Byzantine Failure against Colluding Attacks in Cloud Data

Noman Mazher[1] and Shafiq Hussain[2]

[1]Affiliation not available
[2]Chenab University of Information Technology

July 11, 2022

Byzantine Failure against Colluding Attacks in Cloud Data

Shafiq hussain,

Chenab University of Information Technology

ABSTRACT

Cloud computing is the next stage in evolution of the internet, which provides large amounts of computing and storage to customers provisioned as a service over the internet. However, cloud computing faces so many security challenges due to the possible compromise or byzantine failures. In this paper, we focus on Ensuring data storage security in cloud computing, which is an important aspect of Quality of Service (QoS). We propose an effective and flexible distribution verification protocol to address data storage security in cloud computing. In this protocol, we rely on erasure code for the availability, reliability of data and utilize token precomputation using Sobol Sequence to verify the integrity of erasure coded data rather than Pseudorandom Data in existing systems. Unlike prior works, our scheme provides more security to user data stored in cloud computing. The performance analysis shows that our scheme is more secure than existing systems against Byzantine failure, unauthorized data modification attacks, and even cloud server colluding attacks.
Keywords: Cloud Computing; Data Storage Security; Availability; Reliability; Integrity; Pseudorandom Data; Sobol Sequence.

I. INTRODUCTION

Cloud computing is internet based computing. It dynamically delivers everything as a service over the internet based on user demand, such as network, operating system, storage, hardware, software, and resources. These services are classified into three types: Infrastructure as a Service (IaaS), Platform as a Service (PaaS)[1] and Software as a Service (SaaS)[2]. Cloud computing is deployed as three models such as Public, Private, and Hybrid cloudsCloud data storage (Storage as a Service) is an important service of cloud computing referred as Infrastructure as a Service (IaaS).Data storage in the cloud offers so many benefits to users: 1)It provides unlimited data storage space for storinguser's data.2) Users can access the data from the cloud provider via the internet anywhere in the world, not on a single machine. 3)We do not buy any storage device for storing our data and have no responsibility for local machines to maintain data[3]. Amazon's Elastic Compute Cloud and Amazon Simple Storage Service (S3) are well known examples of cloud data storage. On the other side along with these benefits' cloud computing faces big challenge i.e. data storage security problem, which is an important aspect of Quality of Service (QoS). Once a user puts data on the cloud rather than locally, he has no control over it i.e. unauthorized users could modify user's data or destroy it and even cloud server collude attacks. Cloud users are mostly worried about the security and reliability of their data in the cloud[4].

Amazon's S3 is such a good example.

The following research works have highlighted the importance of ensuring integrity and availability of outsourced data. POR schemes efficiently verifies the server for outsourced data storage correctness. G.Ateniese et al. Introduced a Provable Data Possession (PDP) scheme; it efficiently detects the large number of file corruptions. In addition to the PDP scheme, R.D.Pietro et al. introduced a Scalable Data Possession (SDP) to verify the integrity of remotely stored data dynamically. However, all the schemes could't address all security threats in cloud data storage, since they work only for a single server.[5] Is a practical mutation testing.

In the complementary approach, researchers also proposed a distributed protocol to verify the data storage security on multiple servers, but these schemes will not address all security issues of cloud data storage.

Recently, Wang et al. Proposed a homomorphic distributed verification protocol to ensure data storage security in cloud computing using Pseudorandom Data. Their scheme achieves the storage correctness as well as identifies misbehaving servers. However, this scheme was not providing full protection for user storage data in cloud computing, because Pseudorandom Data does not cover the entire data while verifying the cloud

servers for data storage correctness i.e. some data corruptions may be missing.

In this paper, we propose a distributed verification protocol to guarantee the data storage security in cloud computing. This scheme uses Reed-Solomon erasure code for the availability and reliability of data and utilizes the token pre computation using Sobol Sequence rather than Pseudorandom Data to check the integrity of erasure coded data in cloud data storage. Our method achieves the integrity of storage correctness guarantee and identification of misbehaving servers i.e. whenever data modifications or deletions have been detected during the storage correctness verification across cloud servers, this method should guarantee the identification of misbehaving servers. Compared to previous methods, our method should provide more security to users' data stored in cloud computing.

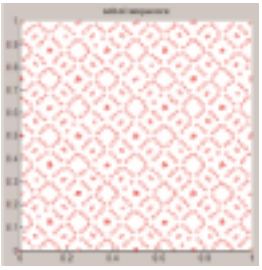

Figure.1 Comparison of Pseudorandom Data and Sobol Sequence.

