Towards Secure and Transparent Cloud Auditing: A Blockchain and IPFS-Driven Framework with Batch Verification

Houaida Ghanmi*†‡, Nasreddine Hajlaoui‡, Haifa Touati‡, Saadi Boudjit§, Mohamed Hadded¶, Mohand Yazid Saidi*, Paul Muhlethaler‖
*L2TI Lab, University Sorbonne Paris Nord, France.
†ENSI, University of Manouba, Tunisia.
‡IReSCoMath Research Lab, University of Gabes, Tunisia.
§LITIS Lab, University of Rouen Normandy, France.
¶Abu Dhabi University, United Arab Emirates.
‖INRIA Paris, France.
{houaida.ghanmi@edu.univ-paris13.fr, nasreddine.hajlaoui@fsg.rnu.tn, haifa.touati@univgb.tn, saadi.boudjit@univ-rouen.fr, mohamed.elhadad@adu.ac.ae, saidi@univ-paris13.fr, paul.muhlethaler@inria.fr}

***Abstract*—With the rapid advancement of cloud storage and the increasing use of connected devices, uploading data to the cloud results in the loss of physical control by data owners, making confidentiality and integrity entirely dependent on Cloud Service Providers (CSPs). This raises concerns about whether CSPs effectively safeguard outsourced data, as any malicious behavior can lead to data tampering or deviation. Traditional auditing schemes rely on Third Party Authorities (TPAs), which are not always trustworthy. Although various cloud data auditing mechanisms have been proposed, few effectively address the challenge of ensuring data integrity without relying on trusted third parties. To overcome this limitation, we propose a secure and efficient distributed blockchain-based data integrity auditing scheme. Specifically, our approach randomly assigns the audit task to a user selected from among the system participants via blockchain. Blockchain and InterPlanetary File System (IPFS) technologies are leveraged to enforce access control. Furthermore, the proposed scheme supports low-cost batch integrity verification without the need for a TPA. Theoretical analyses confirm that our solution ensures data traceability and auditability, and reduces reliance on third parties. Finally, our simulations show that proof generation remains under 0.7 seconds for 600 data blocks at 256-bit security, while verification costs remain negligible.**

***Index Terms*—Cloud storage audit, Blockchain, Batch auditing, Privacy, Integrity verification.**

I. Introduction

In a recent study, International Data Corporation (IDC) estimates that the global volume of data will grow from 33 trillion bytes in 2018 to 175 trillion bytes in 2025 [1]. In the face of this continued growth, the outsourcing of sensitive information to remote cloud servers has grown rapidly. In cloud computing, data owners (DO) can rent hardware, software, and maintenance services from the cloud to store their data [2]. Cloud storage services offer the advantages of pay-as-you-go, easy sharing, and cross-platform access, enabling them to provide highly scalable, and low-cost, and anytime-accessible services [3]. Though cloud storage provides many advantages, outsourcing data results in a loss of direct control over the data, raising major security and integrity concerns [4]. As a result, users are beginning to worry about the security of their outsourced data, and data integrity becomes of utmost importance. The CSP may delete rarely used or highly repetitive data to save storage space, thereby compromising user data integrity in the cloud. Cloud data can also suffer damage due to software or hardware failures or be externally impacted by concurrent attacks [5]. Moreover, the cloud might conceal security vulnerabilities to enhance the application's reputation [6]. Therefore, most users can't know whether their data in the cloud is always complete due to its limited capabilities. To address this issue, many auditing schemes have been proposed [7], in which data integrity auditing is performed regularly by a TPA. These enable users to entrust a third party with auditing the integrity of their data by verifying its accuracy and consistency.

Although trusted third-party verification provides robust audit results and reduces user communication and computational load, some of them [7] do not preserve confidentiality and disclose sensitive customer data. Moreover, the presence of a TPA also introduces new security risks, as collaborating with a TPA allows a CSP to use false evidence to bypass verification, and auditor corruption is possible. For example, a malicious auditor can generate a verified audit result without performing the verification process to reduce consumption costs [8]. Therefore, it cannot be fully trusted because it can collect information about the outsourced data during the audit process. Therefore, establishing a trust relationship between DOs and CSPs becomes a fundamental issue to ensure data integrity.

