International Journal of Ambient Computing and Intelligence
Volume 14 • Issue 1

A Blockchain-Based Security Model for Cloud Accounting Data

Congcong Gou, College of Finance and Economics Management, Sichuan University of Arts and Science, China*
Xiaoqing Deng, College of Intelligent Manufacturing, Sichuan University of Arts and Science, China

ABSTRACT

The popularity of cloud accounting is due to its low cost of entry, efficient data processing, and high business efficiency. However, security issues in cloud storage can affect user trust in the service. To address these security issues, a blockchain-based encryption technology model for cloud accounting data security is proposed. Firstly, the feasibility of integrating blockchain technology and cloud accounting is analyzed. Then, an elliptic curve cryptography-based cloud accounting data security solution is proposed. Blockchain and evidence chain technology are used to ensure data security and support data privacy protection for cloud service providers and third-party auditors. The proposed solution has a small computational overhead, as it does not require exponentiation or bilinear pairing. Experimental results show the proposed solution can enhance user control over cloud accounting data, ensure data transmission security, and improve trust between users and cloud accounting service providers. Moreover, it is more efficient.

KEYWORDS

Blockchain, Cloud Accounting, Data Security, Elliptic Curve Cryptography, Evidence Chain, Third-Party Auditor

INTRODUCTION

In recent years, the importance of cloud accounting has become increasingly significant, with many small- and medium-sized enterprises and organizations using cloud accounting. The cloud accounting model has become one of the main directions of accounting information development (Moll & Yigitbasioglu, 2019; Huttunen et al., 2019). Unlike traditional accounting models, users only need to store their accounting data in the cloud to obtain low-cost, efficient, and flexible online accounting services. At the same time, users can be freed from the high costs of updating accounting software, regularly maintaining financial information systems, and building data storage infrastructure.

Despite the significant advantages of cloud accounting, there are also security issues that users need to be aware of when enjoying the convenience of cloud storage services, including: a) the cloud storage model separates the ownership and control of user data, and the cloud service provider (CSP) may intentionally delete data that users do not frequently access for economic purposes; b) CSP (Content Security Policy) may experience software failures and hardware damage, leading to the loss

DOI: 10.4018/IJACI.332860 *Corresponding Author

or damage of user data; and c) data stored in the cloud may be maliciously damaged by other users (Parast et al., 2022). Ensuring and verifying the security and integrity of cloud accounting data and establishing an effective protection mechanism for cloud accounting data have become urgent tasks in promoting the development of cloud accounting.

To address the issue of cloud data integrity verification, audit schemes have emerged. Early cloud audit (CA) schemes generate absolute evidence, and auditors need to access all original data, resulting in significant computational and communication overhead (Gudeme et al., 2019). Provable data possession (PDP) schemes only select partial data for integrity auditing and can ultimately confirm the integrity of all data with a high probability, reducing the computational and communication overhead of auditors. PDP schemes use homomorphic tags, which can aggregate all tags and have high flexibility. Based on whether the auditor of the integrity scheme is the user or a TPA (Third Party Auditor), they can be divided into private CA schemes and public CA schemes (Rabaninejad et al., 2019).

In private CA schemes, the private key of users will not be leaked, but it requires significant computational and communication overhead, which is a burden for users with limited device resources. Public CA schemes delegate the data possession verification to TPA, and TPA can audit on behalf of users with only a small amount of public information, reducing the burden on users and being able to monitor the behaviors of users and the cloud (Wang et al., 2019). However, TPA schemes have the following disadvantages: a) single point of failure, as all users' cloud data are audited by a unique TPA, the entire audit system will collapse once the TPA fails; b) performance bottleneck, as the number of cloud users and the scale of cloud data increase, the audit time and network overhead of TPA schemes will increase significantly, making TPA the bottleneck of the entire audit system; and c) data privacy, in TPA schemes, TPA may combine user metadata and audit data to infringe user privacy (Razaque et al., 2021).

