are routinely created during monitoring operations by services providers. If attacker corrupts data in Multi cloud, the continuous auditing process helps the verifier to perform Block level and File level checking for remote data Integrity Checking using Verifiable Data Integrity Checking Algorithm. Cloud provides random blocks to Verify Integrity Checking. If the data gets corrupted, then during the process of checking, the file is recovered automatically by the Verifier. The architecture diagram for our proposed DBG framework is given in Fig -1.

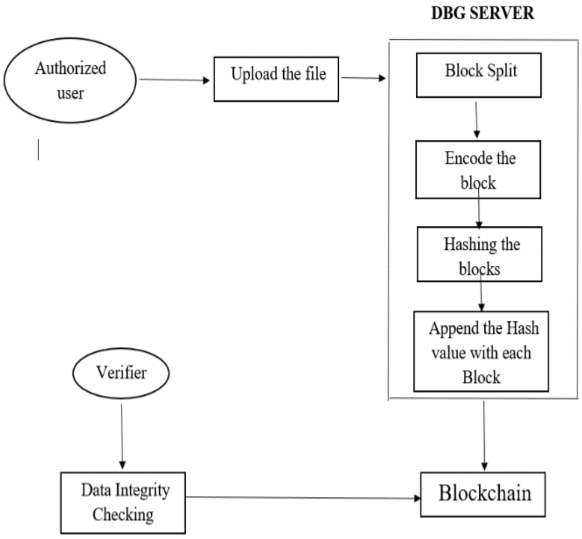

Fig -1: DBG Framework for Data Integrity Verification

The various modules used in our framework are- Server Configuration, Data Upload and Block Split, Data Integrity Checking and update in blockchain, File Recovery and Certificate Generation. The following subsections describe each of these modules in details.

2.1 Server Configuration

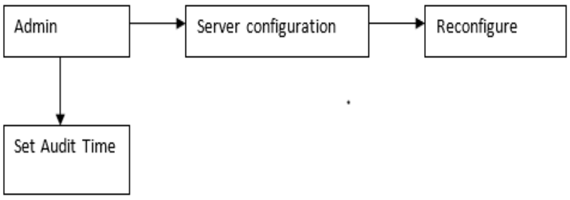

Fig -2: Server Configuration

Admin configures Multicloud server setup. Server IP Address and Port number is given by the admin for each Cloud. Now a Server Architecture is created for MultiCloud Storage. The admin can also reconfigure the old MultiCloud server setup. FAT file can be modified for old server setup. During the Data Integrity checking process the audit time will be set by the admin.

2.2 Data Upload and Block Split

The users provide their own personal information and the user has an initial level registration process at the web end. The information is stored in the database by the server. After Registration, user can upload files to the server. Uploaded files will be stored in a Server. This uploaded file is split into random number of blocks using Dynamic Block Generation Algorithm. Each block is appended with Signatures before storing the data in FATFS. Signature generated using MD5 Algorithm and the data gets encoded using for Base64 Algorithm.

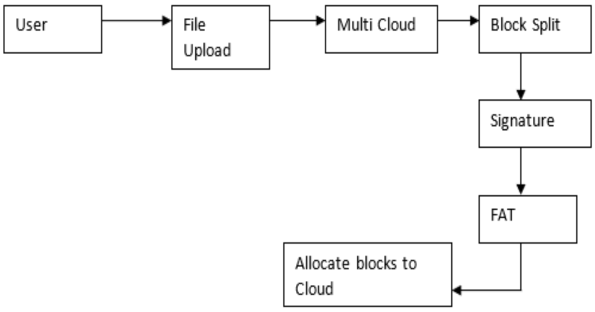

Fig -3: Data Upload and Block Split

2.3 Data Integrity Checking and Update in Blockchain

FATFS have proper Indexing of data and Metadata's for the different Chunks of the Data that is uploaded by User. Verifier performs Remote Integrity Checking on Cloud Data. Cloud allocates random combination of all the blocks to the Verifier, instead of the whole file is retrieved during integrity checking. This is to protect user privacy from a third party (Verifier). Verifiable Data Integrity Checking Algorithm is done in two steps: Block Checking and File Checking. In Block Checking step: Three signatures are generated for Block level Checking.

- A signature of a block retrieved from a FATFS
- A new signature is generated for block to be checked
- A Signature is retrieved from the block appended with the signature which is stored in the Cloud

The above three signatures are cross checked for Block level Integrity Checking. And the block contents are appended to verify with File level Integrity Checking. And update each and every auditing details in blockchain.

