Public Verification for Cloud Storage using Block-chain Technology

[1]Gowthami M, [2]Vasunthura M, [3]Yudhiksha V, [4]Deepak Kumar R

[1] Department, Institution name ,[2] Computer Science, Easwari Engineering College, [3] Computer Science, Easwari Engineering College, Computer Science, [4] Easwari Engineering College

[1]gowthami.m@eec.srmrmp.edu.in,[2]vasunthuramanickavasakam@gmail.com, [3]yudhiksha.v.19104170@gmail.com,[4] deepakram078@gmail.com

Abstract— The utilization of cloud storage services provides many advantages for user data management. Data integrity is one of the numerous security issues it raises, though. By using public verification methods, a user may use a third party. The current public verification schemes allow the user to hire an auditor to examine the data integrity on their behalf rather than depending on late auditors to finish verification. Furthermore, because most public verification systems are due to PKI, they have a problem with certificate management. We introduce the first certificate-less public verification algorithm (CPVPA) against an untimely auditor that is built on a block-chain. The fundamental knowledge is about requiring auditors to add every outcome of a validation as a transaction to a block-chain.

Index Terms— Cloud storage, data integrity, certificateless public verification, procrastinating auditors, blockchain.

1. INTRODUCTION

With cloud storage services, customers offload their data to cloud servers and gain remote access to that data through the Internet. These services provide consumers an effective and flexible solution to manage their data while freeing them from the burden of high local storage prices. Although these services provide significant benefits to consumers, they have also raised serious security concerns. Data integrity is one of the most essential security concerns. Unlike in the conventional data management paradigm, where users retain their data locally, consumers will no longer physically own their data once it has been outsourced to cloud servers. As a result, consumers are always concerned about data integrity, or if outsourced data is adequately kept on cloud servers.

In exercise, the integrity of outsourced data is compromised. For example, cloud servers may always conceal incidences of data corruption in order to maintain a good reputation, or they may erase a portion of data that is never viewed in order to save storage costs. Additionally, for financial or political motives, an external enemy may tamper with the outsourced data. As a result, the integrity of outsourced data should be checked on a regular basis. The users can do the verification themselves. This places a significant communication cost on users in order to retrieve and validate the data.

Users can employ public verification approaches to outsource data integrity verification to a dedicated third-party auditor. The auditor examines the data integrity on a regular basis and notifies the users when the inspection fails. The auditor is presumed to be honest and dependable in the majority of public verification procedures.These strategies would be rendered invalid if the auditor was compromised. To reduce verification expenses, an irresponsible auditor, for example, may always generate a satisfactory integrity report without doing the verification. In this manner, the auditor is almost non-existent. Additionally, a hostile auditor may conspire with cloud servers to provide a biased verification result in order to fool consumers for profit.

Additionally, most public verification systems rely on public key infrastructure (PKI), which requires the auditor to maintain the user's certificate in order to select the relevant public key for verification. As a result, these systems suffer from the certificate management challenge, which includes certificate revocation, storage, distribution, and verification, all of which are exceedingly costly and time-consuming in reality.

In this paper, we present the first certificateless public data integrity verification method, named CPVPA, that is resistant to malevolent and procrastinating auditors. The primary idea behind CPVPA is to employ blockchain-based currencies like

Bitcoin and Ethereum, which enable a tamper-proof and distributed mechanism to conduct transactions without the need for a central authority (i.e., a bank). After each verification, the auditor is required under CPVPA to generate a new transaction in which the information relevant to the verification is integrated and the auditor executes the transaction.

When the transaction is registered in the blockchain, the user may verify the time when the auditor does the verification by looking at the transaction's creation time. We emphasize that the more participants in a blockchain system, the higher the security guarantee it can give. As a result, rather than creating a new blockchain system, we build CPVPA atop a known and extensively used one (e.g., Ethereum). Moreover, CPVPA is based on certificateless cryptography , which eliminates the certificate management issue.

2. PROBLEM STATEMENT

2.1 Public Verification of Data Integrity

The key idea of public verification is that the user splits the data into multiple blocks, computes a signature for each one, and outsources the data blocks as well as corresponding signatures to the cloud server. The auditor then checks the integrity of challenged blocks by verifying the validity of the proof. The key technique used is aggregated signature, which allows the auditor to verify multiple blocks simultaneously without downloading the data. After data outsourcing, the auditor sets a verification period and verifies the outsourced data integrity at the corresponding time. If the verification passes, the integrity of the entire data set is ensured.

