Data integrity verification with blockchain

Smt. Kundanben Dinsha Patel Department of Information Technology,
Chandubhai S. Patel Institute of Technology, Charusat University, Changa,Gujarat.
19IT109 SHIVAM PATEL-19IT113 VEDANT PATEL

Abstract: *Blockchain has shown promise in a variety of application areas in recent years, however it faces technical hurdles like as security and scalability. As the number of transactions increases, blockchains face data storage challenges. Cross-chain solutions are connecting multiple blockchains and relieving data storage load. However, previous research usually targets on the technical implementation of cross-chain interaction, with little extensive study on consistency concerns such as data integration verification of cross-chain interaction.*

INTRODUCTION

Blockchain is steadily expanding from small scalable applications in numerous areas, indicating a promising future. The increased quantity of transactions on the blockchain puts a burden on blockchain data storage, reducing the effectivity of on-chain queries and calculations. To alleviate the burden of data storage, one may store data on many blockchains. When the blockchain requires data from various chains, the data can be across chains. Cross chain realizes chain interaction by connecting independent blockchain networks, relieving blockchain data storage pressure of blockchain. However, the majority of cross-chain applications concentrate on asset trading rather than information call. Because each blockchain has its own internal security algorithm and does not take part in other blockchain consensus processes, it is difficult for blockchains to determine the integrity of the received data. As a result, steps should be made to ensure the integrity of data in cross-chain interactions.

Currently, data integrity verification techniques are largely seen in cloud storage. Users send regional data to the cloud and access data casually over the Internet to alleviate the strain of data storage management. Without acquiring entire data details, users can assess whether the outsourced data is well kept by the cloud. Traditional verification systems, on the other hand, check the integrity of the information saved by the information receiver, which does not applicable to cross-chain scenarios. We must audit the characteristic of the chain that transmits data in cross-chain interactions in cross-chain scenarios. So, typical integrity verification techniques in this research maynot be easily applies to cross-chain circumstances. In past researches on integrity verification, third-party checkers were frequently used for evaluating the data integrity on an untrustable cloud. However, the employment of third-party checkers in cross-chain circumstances impairs the blockchain system's decentralization, and there is also the chance that the auditors collaborate with the cloud to provide a biased verification result in order to fool consumers. As a result, several academics have turned to blockchain for auditing. The laws and contracts for auditing blockchains can be transformed into simple and deterministic code-based rules that are automatically executed by the underlying blockchain network, ensuring that the blockchain always outputs a fair audit result and maintaining the decentralization of the entire scheme in cross-chain interaction.

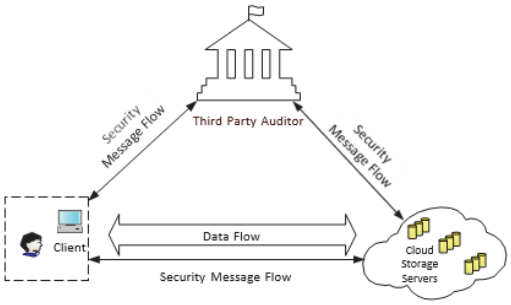
RELATED WORK

Here are the related works:

1. *Cross-chain Technology*

   The separation between blockchains severely limits blockchain application and development. The study of the interactions between chains is in demand. Notary schemes, sidechains/relays, and hash-locking are now in trend cross-chain technologies. However, the majority of cross-chain applications focus on asset transfer other than data call.

2. *Data Integrity Verification in the Traditional Cloud Storage*



   Traditional data based authentication procedures are classified into two groups. The first is Provable Data Possession (PDP), and the other is Proofs of Retrievability (POR). When the data integrity is compromised, PDP can swiftly check whether the data saved in the cloud is safe, whereas POR can recover the damaged data. Deswarte et al. illustrated the fundamental PDP authentication algorithm. Before uploading his own data, the user calculates the Message Authentication Codes (MAC) sense of the information using the Hash-based Authentication Code (HMAC) and saves it locally.

3. *Data Integrity technologies for verification*

   Currently, data integrity verification methods are largely seen in cloud storage. In 2007, Ateniese et al. demonstrated checkable data posession (PDP), which uses the RSA signature's monomial verification labels to validate the file's data integrity. PDP can detect whether distant data is destroyed, but it cannot guarantee availability. In the same year, Juels et al. investigated how to bring back defective data and presented lookout-based proof of retrievability (POR), which is not only way to verify data integrity but also partially bring back corrupted data. Wang et al. launched the public verification process in 2009, using a third-party auditor to ensure data accuracy. Wang et al. (2010) addressed the issue of privacy protection by employing random mask technology to successfully cover sensitive information in the evidence supplied by the cloud server. In this manner, auditors are unable to identify any data details. Nayak et al. (2018) developed a privacy-preserving verifiable data possession system that supports multiple owners, data dynamics, and batch verification.