Given these limitations, researchers are increasingly exploring

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

blockchain as a promising alternative to strengthen trust in the audit process. These methods can develop log files for the audit process and detect unauthorized alterations of cloud data [9]. Despite the immutability and transparency offered by blockchain, these systems have a significant communication overhead. These costs have a crucial influence on the expenses of DOs. To improve the reliability of data audits in the cloud, we can exploit blockchain's decentralization, transparency, and immutability properties to randomly select a network user and assign them the audit task. This dynamic designation would allow for a fair distribution of the audit task while eliminating the dependency on a centralized trusted third party and avoiding single points of failure. We also integrate batch auditing, which would allow the designated auditor to process multiple audit requests simultaneously. The main contributions of this article are as follows:

- We integrate blockchain and IPFS to create a decentralized data audit protocol for cloud storage that eliminates the need for a TPA, enhancing trust and transparency.
- We employ smart contracts to automatically assign audit tasks to randomly selected users, ensuring balanced computational load.
- We introduce an incentive-based mechanism to encourage user participation in the audit process.
- We demonstrate the security and reliability of the proposed scheme through theoretical analysis, and evaluate its performance in terms of computation time and gas costs.

The remainder of this paper is organized as follows: Section II presents the related work, while sections III and IV present in detail the proposed scheme and security analysis. Section V evaluates the performance of the proposed work. Section VI concludes the paper and discusses research directions.

II. Related work

To achieve secure cloud storage, a lot of research has been done to verify the integrity of data stored in a cloud environment. Yue et al. [7] proposed a general blockchain-based framework for data integrity verification, eliminating the need for a TPA. This framework leverages Merkle Hash Tree (MHT) based validation techniques with hard random numbers and sampling strategies to optimize this verification. Moreover, by delegating auditing tasks to minor nodes acting as auditors, only the DO can generate the corrected MHT leaf node, thus enhancing privacy protection. However, this work has some shortcomings, including the lack of incentive mechanisms and the lack of support for batch auditing. Moreover, traceability cannot be guaranteed because operation logs are not stored on the blockchain, no formalized security proof of the scheme is provided, and the blockchain leads to significant communication overhead. Chen et al. [10] proposed the data mining-based auditing model. They used a hybrid AES and ECC cipher to enhance the system security. They also used Shamir secret sharing for data transfer without the intervention of a trusted third party and a re-signing technique to avoid collusion due to user revocation. Although this method avoids needing a trusted entity during the data transfer process, it still relies on a TPA for auditing. Moreover, many authentication methods are involved, increasing the cost and computation time. Shao et al. [11] developed an incentivized public audit scheme integrating threshold signatures (t,n), batch verification, and data masking. Although this work has the advantage of enhancing security and encouraging active participation of auditors, the appointment of auditors remains semi-decentralized, which introduces a risk of a single point of failure. Moreover, despite the blockchain being used as an incentive medium, it is not fully exploited to ensure the transparency of the audit process. Wang et al. [12] propose a public audit framework based on a blockchain consortium to eliminate the need for TPA. The proposed model uses an MHT and smart contracts to generate a challenge for data integrity verification. The CSP generates the proof and calls a verification contract after receiving the challenge from the blockchain. Although this architecture has advantages in transparency and decentralized auditing, the generation of cryptographic tags by smart contracts incurs significant computational overhead expressed by high gas fees. Wang et al. [13] designed a cloud-based data integrity audit protocol. Their solution relies on threshold signatures and ElGamal encryption to enable anonymous participation of multiple users in the transfer of audit rights. Their solution demonstrates that their protocol reduces computational costs by 50% for users and improves overall efficiency by 22% compared to existing approaches by outsourcing computations to aggregators. However, their solution relies on a semi-trusted TPA and does not provide an incentive mechanism for performing the audit task.