The rest of the paper is organized as follows: Section **II** describes the related work, section **III** introduces problem definition. Section **IV** provides a detailed description about our proposed model. Section **V** gives performance evaluation of our model and Section. **VI** give the conclusion of our work.

II. RELATED WORK

Ari Juels et al. described a Proofs' of Retrievability (POR) model to ensure the outsourced data security. This scheme efficiently detects data corruptions and achieves the guarantee of file retrieval. Shacham et al. introduced a new model of POR, which enables unlimited no of queries for public verifiability with less overhead. Kennadi D et al. proposed a theoretical framework for the design of POR. It improves the JK and SW models. All the schemes produce weak security, because they work only for single server. Later, in their subsequent work Kennadi Brow et al. introduced a HAIL protocol, which extended the POR schemes on multiple servers. HAIL achieves the integrity and availability of data in cloud. However, this protocol will not address all the data security threats. Ateniese et al. described a Provable Data Possession (PDP) to verify the integrity of outsourced data; it detects the large fraction of file corruption, but no guarantee of file retrieval. In their subsequent work R.D.Pietro et al. proposed a Scalable Data Possession (SDP), this scheme overcomes all problems in the PDP scheme , but this scheme also works only for a single server. Later, Curtomola et al. described a Multiple Replica-Provable Data Possession (MR-PDP), which is an extension of PDP to ensure data availability and reliability of outsourced data on multiple servers. Compared to PDP, it requires only a single set of tags to challenge servers.

In other existing works, Mohsen et al. Proposed a new location based authentaction system. In their work they used mobile coupon system. In this service, users receive mobile coupons based on their location information from nearby stores[6] . in [7] author presented a framework named Hidden fear, that based on effectiveness of social media. .Internet based backup technique to store client data. It protects data from free riders and disrupter attacks. However, it can't detect data modifications or data changes. Schwarz et al. presents a new model to check the security of data in distributed storage systems. It verifies the large amount of data with minimum bandwidth in distributed storage systems. However, in this scheme the server can access linear no of file blocks per each challenge. Filho et al. Describes a secured hash function to prevent cheating in a P2P system, however it is unusable when a file is large. Shah et al. Proposed a new scheme, which allows Third Party Auditor (TPA) to keep on-line storage honesty with hash values computed by user on encrypted file. However, this scheme works only for encrypted files.

Recently, Mohsen ahmadi a homomorphic distributed verification scheme using Pseudorandom Data to verify the storage correctness of user data in the cloud[8]. This scheme achieves the guarantee of data availability, reliability, and integrity. However, this scheme was also not providing full protection to user data in cloud

computing, since Pseudorandom Data would not cover the entire data.

Sobol Sequence is an example of quasi-random low discrepancy sequences. This sequence was first introduced by I.M.Sobol in 1967. The numbers are generated sequentially to fill the larger "gaps" in the Pseudorandomdata.

III. PROBLEM DEFINITION

The network representative architecture for cloud data storage, which contains three parts as shown in Figure 2, viz Users, Cloud Service Provider (CSP) and Third Party Auditor (TPA). Users Cloud Service Provider.

Figure. 2 Cloud Data Storage Architecture

as shown in Figure 2, the brief descriptions of these parts as follows:

Users: - Users who have data to be stored and interact with the cloud service provider (CSP) to manage their data on the cloud. They are typically, desktop computers, laptops, tablet computers, mobile phones, etc.

Cloud Service Provider (CSP):- Cloud service provider (CSP) has major resources and expertise in building and managing distributed cloud storage servers. A CSP offers storage or software services to user's available via the Internet.

Third Parity Auditor (TPA):- An optional TPA, who has expertise and capabilities that users may not have, is monitoring the risk of cloud data storage services on behalf of users.

To address the data storage security in cloud computing, we propose a distributed verification protocol using Sobol Sequence. Our method allows a user to verify the cloud server whether data is stored correctly or not and identify the misbehaving servers without having a local copy of data. In case users do not have time to verify their data in the cloud, they can assign this job to a trusted Third party Auditor (TPA).

Our method's main goal is to provide more security to user data stored in cloud computing i.e. guarantee the availability, reliability, and integrity of data and Users have to perform storage correctness checks with minimum overhead.

IV. PERFORMANCE EVALUATION

In this section, we analyze the performance evaluation of File Distribution and Token Pre-Computation.

A. File Distribution

We implemented a file encoding for data availability and reliability. Our experiment is conducted using java 1.6 on a system with a core 2 duo processor running at 2.80 GHz, 4GB of RAM and 3GB of hard disk. We are considered 2 parameters for the (m+n, n) Reed-Solomon Encoding over Galois Field GF (2^8).