If it fails, the auditor generates a verification report containing multiple verification results. The most important details in this text are that the auditor is able to verify the data integrity without the user's participation, and that the longest delay within which the user needs to find the data corruption should be the verification period. The frequency at which the auditor checks the data integrity would not be very high in practice due to the following reasons: the auditor serves multiple users simultaneously, the higher frequency to perform the data integrity verification, the more costs to employ the auditor, and the heavy verification burden on the cloud server. Additionally, if integrating security mechanisms into existing cloud systems incurs considerable costs on the cloud service providers, most of the providers would not accept liability for the corresponding security guarantees in their Service Level Agreements (SLAs) and only ensure the service availability.

2.2 On the (in)efficiency of PKI-based public verification schemes

The majority of extant public verification techniques are based on public key infrastructure (PKI), in which a completely trusted certificate authority gives certificates to participants.

In order to select the relevant public keys for verification the auditor must handle user certificates. Certificate management, which involves revocation, storage, distribution, and verification, is, nevertheless, exceedingly expensive and time-consuming in reality. As a result, solving the certificate management issue might be both economical and advantageous in reality.

3. DEFINITIONS AND PRELIMINARIES

3.1 System model

The system model is shown in Fig. 1. There are four different entities in CPVPA: cloud user (data owner), cloud server, third-party auditor (TPA), and key generation center (KGC)

Data Owner: Person whose data has been sent to a cloud server is referred to as a user, the user is free to utilize needed data as necessary. After outsourcing the data to the cloud, the user requires and hires a TPA, who decides on a certification time with the TPA, and permits the TPA to frequently check the database's integrity.

Cloud server: It is known as a service provided by a service provider that is employed and then to supply storage. In addition to having a large amount of computing power, it has a large quantity of storage.

TPA: TPA satisfies the user's needs. It immediately detects data corruption and alerts the cloud server and viewer to the results of the verification. Verified is the communication between TPA and other parties.

KGC: A force exerts control upon the KGC. The user's identity is used to create a partial private key for the individual.

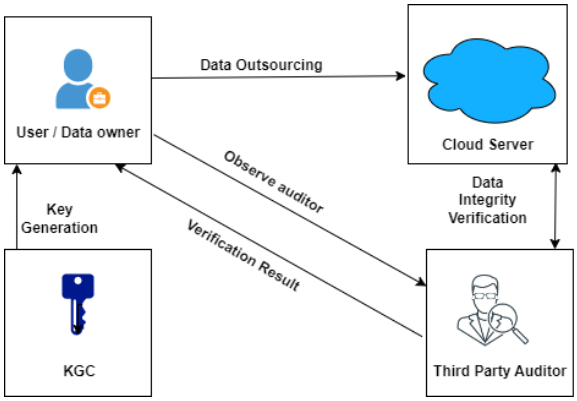

Fig .1. Model

The following is a formal definition of CPVPA:
Definition 1: The five algorithms that make up CPVPA are Setup, Store, Audit, LogGen, and CheckLog.
• Setup. The parameters required by the next algorithm are generated by this one.
• Store. A user can use this algorithm to outsource the data to a cloud server. For a TPA to be able to verify the data's integrity, the user must create verification tags (also known as signatures). The cloud server must also verify that the data was uploaded properly.
• Audit. This technique enables TPA to validate the data's integrity and allows the cloud server to demonstrate the upkeep of the outsourced data.
LogGen: This algorithm enables TPA to create a log file that contains data on the TPA's verification.
CheckLog: With the help of this technique, the user can examine the log file's accuracy and validity in order to audit the TPA's activity.

3.2 DESIGN GOALS

In this paper, we focus on the secure public verification of outsourced data integrity for cloud storage systems, where two issues exist: How to deal with the procrastinating TPA without involving a trustworthy third party. Existing public schemes anticipate that TPA will execute the data integrity verification within the timeframe specified. But, if the auditor procrastinates, the data corruption will not be detected as soon as feasible, and it may be too late to restore the data loss or harm. Without the assistance of a trustworthy participant, such procrastinating is difficult to identify.
Avoiding certificate management is possible. Certificate management is difficult and expensive in practice, as was previously addressed. it could be more cost-effective to allow TPA to check the data integrity without having to deal with user certificates and advantageous in actual use.
The following goals must be met in order to provide secure verification of outsourced data integrity in cloud storage under the aforementioned approach.