TABLE I: Comparison of several auditing schemes

Schemes	BC.B	Pub.V	Inc.M	Dec	Bat.A
[7] (2020)	✓	✓	✗	✓	✗
[10] (2021)	✗	✓	✗	✗	✗
[11] (2023)	✓	✓	✓	✗	✗
[12] (2024)	✓	✓	✗	✓	✗
[13] (2025)	✗	✓	✗	✗	✗
Our	✓	✓	✓	✓	✓

Notes: [✓] Supported, [✗] Not supported, [BC.B] Blockchain-Based, [Pub.V] Public Verifiability, [Inc.M] incentive mechanism, [Dec] Decentralized, [Bat.A] Batch Auditing

Based on the above literature review summarized in Table I, we find that many current audit solutions suffer from a lack of decentralization, transparency, and incentives. Additionally, current audit protocols have several limitations, including the risks of centralization and single points of failure that make it difficult for participants to engage in the audit actively. Furthermore, the high computational and communication costs of several existing schemes represent an obstacle to their practical effectiveness. However, in the field of cloud storage, data integrity verification and privacy protection are essential. Motivated by these observations, we propose a decentralized audit scheme in which a user is randomly selected via the blockchain to verify data integrity. An incentive mechanism is also introduced to encourage the active participation of

auditors.

III. Proposed scheme

In this section, we present a novel architecture for auditing data stored in the cloud based on Blockchain. The proposed system in this study aims to ensure data integrity of data stored in the cloud, eliminating the need for a TPA.

A. Preliminaries

(i) **Bilinear mapping** G_1 and G_T be two multiplicative cyclic groups of prime order p, and g is a generator . Let $e : G_1 \times G_1 \rightarrow GT$ be a bilinear application which has the following properties:

- The Non-Degeneracy property: $\exists u, v \in \mathbb{G}_1$ such that $e(u, v) \neq 1$.
- The bilinear property of $e(u^a, v^b) = e(u, v)^{ab}$ for all $u, v \in G_1$ and $a, b \in \mathbb{Z}_p^*$
- There exists an efficient algorithm to compute the pairing operating $e(u, v)$ for $u, v \in G_1$.

(ii) **Computational Diffie-Hellman problem:** For any input $x, y \in \mathbb{Z}_p^*$, the decisional Deffi-Hellman problem means that, it is computationally infeasible to compute $g^{xy} \in G_1$ given the tuple (g, g^x, g^y).

(iii) **Discrete Logarithm (DL) problem:** Let G_1 be a multiplicative cyclic group, g be a generator of G_1, where a is a random element of $\mathbb{Z}_p^*$. It is computationally infeasible to output a given the value of $g, g^a \in G_1$ as input.

B. System model

As can be seen in Figure 1, the system model of our proposal consists of four types of entities: Blockchain (BC), IPFS, Data Owner (DO), and Random Auditor.

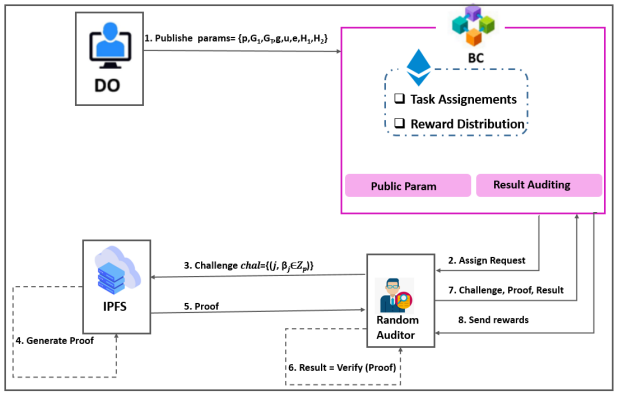

Fig. 1: The workflow of the proposed model.

In order to achieve accurate auditing without *TPA*, we leverage *blockchain* and smart contracts to manage audit tasks in a decentralized manner. Before outsourcing data to *IPFS* (IPFS serves as a decentralized cloud storage), the *DO* initializes the system by generating public parameters, which auditors then use for proof verification. When an audit request is triggered by a *DO*, the *blockchain* assigns the task to a random user via a smart contract. Once an *auditor* is selected, they generate a challenge and send it to *IPFS*. To prove that the data is stored reliably, *IPFS* generates a proof in response and sends it back to the *auditor*. The *auditor* then verifies the proof and publishes the challenge and the audit result on the *blockchain* to receive their reward and ensure a fair and decentralized audit process, as shown in Figure 2. The details of our scheme are shown below.