   The above studies' main concept is based on the challenge-response mechanism. The cloud is required to respond to user or auditor challenges. These systems are theoretically successful, but there may be security risks such as data leaking or

collaborative assaults with the cloud. Furthermore, third-party auditors may be hostile and cooperate with participants to provide incorrect audit results, reducing the credibility of the audit results. With the introduction of blockchain, the decentralization and immutability of blockchain enable new possibilities for data integrity verification. The blockchain's decentralized nature enables it to always offer fair and correct audit findings based on the received audit proof, hence improving system security. A cloud environment security architecture that analyses data integrity using blockchain to improve cloud storage security. Zhang et al. suggested a certificateless public verification approach against procrastinating auditors using blockchain in 2019, allowing users to monitor whether auditors complete the verifications on time. Wang et al. created a blockchain-based private PDP system in 2020 by combining RSA and the quadratic residue group modulo the huge composite number N. Duet al. suggested a zero-knowledge storage auditing approach with an on-chain check mechanism in 2021 to address the storage freeriding security problem in rational actor environments (Du et al., 2021). However, these blockchain-based audits focus on data integrity in cloud storage rather than data integrity during cross-chain interaction. checked data integrity using blockchain but did not consider data auditing in cross-chain interaction. When validating this data, the client first download the data from the cloud, then determines and compares the MAC number of the downloaded file to the MAC value previously done to establish if the data integrity is assured. Although this approach is easy, obtaining comprehensive data needs a lot of energy and may result in data privacy leaks. Then, to decrease the processing cost, Seb et al. presented a block-based approach. The verification outcome may not be totally correct due to the predictable verification approach. Then, To finish the integrity verification, Ateniese et al advocated employing probabilistic techniques. They leveraged the homomorphic features of the RSA signature system to collect evidence in a quite tiny value, resulting in a significant reduction in transmission overhead. Curtmola et al. then built the data security verification technique in the situation of different versions, but it did not enable dynamic data operations. Ateniese et al. started with the motor operation of data. They introduced a basic modified PDP technique based on their prior work, allowing it to accommodate dynamic data modification. This technique, however, does not enable insert data. In answer to this issue, Wang et al. developed a PDP method that enables for complete dynamic functioning. The Merkle tree is used to ensure the integrity of the information block, and the Specifically refers (BLS) sign is used to ensure the accuracy of the information block content. Later, they devised a privacy protection authentication protocol that used random masking methods to prevent TPA from knowing the data information given by cloud service suppliers.

Even though the PDP verification system is capable of effectively verifying data integrity, it cannot recover faulty data. Juels et al. demonstrated a POR method based on

sentinels. However, it can only do a limited amount of verifications. Shacham et al. then employed the BLS quick text signature technique to create monomial confirmation tags, which can minimise verification communication overhead. However, it is hard to implement. Wang et al. proposed exploiting the linear properties of the error detection code to provide partial dynamic operations, but not dynamic data insertion. Chen et al. improved Wang's process and applied the Reed-Solomon erasing code approach to recover lost data, which improves recovery efficiency but increases computational cost.

MODELS

1. System Model

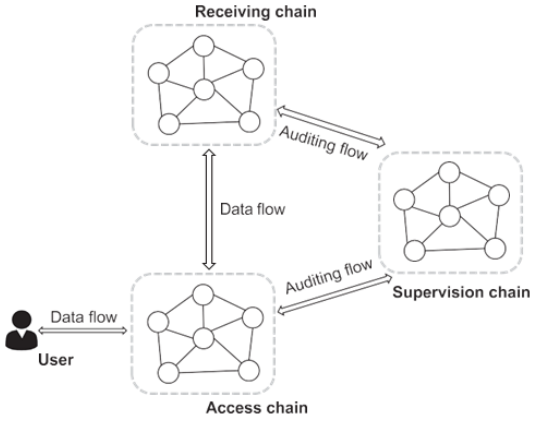


Figure 1 System model

In this research, we introduced DCIV, which takes data integrity verification into account in cross-chain interactions. We develop three consortium blockchains using Hyper ledger Fabric: an access chain (AC), a receiving chain (RC), and a supervision chain (SC). A data owner (DO), as indicated in Fig. 1, engages with in model to pre-process information off-chain to create audit information.

