STORAGE SECURITY IN CLOUD COMPUTING: DATA AUDITING PROTOCOLS

Mihai Iacov
Faculty of Automatic Control and Computer Science
Politehnica University of Bucharest
mihai.iacov@stud.acs.upb.ro

Andrei-Alexandru Brebu
Faculty of Automatic Control and Computer Science
Politehnica University of Bucharest
andrei.brebu@stud.acs.upb.ro

Emil Simion
Faculty of Applied Sciences
Politehnica University of Bucharest
emil.simion@upb.ro

ABSTRACT

Cloud computing has emerged as a necessity for hosting data on cloud servers so that information can be accessed and shared remotely. It was quickly adopted because it provides quality of service for various remotely available, easy-to-configure, and easy-to- use products, such as IaaS (Infrastructure as a Service) or PaaS (Platform as a Service). However, this new paradigm of data hosting brings new challenges. Some of the challenges related to the issue of security require independent audit services to verify the integrity of cloud-hosted data. With many end users and companies moving from on-premise to cloud models for their business, cloud data security is a critical concept that needs to be managed. First, we identify security requirements. Second, we look at potential solutions to ensure data integrity in cloud storage. Last, we propose a data auditing solution that can be used to detect corrupt data or file anomalies in the storage system.

***K*eywords** Cloud Storage · Storage Security · Data Integrity · Blockchain · InterPlanetary File System

1 Introduction

Cloud computing has emerged as a consequence on the development of information technology. Huge amounts of data, new artificial intelligence paradigms, and interconnectivity between devices have led academia and industry to focus on computing and storage. This was the right environment for Cloud Computing to address scalability and availability issues very efficiently[1].

Various cloud service providers have understood the opportunity and come up with solutions that many companies have adopted because of their low cost and other benefits, regardless of measures to prevent potential problems [2]. For example, some cloud service providers may sometimes be dishonest when they provide us with data integrity, but servers lose blocks of files that were rarely accessed or not accessed to save storage space. Others ensure that data is available to us at all times, but they are unable to effectively handle unforeseen server and connection issues.

Therefore, issues such as data integrity, exposure, or availability have caused customers to worry about the security of their data. Many examples confirmed customer fears: the failure of Amazon S3 and the disruption of Amazon EC2 services, the deletion of emails in Gmail, the Sidekick cloud disaster. Therefore, issues such as data integrity, exposure or availability have led customers to worry about the security of their data. Cloud service providers are aware of this and are focusing on resolving these issues of trust and security.

The evidence and access to data have thus become challenges. Therefore, remote data audit protocols (RDAs) have been developed that can perform these validations efficiently using different logical methods. We will further explore the current solutions and try to intuit the direction of this trend of Cloud Computing considering the evolution of decentralization concepts.

2 Preliminary knowledge

We want to define a set of popular concepts and algorithms which are employed in cryptography, when working with data integrity and auditing. To ensure that the data is verifiable, the following algorithms are commonly used.

2.1 KeyGen

KeyGen algorithm stands for Key Generation, a random process which accepts a security parameter (lambda) and returns a set of private and public keys.

$$KeyGen(F, \lambda) \rightarrow (PrK, PubK) \tag{1}$$

2.2 ProbGen

ProbGen algorithm stands for Problem Generation, it is an algorithm that generates a random problem (might also be found as **genChal**), it accepts an input(i) and a public key, and it returns a verification key and a temporary challenge token(σ) which is used further for computation.

$$ProbGen(i, PubK) \rightarrow (VerK, \sigma_i) \tag{2}$$

2.3 Compute

The computation function for generating the proof (also seen as **ProofGen**), which is consistent across repeated calls, accepts the challenge value and a private key, and returns the output proof value (σ_o).

$$Compute(\sigma_i, PrK) \rightarrow \sigma_o \tag{3}$$

2.4 VerifyProof

The verification algorithm, which is consistent across repeated calls, accepts the proof value (σ_o), a public key, and a verification key, and it returns a confirmation flag value for whether the proof is valid or not (true or false).

$$VerifyProof(\sigma_o, PubK, VerK) \rightarrow Verdict \tag{4}$$

3 Related Works

3.1 Data integrity: remote auditing

Data integrity is an important topic in cloud storage and security. The more general concept of remote data auditing or remote data checking has been used in the past in various contexts, but more recently it was also used in the cloud context, as described by Ren et al. [3] in 2015, when defining an MV-PDP scheme for public cloud (Mutual Verifiable Provable Data Possession). The paper introduced a homomorphic authenticator scheme based on DH key pairs (Diffie-Hellman), in a system model composed of three entities: the client, the cloud storage server, and the private verifier. This security model proposed contains algorithms for key generation (KeyGen), tag generation (TagGen), challenge generation (GenChal), possession proof generation (GenProof), and proof verification (VerifyProof).