- **Setup**$(\lambda) \rightarrow$ (params). On inputting the security parameter λ, the system performs the following operations. Let G_1, G_T be two multiplicative cyclic groups of prime order p, and $e : G_1 \times G_1 \rightarrow GT$ a bilinear map, and g and u be two independent random generators such that $g, u \in G_1$. Let $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : \{0,1\}^* \rightarrow G_1$ be two hash functions. The DO randomly selects a value $\alpha \in \mathbb{Z}_p^*$ as its private key, and computes the public key pk $= g^\alpha$. Finally, the parameters $params = \{p, G_1, G_T, H_1, H_2, g, u, e\}$ are published by DO into the blockchain, and the private key is kept secret to himself.
- **DataUpload** $(F, \{mj\}) \rightarrow (\{\sigma_j\}, sig)$: Let F be the data file, the Do first splits the file F into n blocks $F = \{m_1, m_2, \ldots, m_n\}$. Then, the DO generates a tag for each m_i where $i \in X = \{1, 2, \ldots, n\}$. Next, for each m_i block, the DO computes $h_1 = H_1(F)$, $h_2 = H_2\{ID_{DO}, m_i, ID_i\}$, $\sigma_j = H_2(F)^\alpha \cdot H_1(F \parallel j) \cdot u^{m_j}$, where, $\{\sigma_j\}$ is the tag set, ID_{DO} is the identity of the DO, and $\{ID_i\}$ is a set of the unique identities. Finally, the DO sends $\{F, \{\sigma_j\}, h_1, \{ID_i\}\}$ to the cloud, and publishes, $\{h1, \{ID_i\}, \{h_2\}\}$ in the blockchain network.
- **AuditManager** After the data transfer, a smart contract *AuditManager* performs the audit task. This smart contract is dynamically triggered to manage a list of potential auditors $U = \{U1, U2, \ldots, Un\}$. For each audit request via blockchain, the smart contract selects one auditor from the set U.
  $U_{aud} \leftarrow Random(U1, \ldots, Un)$.
- **Chal and Proof Generation** When a user receives a cloud data integrity verification request, it uses the metadata stored on the blockchain to generate the following challenge:
  1) U_{aud} generates random subset $l = \{j_1, j_2, ..., j_c\}$ of set $X = \{1, 2, \ldots, n\}$.
  2) U_{aud} generates the random number $\beta_j \in \mathbb{Z}_p^*$.
  3) U_{aud} sends the auditing challenge $chal = \{(j, \beta_j)\}_{j \in l}$ to the cloud.
  4) Receiving $chal = \{(j, \beta_j)\}_{j \in l}$, the Cloud calculates a block proof $B_P = \sum_{(j,\beta_j) \in \{chal\}} m_j \beta_j$ and a tag proof $T_P = \prod_{(j,\beta_j) \in \{chal\}} \sigma_j^{\beta_j}$.
  5) The cloud sends the proof $proof = \{B_P, T_P\}$ to the U_{aud}.
  6) Upon receiving the $proof = \{B_P, T_P\}$, the U_{aud} checks the following equation:

$$e(T_P, g) \stackrel{?}{=} e\left(H_2(F)^{\sum_{(j,\beta_j) \in \text{chal}} \beta_j}, \ pk\right).$$

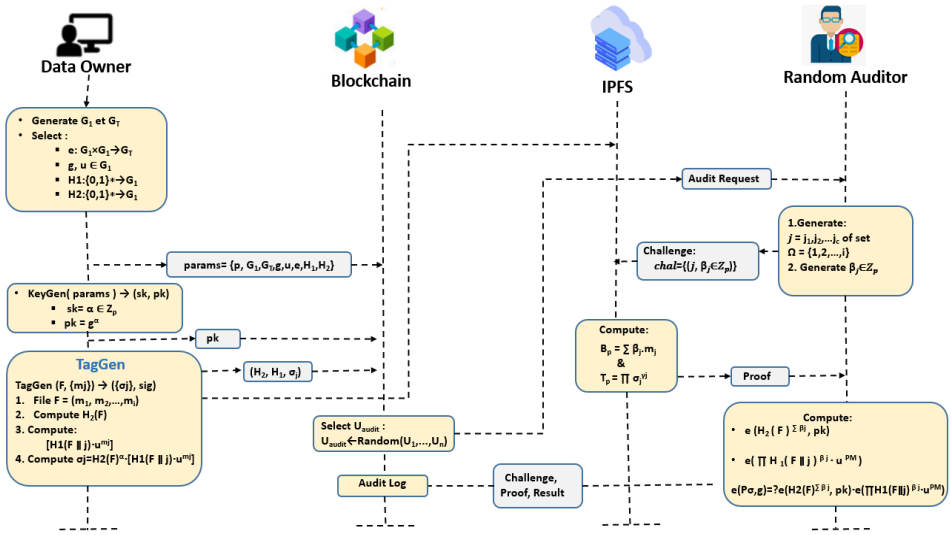

Fig. 2: System architecture.