The data storage infrastructure which accepts and truly saves information F from DO is known as the access chain (AC). When there is cross-chain contact, it generates labels and promises the data to be transferred and provide the commitment and data F to RC. SC creates an audit evidence based on the transmitted data when it launches an audit challenge.

Receiving chain (RC) is a cross-chain data receiving platform that receives data and audit information from AC. When SC launches an audit challenge, RC creates an inspection proof based on the data received. The supervision chain (SC) is indeed a platform for assessing data integrity in cross-chain interactions that is collaboratively created by national regulatory bodies and centralised institutions in many industries. When it launches an audit challenge, RC and AC prepare evidence based on the challenge and faithfully deliver them to SC. SC accomplishes the process can be achieved after obtaining proofs in order to generate a reasonable judgement of this cross-chain interactions.

2. Threat Model

Off-chain Data owner holds the actual information, which is utilised to compare and validate the data supplied by Access Chain . We assume that Data owner is entirely honest in this scheme since we

are primarily interested in data integrity of cross-chain interaction.

(1) Access chain may give Receiving chain with partial data (not at all the data supplied by Data owner) in order to save communication costs or to deceive Receiving Chain into believing that the data it got is full.

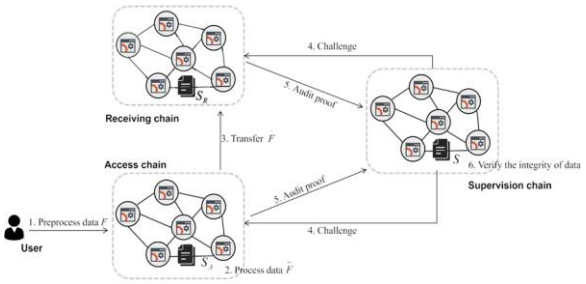
(2) For profit, Receiving chain may save the data from Access chain inadequately and attempt to persuade Supervision Chain that perhaps the data given by Access chain is incomplete.

(3) When Receiving chain provides the evidence to Supervision Chain, information leakage may occur.

DESIARABLE GOALS

We suggest the following aims to maintain data integrity in cross-chain interaction based on the study of the model.

- ➔ The data's integrity. Only if Access Chain transmits entire data to Receiving chain will it satisfy the data integrity test.
- ➔ Adaptable monitoring. In a pluggable version, Supervision Chain is independently of Access chain and Receiving chain. The Supervision Chain also isn't required for the cross-chain interaction. Only when a cross-chain interaction has to be audited does Supervision Chain become involved and do the verification.
- ➔ Low-cost supervision. Access Chain and Receiving chain provide audit evidence to SC. In the method of cross-chain contact, do not requires consideration of Access Chain or Receiving chain is stored.
- ➔ Public auditability. Any third party (whether in the form of a third-party auditor or blockchain) can perform the data integ- rity verification between chains without obtaining the complete data.



THE PROPOSED METHOD

In this section, we will first describe our approach for verifying data integrity. The design of Merkle trees is then described. The performance of various Merkle tree architectures is evaluated in terms of compute and transmission overhead. Finally, we explain different sampling verification methodologies and suggest a method for estimating the appropriate sample size.

A. *Data Integrity Verification Framework*

Clients, Cloud Storage Servers (CSS), and Blockchain are the three entities in this system (BC). Clients submit their own content to the CSS and utilise BC to ensure the integrity of the data. There are two steps to the total workflow. The preparation stage consists of five processes, as seen in Fig. 2(a). The client will divide his data into many shards in the first stage, then utilise these shards to create hash Merkle trees. The client and CSS will decide on the hash Merkle trees in the second stage. *CSS sends Digest i'* and the associated secondary source to BC in the third stage.

Clients will set the Merkle tree roots on the blockchain prior uploading data in this architecture. Because blockchain has a non-tamperability attribute, no client or CSS may edit the root recorded on the blockchain, making integrity verification more believable. At the same time, because the blockchain is dispersed, there is minimal chance that the information on the blockchain will be corrupted. As a result, data integrity verification is much more trustworthy.