• **Efficiency:** The communication and computation overhead should be as minimal as possible; TPA can verify the data integrity without managing users' certificates and without having an a priori limit on the number of verification interactions; TPA should be stateless and not be required to maintain and update state during verification.

• **Security:** A cloud server must have the required data intact in order to pass the TPA's verification;
Both a malevolent and a tardy TPA cannot fool the user; Any two participants' collusion cannot undermine the proposed scheme's security.

3.3 PUBLIC BLOCKCHAIN AND ON-CHAIN CURRENCIES

Each data element in a blockchain is referred to as a block, and the collection of data elements is linear. A chain is created by connecting each block together, and each link is encrypted using a hash function. Each block typically includes transaction information, a timestamp, and a hash reference linking it to a previous block. A transaction can only be included in the block when its legitimacy has been confirmed. Broadly speaking, there are two forms of blockchain technology: private blockchain and public blockchain. Authorized participants who may be employed by the blockchain managers or the managers themselves carry out the verifications for a private blockchain (including the consortium blockchain). Any network user can carry out the verifications for a public blockchain, which allows for the recording of transactions.The verifications for a public blockchain can be carried out by any user on the network: A transaction could be documented.
The most popular incarnation of public blockchain\sis on-chain currency, such as Bitcoin and Ethereum. The public blockchain is employed in such currencies to act as an open and distributed ledger that effectively records transactions between two members. The people who carry out the transaction verifications are referred to as Bitcoin miners. Such a ledger is verifiable and naturally resistant to change of chained blocks. In reality, a blockchain system has a stronger security guarantee the more miners that participate in it. Since Ethereum is more expressive than other on-chain currencies and is one of the most widely used blockchain systems, we use the Ethereum blockchain to build CPVPA in this study.
The terms BlockHash and PrevBlockHash signify the hash values of the most recent block, respectively, while Time signifies the timestamp and MerkleRoot denotes the root value of a Merkle hash tree made up of all the transactions that were recorded in the most recent block. It is possible to think of the Ethereum ledger as a state transition system, with a "state" made up of the ownership status of all existing Ethers and a "state transition function" that accepts a state and a transaction as inputs and produces a new state as the output.
The block of transactions and this nonce are broadcast jointly by the first miner to find the nonce. The new block can then be added to the blockchain of other participants following their confirmation that the cretin is a valid solution. All the state diagram data has been updated after the block has been added to the chain.The "account" objects that make up Ethereum's state are. Users can send accounts and agreement accounts are the two main categories of accounts in Ethereum.

There are three fundamental properties in secure blockchain systems:

1) **ϕ-chain consistency:** Only the most recent blocks can be different between the blockchains of any two honest miners at any one time during the mining execution.
2) **(ι, ϕ)-chain quality:**In any series of or more succeeding blocks for an honest miner's blockchain, at least of the blocks were mined by honest miners. To put it another way, it is possible that no subsequent blocks in the blockchain will ever be produced by a hostile miner whose hashrate is less than 51% of the network's mining hashrate.Ethereum, $\phi \geq 12$.

3)**Chain Growth:** It is deterministic how many new blocks will be added to the blockchain at any given moment. In other words, the height of the blockchain can be relied upon to consistently rise in the short- or long-term.

4. THE PROPOSED CPVPA

4.1 Overview

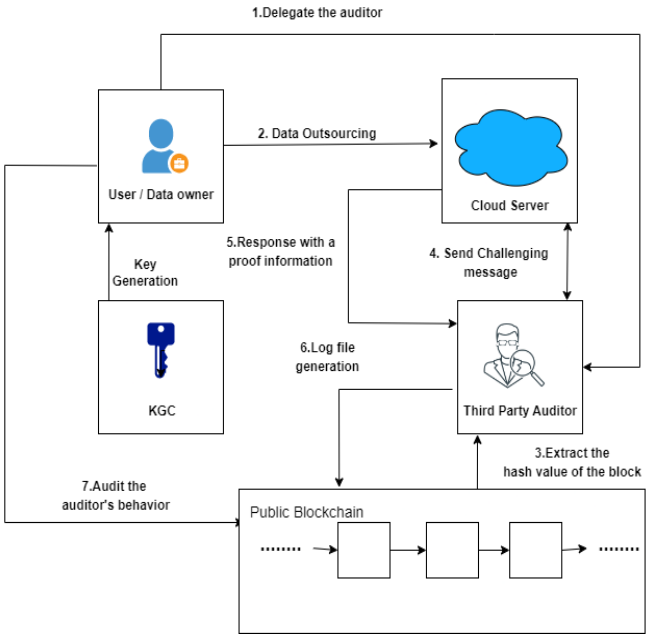

Fig. 2. Proposed CPVPA

CPVPA consists of two phases. In the first phase, the auditor verifies the integrity of outsourced data. In the Second phase, the user audits the auditor's behavior.