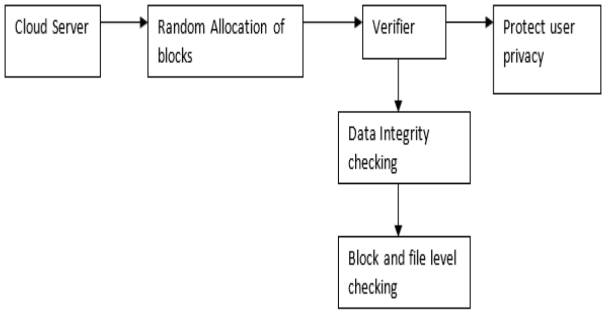

Fig -4: Block Level Integrity Checking

2.4 File Recovery and Certificate Generation

Attacker can corrupt data in any one of the cloud servers. On Data Integrity Checking done by the Verifier, Verifier informs Corrupted blocks to the Cloud. This corrupted block will be recovered by the verifier automatically. User can complaint to the Cloud if the user file get corrupted. Verifier doesn't perform checking on this file. Whenever user access file, Blocks will be reallocated dynamically to provide access confidentiality in cloud and FAT File System will get updated.

Auditor will monitor the cloud continuously and they provide the certificate based on the cloud performance. When new user join in the cloud they will read the certificate and then they can create an account in the cloud.

.

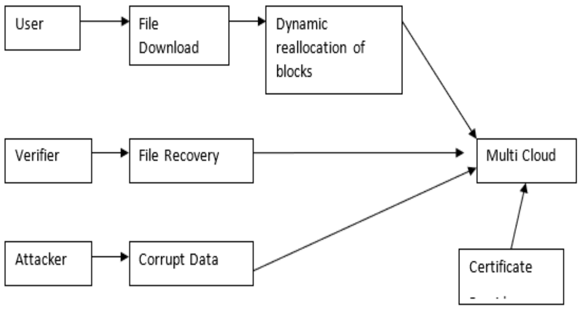

Fig -5: File Recovery in Multi Cloud

3. EVALUATION

Our experiments executed directly on a physical Intel Core i5 machine with 4 cores of 2.70 GHz speed and 4 GB RAM. We had used Windows 10 as host operating system and Apache Tomcat Server. Java language was used primarily to write the various routines to implement our framework. The benchmarking tool- IOZone was used to evaluate system performance. It is a benchmarking tool for file system that generates and measures variety of file operations. It performs 13 types of test covering read and writes operations on disk. Two tests namely read and write operations on the file system were performed. The tests were conducted with record size of 4 Kb, 64 Kb and 1 Mb. The file size used for each of these record sizes were 2 GB, 4 GB and 8 GB. The results of the cryptographic file system performance for disk read and write operations for varying file and record sizes have been recorded in Table-1 as shown below-

Table-1: File System Read/Write Performance

Record Size	**File Size GB**	**Without DBG**		**With DBG**	
		Read MB/s	**Write MB/s**	**Read MB/s**	**Write MB/s**
4 Kb	2	69.45	67.87	67.71	68.95
	4	69.21	68.05	69.2	68.62
	8	68.52	67.71	68.74	68.25
64 Kb	2	69.48	68.46	69.53	69.16
	4	69.21	68.68	63.53	68.44
	8	68.25	67.99	68.71	68.54
1 Mb	2	67.96	68.84	69.61	68.48
	4	67	68.92	68.65	68.79
	8	68	67.09	68.69	67.34

The disk read operation is slower with DBG than without it. However, the disk read in DBG performs activities like block splitting and computing hash code. The latency due to these task results in extra 1-2 ms which is quite reasonable. The disk read operation was performed on file with varying sizes and varying record size. Some of the values of our experimental data is shown in Table-1.

The results of disk write operation is depicted in Figure-3. The record size used was 1 Mb and file sizes used were 2 GB, 4 GB and 8 GB. The write operation is faster with DBG for file size of 8 GB. Without DBG, the disk write operation executes faster than DBG. It can be observed from the Fig-6 that the overhead due to DBG is marginally lesser for 2GB and 4GB file sizes.

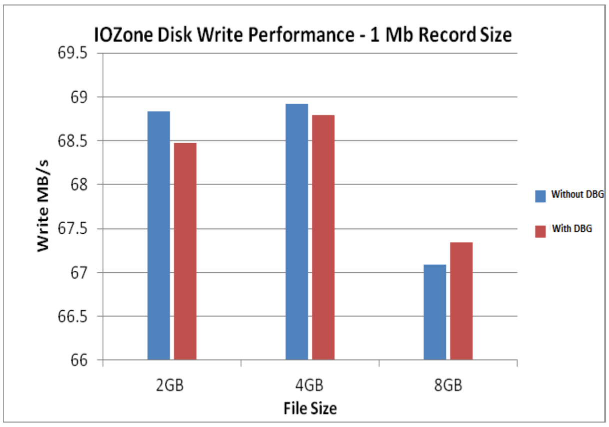

Fig -6: Disk Write Performance

4. RELATED WORK

The cloud storage security is very vital to preserving the privacy and integrity of the data.