B. *Structure of the Merkle Tree*

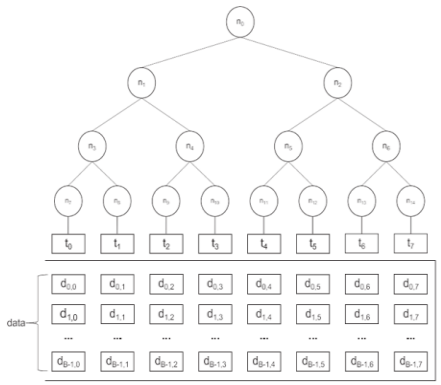


Figure 2 Rule of building merkle tree

The benefit of employing Merkle trees for data integrity verification is that the entire data file may be validated by a tiny section of the complete data shards, which is relatively modest regardless of its size of the original file. Figure 3 depicts the structure of Merkle trees. To aid in verifying every data shard of the private section, the public portion of this tree must be posted to the P2P CSS. This tree's private section is made up of data shards and random challenges. As a result, the clients save the confidential information locally. After data slicing, the data shards are the data submitted by clients to P2P CSS. This study examines the various Merkle tree architectures before discussing the system's communication and processing overhead under these configurations.

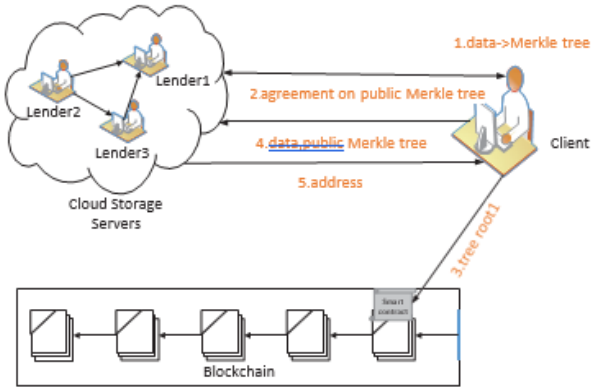


Figure 3 Preparation stage

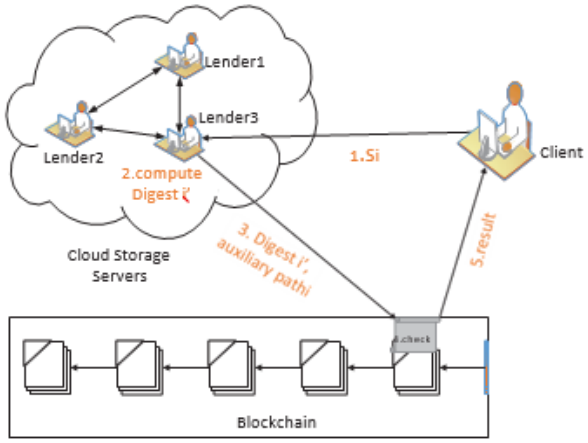


Figure 4 Verification stage

CONCLUSION

From this research, We learned about a novel cross-chain audit system based on the idea of controlling one chain after another, in which blockchain is used to verify the integrity of data of cross-chain interaction. Also researched about off-chain and on-chain data processing and integrity proof generation technologies that successfully assure cross-chain data integrity. Furthermore, we present an universal verification methodology for peer-to-peer cloud storage. It uses blockchain to overcome the problem of untrustworthiness in traditional verification mechanisms.

ACKNOWLEDGMENT

I would like to express our gratitude and thanks to our professor Mr. Chandrashekhar Goswami sir & Mr. Pritesh Prajapati sir, Dept. of Information Technology for their continuous motivation towards making my research and giving helpful and important guidelines that leads my project more and more towards achieving its efficiency.

REFRENCES

1. B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu. Blockchain based data integrity service framework for iot data. In *Proceedings of the 24th IEEE International Conference on Web Services (ICWS 2017)*, pages 468–475. IEEE, 2017.
2. C. Wang, Q. Wang, K. Ren, and W. Lou. Ensuring data storage security in cloud computing. In *17th International Workshop on Quality of Service (IWQoS 2009), Charleston, South Carolina, USA*, pages 1–9, 2009.
3. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou. Enabling public veri- fiability and data dynamics for storage security in cloud computing. In *Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS 2009), Saint-Malo, France. Proceedings*, pages 355–370, 2009.
4. S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin. Storj a peer- to-peer cloud storage network. https://storj.io/, 2014.
5. H. He, R. Li, X. Dong, and Z. Zhang. Secure, efficient and fine-grained data access control mechanism for p2p storage cloud. *IEEE Transactions on Cloud Computing*, 2(4):471–484, 2014.
6. Y. Zhang, C. Xu, X. Lin, and X. S. Shen, "Blockchain-based public integrity verification for cloud storage against pro- crastinating auditors," *IEEE Transactions on Cloud Com- puting*, vol. 23, p. 1, 2019.