$e\left(\prod_{(j,\beta_j)\in \text{chal}} H_1(F \parallel j)^{\beta_j} \cdot u^{B_P}\right)$, if satisfies above equation holds, U_{aud} outputs 1, otherwise 0. Finally, U_{aud} publishes the audit result $< chal; proof; result >$ in the blockchain.

- **Correctness:** The following equation justifies the correctness of the proof verification as follows, which holds only for valid proof:

$$
\begin{aligned}
e(T_P, g) &= e\left(\prod_{(j,\beta_j)\in \text{chal}} \left\{H_2(F)^{\alpha} \cdot \left[H_1(F\|j)\cdot(u)^{m_j}\right]^{\beta_j}\right\}, g\right) \\
&= e\left(\prod_{(j,\beta_j)\in \text{chal}} H_2(F)^{\alpha\beta_j}, g\right) \\
&\quad \cdot e\left(\prod_{(j,\beta_j)\in \text{chal}} \left[H_1(F\|j)\cdot(u)^{m_j}\right]^{\beta_j}, g\right) \\
&= e\left(H_2(F)^{\sum_{(j,\beta_j)\in \text{chal}}\beta_j}, pk\right) \\
&\quad \cdot e\left(\prod_{(j,\beta_j)\in \text{chal}} H_1(F\|j)^{\beta_j}\cdot u^{B_P}\right)
\end{aligned}
$$

- **RewardManager** Users with significant computing power (potential users) in the blockchain network have the same rights to perform auditing tasks. However, some users may hesitate to get involved in these activities, resulting in inefficient computations. To address this issue, we propose in this paper a blockchain-based incentive mechanism to incentivize all potential users to play an active role in the auditing process. Our system leverages the characteristics of blockchain, including decentralization, tamper resistance, and traceability, to record certain user information and reward them fairly. The owner of the outsourced data block file sets the value of the incentive reward. Then, the blockchain helps identify the first valid contributors and allocate rewards to them.

- **Batch auditing** The support for batch auditing enables multiple users to send audit queries to the blockchain simultaneously. In this case, the U_{aud}, randomly chosen by the blockchain, consolidates all queries and sends only one query to the cloud server, for which it receives only single response. The U_{aud} randomly chooses, for each file $fi \in \{f_1, f_2, \ldots, f_v\}$, a subset of block indices i and associates a random coefficient $\beta_j^i \in_R \mathbb{Z}_p$ at each block m_j. Then, U_{aud} generates the challenge $chal^*$ and sends it to the cloud:
chal$^* = \left\{\left(\{(j, \beta_{j,i})\}_{j\in l}\right)\right\}_{i=1}^{v}$.
Afterwards, the following equation can be used to confirm the correctness of the answer:
$\prod_{i=1}^{v} e(T_p, g) = \prod_{i=1}^{v} e\left(H_2(F_i)^{\sum_j \alpha\beta_j^i}, pk_i\right) \cdot e\left(\prod_j H_1(F_i\|j)^{\beta_j^i}\cdot u^{B_{Pi}}\right)$

IV. Security Analysis and Performance Evaluation

In this research work, the security level offered by this solution is demonstrated through the following aspects:

A. Preserving data privacy from auditor

Theorem 1 The U_{aud} is restricted to verifying data blocks only. As a result, the U_{aud} cannot access the uploaded data or the identity of the DOs, thereby ensuring data integrity.

Proof To answer the challenge $chal = \{(i_j, \beta_j)\}_{j\in l}$, the cloud sends $B_P = \sum_{(j,\beta_j)\in\{chal\}} m_j\beta_j$ and $T_P = \prod_{(j,\beta_j)\in\{chal\}} \sigma_j^{\beta_j}$. Moreover, while having $B_P = \sum_{(j,\beta_j)\in\{chal\}} m_j\beta_j$, the U_{aud} might try to identify the value of m_j. In this case, the U_{aud} cannot identify any useful information about the audited data blocks m_j, because B_P contains the product of the entire sequence of multiple audit blocks.