4.1 Blockchain

Zhong et al. [4] describe a secure versatile light payment system that was based on blockchain for future systems. Shacham et al. [14] describe proof of retrievability that is used in blockchain applications like bitcoins.

4.2 Cloud Storage

Zhang et al. [10] proposed a identity based key exposure resilient cloud storage public auditing scheme and used the mathematical concept of lattices. Wang et al. [9] described a public verification based dynamic data storage security for cloud storage environment. Yang et al. [15] describe a secure auditing protocol for cloud storage. Zhang et al. [13] describe a public verification of data integrity in cloud storage.

4.3 Authentication

Wang et al. [11] describe pre-authentication approach to proxy re-encryption in big data environment. Yang et al. [8] describe centric networks with content-based encryption. Xu et al. [5] describe access control over encrypted spatial data.

5. CONCLUSIONS

The storage of files and metadata in a Cloud environment requires data security. In this paper, we have proposed a data integrity verification scheme in cloud by leveraging the concepts of blockchain. The Dynamic Block Generation (DBG) algorithm splits the blocks and applies various encoding and hashing techniques to obtain a unique signature for the data block. The signature of the data blocks and other metadata are stored in the blockchain. An attacker can tamper with the data blocks but our proposed framework can detect the data integrity by referring the signature stored in blockchain. Our system provides the strongest security guarantee compared with existing schemes. In future work, we will try to reduce the overhead in processing of the data blocks for storing in the blockchain.

REFERENCES

[1] J. Yu, K. Wang, D. Zeng, C. Zhu, and S. Guo, “Privacy-preserving data aggregation computing in cyber-physical social systems,” ACM Transactions on Cyber-Physical Systems, vol. 3, no. 1, p. 8, 2018.

[2] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, “Querying in internet of things with privacy preserving: Challenges, solutions and opportunities,” IEEE Network, vol. 32, no. 6, pp. 144–151, 2018.

[3] J. Li, H. Ye,W.Wang,W. Lou, Y. T. Hou, J. Liu, and R. Lu, “Efficient and secure outsourcing of differentially private data publication,” in Proc. ESORICS, 2018, pp. 187–206.

[4] L. Zhong, Q. Wu, J. Xie, J. Li, and B. Qin, “A secure versatile light payment system based on blockchain,” Future Generation Computer Systems, vol. 93, pp. 327–337, 2019.

[5] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, “Enabling efficient and geometric range query with access control over encrypted spatial data,” IEEE Trans. Information Forensics and Security, vol. 14, no. 4, pp. 870–885, 2019.

[6] K. Yang, K. Zhang, X. Jia, M. A. Hasan, and X. Shen, “Privacypreserving attribute-keyword based data publish-subscribe service on cloud platforms,” Information Sciences, vol. 387, pp. 116– 131, 2017.

[7] W. Shen, B. Yin, X. Cao, Y. Cheng, and X. Shen, “A distributed secure outsourcing scheme for solving linear algebraic equations in ad hoc clouds,” IEEE Trans. Cloud Computing, to appear, doi: 10.1109/TCC.2016.2647718.

[8] H. Yang, X. Wang, C. Yang, X. Cong, and Y. Zhang, “Securing -centric networks with content-based encryption,” Journal of Network and Computer Applications, vol. 128, pp. 21–32, 2019.

[9] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,” in Proc. of ESORICS, 2009, pp. 355–370.

[10] X. Zhang, H. Wang, and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," Information Sciences, vol. 472, pp. 223–234, 2018.

[11] K.Wang, J. Yu, X. Liu, and S. Guo, "A pre-authentication approach to proxy re-encryption in big data context," IEEE Transactions on Big Data, 2017, to appear, doi. 10.1109/TBDATA.2017.2702176.

[12] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. Shen, "Providing task allocation and secure deduplication for mobile crowdsensing via fog computing," IEEE Transactions on Dependable and Secure Computing, to appear, doi. 10.1109/TDSC.2018.2791432.

[13] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, and X. Zhang, "Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation," IEEE Trans. Information Forensics and Security, vol. 12, no. 3, pp. 676–688, 2017.

[14] H. Shacham and B. Waters, "Compact proofs of retrievability," of Cryptology, vol. 26, no. 3, pp. 442–483, 2013.

15. K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel Distributed Systems, vol. 24, no. 9, pp. 1717–1726, 2013.