In the first phase, the verification period is determined by the user. For a point in time when the data integrity should be verified, TPA first extracts the hash values of ϕ successive blocks that are the latest ones confirmed on the Ethereum blockchain, where ϕ denotes the number of blocks deep used to confirm a transaction (in the Ethereum, ϕ = 12), and these hash values are denoted by {Blt−ϕ+1, Blt−ϕ+2, ..., Blt}. Then TPA generates a challenging message on {Blt−ϕ+1, Blt−ϕ+2, ..., Blt}, and sends the challenging message to the cloud server. Upon receiving the challenging message, the cloud server computes the corresponding proof. TPA checks the validity of the proof to verify the data integrity. If the checking fails, TPA informs the user that the data may be corrupted; Otherwise, TPA sets {Blt−ϕ+1, Blt−ϕ+2, ..., Blt} and the proof as a log entry,stores the entry to a log file, and creates a transaction that transfers 0 deposit from its account to the user's account wherein the data field is set to the hash value of the entry. In this paper, for the sake of simplicity, we assume the transaction is recorded to the block whose height is t+ϕ+ 1 and PrevBlockHash = Blt+ϕ.

In the second phase, the user audits the TPA's behavior in a much longer period compared with the verification period. We first show how is a single entry (without loss of generality, {Blt−ϕ+1, Blt−ϕ+2, ..., Blt} and the corresponding proof) in the log file audited by the user. The user first determines the verification time that TPA should perform data integrity verification. Then she/he obtains {Blt−ϕ+1, Blt−ϕ+2, ..., Blt} from the Ethereum blockchain according to the agreed verification time, and extracts the hash value of the entry from the transaction. Next she/he regenerates the challenging message on {Blt−ϕ+1, Blt−ϕ+2, ..., Blt}, and checks the validity of the corresponding proof by using the challenging message generated by herself/himself. If the checking passes, it means that TPA performs the verification correctly. Multiple entries can be audited simultaneously, and the auditing costs can be amortized over these entries.

5. REMARKS

Since there's a chance that the timestamp in the Ethereum block can be inaccurate, we don't use it as the transaction time in CPVPA.The height-derived practical time at which a block was created.
Using blockchain-based currency is mostly intended to fend off tardy auditors. Moreover, CPVPA accomplishes a desirable feature that may be of independent interest. Due to the unpredictable nature of each block's hash value, the auditor is unable to complete the scheduled verifications in advance. The majority of the public verification of data integrity systems already in use are fully compatible with the mechanism we suggest to thwart tardy auditors. For blockchain platforms based on alternative consensus algorithms, CPVPA can likewise be built. In comparison to Ethereum, prior PoS-based blockchain systems have a significantly smaller number of participants. As a result, if we build CPVPA on these blockchains, it will be much less expensive for an enemy to compromise its security.

6. RELATED WORK

The "proofs of retrievability" (POR) approach was suggested by Juels et al. to guarantee the integrity of data saved on an untrusted server. Nevertheless, public verification is not taken into account, hence the data owner needs to frequently check the data's integrity. The owner of the data must continue to maintain it available for validation. As a result, in order to acquire and utilize the data, the data owner must carry a significant communication and verification load. Around the same time, Ateniese et al. presented the "provable data possession" (PDP) strategy, the first method to take public verification into account. With this technique, the data owner might hire a third-party auditor to verify the data's integrity on their behalf.Subsequently, under the strongest model put forward by Juels et al., Shacham et al. provided the first compact POR strategy with comprehensive proofs of security against arbitrary attackers. In the wake of the study of Shacham et al., other public verification systems have been suggested. On top of a homomorphic signature approach, these schemes are constructed.Public verification that protects privacy has received attention in recent literature as well. The integrity of data from outsourced sources may be checked by the auditor without revealing the data's content thanks to a privacy-preserving public verification mechanism. A random mask must be used by the cloud server to blind the proof information in the majority of current privacy-preserving systems, so that the auditor may verify the accuracy of the proof information without having to extract the data content. In our approach, we encrypt the data before creating the tags in order to shield user information from the auditor. As the auditor and cloud server are both considered potential collaborators in the CPVPA, the cloud server would simply provide the auditor the data in violation of the users' right to privacy.Because the aforementioned techniques rely on certificate-based encryption, they must deal with the certificate management issue. Several approaches based on identity-based signature techniques were put out to avoid handling certificates in a public verification process. The key escrow problem is an inherent weakness of these approaches, though. The first certificateless public verification approach was put out by Wang et al.