B. Soundness

In the case where a CSP discovers that it has lost some blocks m_j of data due to an attack or internal failure. However,

he may choose not to disclose this fact. Therefore, it returns $B_P^* = \sum_{j=1}^{n}(m_j^*\beta_j)$ and $T_P^* = \sum_{j=1}^{n}\sigma_j^{*\beta_j}$ rather than $B_P = \sum_{j=1}^{n}(m_j\beta_j)$ and $T_P = \sum_{j=1}^{n}\sigma_j^{\beta_j}$, where m_j^* refers to forged blocks in an attempt to pass the audit.
Let us assume that, for m_j blocks, the possession of the data blocks proof are:

$$\sigma_j = H_2(F)^{\alpha} \cdot H_1(F \parallel j) \cdot u^{m_j}$$

So

$$\sigma_j^{\beta_j} = H_2(F)^{\alpha\beta_j} \cdot H_1(F \parallel j)^{\beta_j} \cdot u^{\beta_j m_j}$$

This means that

$$T_P = \sum_j \sigma_j^{\beta_j} = H_2(F)^{\alpha\Gamma} + \sum_j H_1(F \parallel j)^{\beta_j} + u^{B_P}$$

, where $\Gamma = \sum_j \beta_j$.

With m_j^*, we have:

$$T_P^* = H_2(F)^{\alpha\Gamma} + \sum_j H_1(F \parallel j)^{\beta_j} + u^{B_P^*}$$

In this case, the U_{aud} efficiency can be analyzed through the following derivation.

$$\frac{T_P}{T_P^*} = \frac{u^{B_P}}{u^{B_P^*}} = u^{B_P - B_P^*} = u^{-\Delta}$$

In this case, $B_P^* - B_P = \Delta$ and Δ can be surprisingly a negligible or a notable difference. But it is known that, for even a small change in $\beta_j \in \{chal\}$ or m_j, $m_j\beta_j$ is entirely a different value. Finding a Δ which will produce $u^{B_P^* - B_P} = 1$ is a hard problem and guessing Δ will contradict the Discrete Logarithm Problem (DLP).

C. Removing single points of failure

The scheme proposed in this paper uses decentralized blockchain technology to manage the audit task instead of using a TPA and to effectively reduce the risk of system failure due to node unavailability or malicious attacks. Distributing audit tasks via smart contracts increases system resilience and eliminates single-point-of-failure vulnerability. This approach distributes computing power and enhances data security, making it easier to verify data integrity and improving overall system efficiency.

D. Collusion and Denial-of-Service Attacks

To enhance security, specific measures are implemented to counter potential attacks, such as denial-of-service (DoS) attacks and auditor collusion. First, the audit task is randomly assigned, which prevents a malicious auditor from predicting or controlling the upcoming task, making any collusion strategy ineffective. Furthermore, auditors do not interact with each other and do not share any common information other than that publicly recorded on the blockchain. Therefore, they can't act maliciously. Furthermore, our system does not depend on any single entity, as all interactions (audit, storage, verification) are distributed. Our audit system integrates DoS attack tolerance mechanisms. If an audit request fails (high latency or no response), the system can immediately reassign the audit to another available auditor via a new random selection to ensure that audits are neither blocked nor delayed.

V. Experimental Results

In this section, we evaluate the performance of our scheme based on the GNU Multiple Precision Arithmetic (GMP) [14], Pairing-Based Cryptography (PBC) [15] libraries, and the C language. The experiment adopts the elliptic curve group of type A, then the sizes of G_1 and Z_p are set to 160 bits. Our auditing scheme was implemented using VMware Workstation. The virtual machine's configuration comprises a 25 GB hard disk and 4 GB memory. The virtual machine's operating system type is Ubuntu and the version is 24.10 with a local computer configured with an Intel(R) Core(TM) i5-8250U CPU @ 1.80 GHz and 12G RAM. Each series of tests was carried out 10 times to collect and obtain the average data results.

The computational overhead in our approach can be divided into two parts, off-chain and on-chain, depending on whether the operation is performed on the blockchain. We start by analyzing the time costs of the off-chain operation and then the on-chain computational overhead of our system. In particular, we evaluate the time consumption of CSPs handling both tag generation and proof generation, and U_{aud} verifies the proofs under different numbers of data blocks.