A more recent approach by Fen et al. [4] proposes a similar remote data auditing protocol to address this issue of data integrity in clouds, with slight differences added to the problem. The system model described in this paper contains four entities instead of three. There's an almost exact 1-to-1 mapping between the first three entities, with the key difference that the TPA (Third Party Auditor) in this model is a potentially untrusted entity, as opposed to the safe and trusted PV (Private Verifier) introduced in the previous model by Ren et al. [3]. The fourth entity from the system model is a common platform which serves as a proxy/delegate between the cloud server and the TPA. All the authentications or verification proofs are marked as events in the common platform records, where the cloud server can intercept the activities to verify the integrity of the TPA's records. In this system model interaction, the auditor's goal is to verify the integrity of cloud storage data, while the cloud server's goal is to ensure that the auditor can be trusted and offers correct information. By using bilinear mapping in the algorithms for verifying possession proofs, this system allows for a robust checking of integrity with a variate setup for the auditor entity. It is also important to note that checking for the auditor's correctness adds a computation overhead and using bilinear mappings to achieve this purpose is more expensive than the previous approach which assumed the auditor as always trusted by default.

3.2 Collaborative auditing with blockchain

Proposal - the system model approach adopted by Fen et al. [4] to assume that third-party auditors are not trusted implicitly presents a good starting point for exploring a different paradigm for reaching consensus and verifying integrity in distributed networks. The blockchain technology can be a good candidate for the presented problem. A blockchain is a distributed ledger, a database of records shared in a distributed network, which can be used to ensure consistency in a decentralized environment. There are many consensus protocols [5] used across blockchain solutions, each one of them having different advantages and disadvantages. For example, according to Nguyen et al. [6], the Proof-of-Stake consensus is a better choice than Proof-of-Work when the system's focus is on energy efficiency, reducing vulnerabilities to security threats.

The remote data auditing problem can be studied together with blockchain technologies. Blockchain can help in improving the trust the users have in their system. As described by Pei Huang et. al. [7], the problem of introducing a 3rd party auditor entity is a public auditing method which enhances the security and trust in the integrity of the cloud storage, but there is a preliminary assumption required here - the fact that a centralized third party is neutral in the auditing process. To avoid this assumption which could lead to problems, they adressed the problem with a blockchain-basedcollaborative auditing framework. Once again, the key feature obtained from this architecture is preserving the integrity of data stored in the cloud.

The blockchain-based auditing employs multiple nodes that verify records for integrity and need to reach a consensus about its validity. This process also reduces the direct resource-cost from the cloud storage, since it's deferred to the auditors. The cloud storage owners (or simply data owners) and the auditors are ranked by "credit-scores", a score which signifies trust in the system or entity, based on previous behavior. This way, the auditors can expose bad data owners that attempt to wrongfully pass audits by forging fake data entries to hide the corrupt data, or prevent cases where the data is deleted without the user's consent. The effective steps of the auditing process itself are similar with the ones introduced in previous studies [3], but the distributed ledger accounting adds the novelty, by adding an election process to the distributed consensus, where scores are factored in.

3.3 Modern file systems with blockchain

A more recent study from 2021 [8] also proposes a similar architecture for combining the remote auditing process with a blockchain ledger consensus, where they also add the concepts of a private-blockchain with authorized nodes and re-use cummon consensus algorithms, such as RAFT. They also summarize the contributions from other research efforts so far to the problem of the auditing process by using blockchain. One of the mentioned contributions in their list is auditing based on data stored through InterPlanetary File System (IPFS), at the cost of increasing the complexity of the system. We will later try to showcase how IPFS can be used to provide a storage service with increased availability and integrity.

In the same paper, Regueiro et. al. [8] mention some architecture requirements fo this kind of systems. Aside from the common CIA (Confidentiality, Integrity, Availability), they also list REST APIs as an expected requirement. In other words, it is expected from the system to provide an interface that is easy to work with. In our research, we also make the proposal to showcase an intuitive API to the IPFS blockchain-based system.