Then, certain increased security certificateless public verification systems were put out. The homomorphic certificateless signature techniques serve as the foundation for these systems. As a result, under these schemes, the auditor is not required to handle the user certificates without dealing with the key escrow issue.

In furthermore, it is considered that the auditor is trustworthy and honest for the current schemes. This is a fairly strong assumption, though, because corrupting auditors is a real possibility. Recently, Zhang et al. provided the first public verification technique with resistance against malicious auditors, while Armknecht et al. proposed the first PoR strategy that thwarts bad auditors. These frauds are powerless to stall auditors who might not finish the data integrity verification on time. A tardy auditor may stray from the main goal of public verification, which is to find data corruption as soon as feasible.It is necessary to emphasize that, for public verification methods to work in practice, opposition against tardy auditors is crucial. You may find a more thorough survey on public data integrity checking here.

CONCLUSION

In this paper, we present CPVPA, a certificateless public verification technique against the procrastinating auditor. CPVPA makes use of on-chain currencies, with each audit being linked into a transaction on the blockchain using on-chain currencies. Moreover, CPVPA is not affected by the certificate management issue. As compared to current schemes, the security study shows that CPVPA gives the greatest security guarantee. We have performed a thorough performance study, which shows that CPVPA has constant communication overhead and is efficient in terms of computing overhead.

REFERENCES

[1]J. Xue, C. Xu, and L. Bai, Dstore: A distributed system for out sourced data storage and retrieval, Future Generation Computer Systems, vol. 99, pp. 106-114, 2019.

[2] Y. Zhang, C. Xu, N. Chen, H. Li, H. Yang, and X. Shen, Chronos+: An accurate blockchain-based time-stamping scheme for cloud storage, IEEE Trans. Serv. Comput., vol. 13, no. 2, pp. 216229, 2020.

[3] L. Sun, C. Xu, Y. Zhang, and K. Chen, An efficient iO-based data integrity verification scheme for cloud storage, *Sci. China Inf. Sci.*, vol. 62, no. 5, pp. 59 101:159 101:3, 2019.

[4] Y. Zhang, C. Xu, X. Lin, and X. S. Shen, ”Blockchain-BasedPublic Integrity Verification for Cloud Storage against Procrastinating Auditors,” *IEEE Transactions on Cloud Computing*, to appear,doi:10.1109/TCC.2019.2908400.

[5] X. Zhang, J. Zhao, C. Xu, H. Li, and Y. Zhang, Cipppa: Conditional Identity privacy-preserving public auditing for cloud-based wban against malicious auditors, *IEEE Transactions on Cloud Computing*,to appear, doi: 10.1109/TCC.2019.2927219.

[6] Y. Zhang, C. Xu, J. Ni, H. Li, and X. Shen, Blockchain-assisted public-key encryption with keyword search against

keyword guessing attacks for cloud storage, *IEEE Transactions on Cloud Computing*,to appear, doi:10.1109/TCC.2019.2923222.

[7] Y. Tu, J. Gan, Y. Hu, R. Jin, Z. Yang, and M. Liu, Decentralized identity authentication and key management scheme, in *2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2)*,2019, pp. 26972702.

[8] Y. Miao, Q. Huang, M. Xiao, and H. Li, Decentralized and privacy preserving public auditing for cloud storage based on blockchain,*IEEE Access*, vol. 8, pp. 139 813139 826, 2020.

[9] X. Jia, N. Hu, S. Su, S. Yin, Y. Zhao, X. Cheng, and C. Zhang, Irba: An identity-based cross-domain authentication scheme for the internet of things, *Electronics*, vol. 9, no. 4, p. 634, 2020.

[10] C. Lin, D. He, X. Huang, X. Xie, and K.-K. R. Choo, Blockchain based system for secure outsourcing of bilinear pairings, *Information Sciences*, vol. 527, pp. 590 601, 2020.