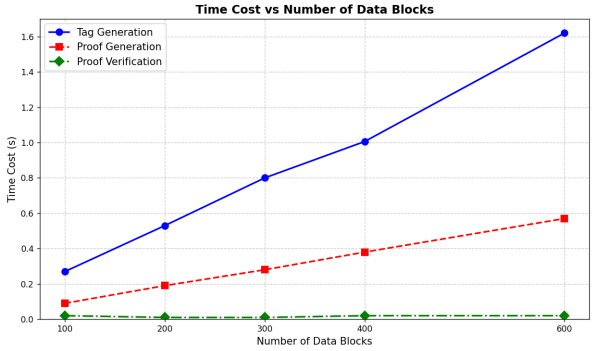

Fig. 3: Time costs of each operation.

In Figure 3, we analyze the computational overhead incurred in the proposed protocol concerning the tag generation , proof generation, and proof verifcation. Since the proposed protocol incurs a constant time of 0,02 s for proof verifcation, the cost of proof verifcation is parallel to the x-axis in the graph. The cost of tag generation is proportional to the number of blocks to be downloaded, and the cost of proof generation is proportional to several challenged blocks for the data audit.

As shown in Figure 4, as the security level increases up to $|p| = 256$, the growth in proof generation time remains limited. Therefore, under high security standards, the proof generation time overhead of the proposed scheme is deemed acceptable and demonstrates the scalability of the scheme for

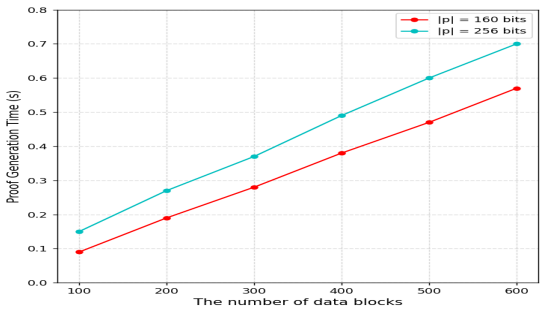

Fig. 4: Effect of security level $|p|$ on proof generation time.

stronger cryptographic settings.

In Ethereum, each transaction will consume gas, the cost

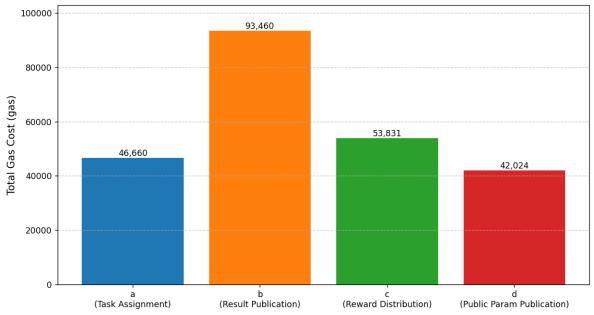

Fig. 5: Toal Gas Costs.

of which represents the computational resources needed to execute transactions on the blockchain. There are many blockchain-based data integrity audit schemes. Still, most of them store data or data block labels on the blockchain, such as scheme [16], which causes significant memory overhead for the blockchain. By storing only the bilinear map data and audit results, our scheme achieves minimal storage overhead. Figure 5 provides an overview of the gas fees of transactions performed during the audit process across various Ethereum smart contracts. These transactions encompass a range of actions, from public parameters publication and task assignments to result publication and reward distribution. It should be noted that the gas costs associated with these transactions vary, demonstrating differences in computational complexity. For instance, when publishing audit results with user details, the gas cost is significantly higher than publishing public parameters. This reflects the additional compute resources required to process the expanded data size. Similarly, transactions involving large numerical values, such as reward distribution, tends to incur higher gas costs. These blockchain operations are heavily influenced by gas costs, which impact transaction fees and the overall efficiency of blockchain applications.