4 Research Proposal

The recent growth of cloud computing comes together with growth in other fields, such as big data, computer networks, IoT, machine learning, and a new concept of internet protocol that we aim to explore. As an alternative to the popular HTTP protocol for accessing content online, the IPFS, combined with the blockchain framework, brings a potential solution for data auditing protocols.

IPFS [9] (InterPlanetary File System) is a network protocol and a peer-to-peer network for sharing data in a distributed file system. It also has a web platform and APIs for developers. Uses hash-trees (Git versioning / blockchain principles) to represent files, versioning, and records. The system is decentralized and does not require trust between nodes.

By using hash-trees, the IPFS system provides increased reliability, with its ingtegrity and availability being reinforced by distributing the data across multiple nodes. The integrity is also enhanced through the nature of the blocks - the body of the block is immutable, which means every block in the blockchain can only be appended or removed. Once a file is uploaded in a block body, we cannot temper with it by attempting to update or corrupt the data content, since the API will not allow updating content, since the "updated" file can only be added as a new version in a separate block. The integrity of the data is our main interest here, which we want to study in our project setup by testing how we can recover a "lost" block of data on a node, by asking peer nodes to share their knowledge.

The IPFS API is intuitive [1], they provide a minimal REST API for working with data (reading, writing files), but it is also accompanied by management endpoints which can be used to study the ledger, the block DAGs, and other internals of the underlying blockchain framework, which we want to test out. IPFS has also been used in the past for implementing distributed social networks [10][11], to leverage its security advantages. We will also attempt to benchmark the performances of common IPFS operations in such a setup (e.g.: listing available files, downloading files, uploading files).

The InfuraAPI [2] is an opensource framework which allows developers to connect to the Ethereum blockchain network without having to run a full node and it also provides access to storage via IPFS APIs.

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, apr 2010.

[2] Vishal Kher and Yongdae Kim. Securing distributed storage: Challenges, techniques, and systems. In *Proceedings of the 2005 ACM Workshop on Storage Security and Survivability*, StorageSS '05, page 9–25, New York, NY, USA, 2005. Association for Computing Machinery.

[3] Yong-Jun Ren, Jian Shen, Jin Wang, Jin Han, and Sung-Young Lee. Mutual verifiable provable data auditing in public cloud storage. , 16(2):317–323, 2015.

[4] Bin Feng, Xinzhu Ma, Cheng Guo, Hui Shi, Zhangjie Fu, and Tie Qiu. An efficient protocol with bidirectional verification for storage security in cloud computing. *IEEE Access*, 4:7899–7911, 2016.

[5] Sarah Bouraga. A taxonomy of blockchain consensus protocols: A survey and classification framework. *Expert Systems with Applications*, 168:114384, 2021.

[6] Cong T Nguyen, Dinh Thai Hoang, Diep N Nguyen, Dusit Niyato, Huynh Tuong Nguyen, and Eryk Dutkiewicz. Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities. *IEEE Access*, 7:85727–85745, 2019.

[7] Pei Huang, Kai Fan, Hanzhe Yang, Kuan Zhang, Hui Li, and Yintang Yang. A collaborative auditing blockchain for trustworthy data integrity in cloud storage system. *IEEE Access*, 8:94780–94794, 2020.

[8] Cristina Regueiro, Iñaki Seco, Iván Gutiérrez-Agüero, Borja Urquizu, and Jason Mansell. A blockchain-based audit trail mechanism: Design and implementation. *Algorithms*, 14(12):341, 2021.

[9] Juan Benet. Ipfs - content addressed, versioned, p2p file system. *ArXiv*, abs/1407.3561, 2014.

[10] Quanqing Xu, Zhiwen Song, Rick Siow Mong Goh, and Yongjun Li. Building an ethereum and ipfs-based decentralized social network system. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1–6, 2018.

[11] Hrishikesh Bawane, Tanuja Shinde, Abhishek Kadam, Yash Budukh, and Pooja Mundhe. Ethegram-an ethereum and ipfs-based decentralized social network system. 2020.

[1] IPFS API - https://docs.ipfs.io/reference/http/api/getting-started

[2] Infura API - https://infura.io/docs