B. Token Pre-Computation

Like previous schemes, in our scheme the number of verification tokens t is limited and decided before file distribution. Our method overcomes this problem by deciding the number of tokens t in dynamically. For example, when t is

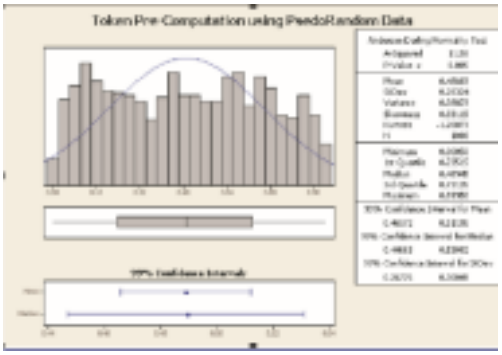

Figure 3: Token Pre-Computation using Pseudorandom Data

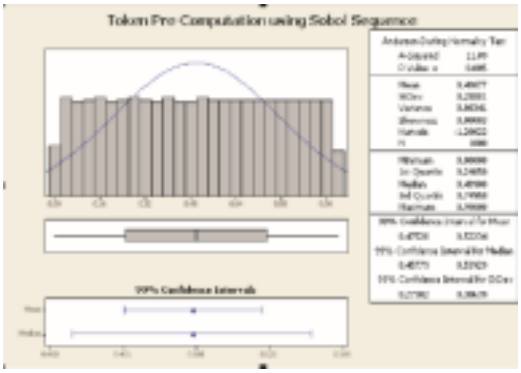

Figure 4: Token Pre-Computation using Sobol Sequence

selected to be 3650 and 5475, the data file can be verified every day for next 10 and 15 years. Practically, users select r= 460 indices, in order to identify misbehaving servers with high probability. If we use Pseudorandom Data for token Pre-Computation, it will not detect all data corruptions in cloud computing, because this method does not cover the entire data while verifying the cloud servers for data storage correctness. Compare to previous method, our method should detect all data corruptions in cloud computing with high probability. Figure 3 & 4 shows that our method is **more secure** than the existing method using Pseudorandom Data.

V. CONCLUSION

In this paper, we proposed a more effective and flexible distributed verification scheme to address the data storage security issue in cloud computing. We rely on ReedSolomon erasure code in file distribution to guarantee the availability and reliability of data and utilize token precomputation using Sobol Sequence to check integrity of erasure coded data in cloud data storage. Our method achieves the availability, reliability and integrity of erasure coded data and simultaneously identifies misbehaving servers i.e. whenever data corruptions will occur during the storage correctness verification, our method should identify the misbehaving servers. Through detailed performance analysis, we show that our scheme should provide more security to user's data in cloud computing against Byzantine failure, unauthorized data modification attacks and even server colluding attacks.

VI. REFERENCES

[1] B. Ahmed, A. W. Malik, T. Hafeez, and N. Ahmed, "Services and simulation frameworks for vehicular cloud computing: a contemporary survey," *EURASIP Journal on Wireless Communications and Networking,* vol. 2019, no. 1, pp. 1-21, 2019.

[2] "47_DATA DYNAMICS_H.pdf." http://ijircce.com/upload/2013/april/47_DATA%20DYNAMICS_H.pdf (accessed.

[3] G. K. Sriram, "Edge Computing vs. Cloud Computing an Overview of Big Data Challenges and Opportunities for Large Enterprises," *International Research Journal of Modernization in Engineering Technology,* 2022.

[4] M. Ahmadi, K. Leach, R. Dougherty, S. Forrest, and W. Weimer, "Mimosa: Reducing malware analysis overhead with coverings," *arXiv preprint arXiv:2101.07328,* 2021.

[5] M. Ahmadi, P. Kiaei, and N. Emamdoost, "SN4KE: Practical Mutation Testing at Binary Level," *arXiv preprint arXiv:2102.05709,* 2021.

[6] M. Ahmadi and B. S. Ghahfarokhi, "Preserving privacy in location based mobile coupon systems using anonymous authentication scheme," in *2016 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, 2016: IEEE, pp. 60-65.

[7] M. Ahmadi, "Hidden fear: Evaluating the effectiveness of messages on social media," Arizona State University, 2020.

[8] P. Kiaei, C.-B. Breunesse, M. Ahmadi, P. Schaumont, and J. Van Woudenberg, "Rewrite to reinforce: Rewriting the binary to apply countermeasures against fault injection," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, 2021: IEEE, pp. 319-324.