VI. Conclusion

Cloud data auditing technology is essential for enhancing cloud data security. Our improved system eliminates the need for a TPA in the audit process by assigning the audit task to a randomly selected user through smart contracts. This approach enhances security, resists malicious auditors, and ensures a decentralized, verifiable, and tamper-proof audit process. In this study, we also design an incentive-based public auditing system that rewards auditors for participating. Moreover, the batch auditing technique reduces computational overhead by allowing simultaneous verification of multiple data blocks. Security and performance analyses demonstrate that our system achieves the desired levels of accuracy, robustness, security, and efficiency. Future research will address the scalability challenges of using blockchain for auditing in dynamic cloud environments, aiming to support additional features for cloud storage.

References

[1] Ge Kan, Chunhua Jin, Huihui Zhu, Yongliang Xu, and Nian Liu. An identity-based proxy re-encryption for data deduplication in cloud. *Journal of systems architecture*, 121:102332, 2021.

[2] Houaida Ghanmi, Nasreddine Hajlaoui, Haifa Touati, Mohamed Hadded, Paul Muhlethaler, and Saadi Boudjit. Blockchain-cloud integration: comprehensive survey and open research issues. *Concurrency and Computation: Practice and Experience*, 36(15):e8122, 2024.

[3] Houaida Ghanmi, Nasreddine Hajlaoui, Haifa Touati, Mohamed Hadded, and Paul Muhlethaler. A secure data storage in multi-cloud architecture using blowfish encryption algorithm. In *International Conference on Advanced Information Networking and Applications*, pages 398–408. Springer, 2022.

[4] Houaida Ghanmi, Nasreddine Hajlaoui, Haifa Touati, Mohamed Hadded, Paul Muhlethaler, and Saadi Boudjit. A decentralized blockchain-based platform for secure data sharing in cloud storage model. In *International Conference on Advanced Information Networking and Applications*, pages 338–348. Springer, 2024.

[5] Nasreddine Hajlaoui, Chaima Bejaoui, Tayssir Ismail, Houaida Ghanmi, and Haifa Touati. A hybrid architecture for secure data sharing in multi-clouds system. *The Computer Journal*, 68(1):58–73, 2025.

[6] Yang Xu, Cheng Zhang, Guojun Wang, Zheng Qin, and Quanrun Zeng. A blockchain-enabled deduplicatable data auditing mechanism for network storage services. *IEEE Transactions on Emerging Topics in Computing*, 9(3):1421–1432, 2020.

[7] Dongdong Yue, Ruixuan Li, Yan Zhang, Wenlong Tian, and Yongfeng Huang. Blockchain-based verification framework for data integrity in edge-cloud storage. *Journal of Parallel and Distributed Computing*, 146:1–14, 2020.

[8] Mario Felipe Munoz, Kaiwen Zhang, Aamir Shahzad, and Mustapha Ouhimmou. Loglog: A blockchain solution for tracking and certifying wood volumes. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9. IEEE, 2021.

[9] Fran Casino, Eugenia Politou, Efthimios Alepis, and Constantinos Patsakis. Immutability and decentralized storage: An analysis of emerging threats. *IEEE access*, 8:4737–4744, 2019.

[10] Yange Chen, Hequn Liu, Baocang Wang, Baljinnyam Sonompil, Yuan Ping, and Zhili Zhang. A threshold hybrid encryption method for integrity audit without trusted center. *Journal of Cloud Computing*, 10:1–14, 2021.

[11] Bilin Shao, Li Zhang, and Genqing Bian. Incentive public auditing scheme with identity-based designated verifier in cloud. *Electronics*, 12(6):1308, 2023.

[12] Chenxu Wang, Yifan Sun, Boyang Liu, Lei Xue, and Xiaohong Guan. Blockchain-based dynamic cloud data integrity auditing via non-leaf node sampling of rank-based merkle hash tree. *IEEE Transactions on Network Science and Engineering*, 2024.

[13] Yamei Wang, Yuexin Zhang, Ayong Ye, Jian Shen, Derui Wang, and Yang Xiang. Anonymous and efficient (t, n)-threshold ownership transfer for cloud emrs auditing. *IEEE Transactions on Information Forensics and Security*, 2025.

[14] The GNU Multiple Precision Arithmetic Library . Accessed: May 2025.

[15] Ben Lynn. Pairing-Based Cryptography Library. Accessed: May 2025.

[16] Kun Hao, Junchang Xin, Zhiqiong Wang, and Guoren Wang. Outsourced data integrity verification based on blockchain in untrusted environment. *World Wide Web*, 23:2215–2238, 2020.