To reduce computational overhead and improve audit efficiency, many CA schemes based on elliptic curve cryptography (ECC) have been proposed. Xue et al. (2019) proposed an identity-based CA scheme based on ECC, which uses the user identity information as a public key to solve the complex certificate management problem. In addition, malicious deceptive behaviors from TPA can be detected by checking the audit results in batches. Huang et al. (2020) proposed a certificateless CA scheme to solve the complex certificate management issues and the key escrow problem, and batch auditing was also supported. Ming and Shi (2019) proposed a privacy-preserving certificateless CA scheme that has higher audit efficiency compared to the CA scheme based on Bohen-Lynn-Shacham (BLS) signatures.

Blockchain technology has significant advantages in ensuring data security and integrity, which coincides with the urgent need of cloud accounting to enhance data security (Ionescu, 2019). Blockchain technology has the following advantages: a) decentralization, blockchain is based on a distributed network, using mathematical methods instead of a central organization to establish trust relationships between nodes; b) scalability, blockchain uses specific economic incentive mechanisms to attract users to participate in the blockchain system. With an increasing number of active users in the distributed network, the overall computing power of the network becomes stronger; and c) security and trustworthiness, blockchain encrypts data and ensures data integrity and authenticity through consensus algorithms between nodes, ensuring the privacy of user data (Xu et al., 2019).

Wang et al. (2020) proposed a blockchain-based CA scheme in which smart contracts were utilized to ensure fairness among different entities. However, this scheme did not provide data privacy protection for CSPs and did not explain how to dynamically update data. Wei et al. (2020) proposed a blockchain CA scheme that supported privacy protection and can resist collusion attacks between TPAs and CSPs. However, in the process of dynamic updating, tags were generated with the true index of data blocks, which can lead to additional computational overhead if the true index changes. Sun et al. (2020) proposed an adaptive authenticated data structure with privacy-preserving

capabilities for big data streams in the cloud, supporting dynamic data operations and dynamic scalable public auditing. By combining the trap-door hash function and BLS signature technique, dynamic data operations were allowed. Mishra et al. (2022) combined the Fibonacci tree structure and circular linked list to store outsourced data in the cloud environment. The index hash table was maintained by the TPA to accelerate the auditing and verification efficiency. However, CSPs had to store all the data before and after the dynamic operations to trace data changes, increasing the data redundancy of the system.

Jayaprakash et al. (2022) proposed an enhanced Merkle hash tree method for an effective authentication model in a multiple-owner cloud. Merkle hash trees provide effective data mapping, making it easy to identify changes in the data. And the developed model supported privacy protection public auditing to provide a secure cloud storage system. Zhe et al. (2022) considered the forward security for Public-key Authenticated Encryption with Keyword Search (PAEKS) and introduced a new primitive: forward secure public-key authenticated encryption with keyword search. Wahhab et al. (2022) explored the role of internal auditors in auditing and analyzed large amounts of data through a large number of questionnaires. Singh et al. (2019) adopted multiple technologies, such as big data collection, Gephi network visualization analysis, feature extraction, and scoring of doubtful points. This framework is based on the big data-driven paradigm of a policy-tracking audit model. Parmoodeh et al. (2023) studied the prospects of blockchain in auditing practice. Due to its tamper resistance and other characteristics, the possibility of being deceived was greatly reduced. However, blockchain might also cause delays in big data auditing programs, and overall, the benefits outweigh the disadvantages.

Traditional auditing typically consumes a large amount of manpower, material resources, time, and unforeseeable errors. In order to make the auditing process safer and more efficient, we propose a CA data security scheme based on ECCs to improve data security and computational efficiency of cloud accounting systems. Using blockchain technology, the scheme provides user privacy protection against both CSP and TPA.

The main contributions of this paper are summarized as follows: First, in the dynamic updating phase, virtual indexing technology is used to avoid additional computational overhead caused by changes in the true index, improving the update efficiency. Second, in the deletion and verification phases, the concept of an evidence chain is introduced to further guarantee the private protection and operation accountability against potential malicious CSPs or users. Third, the authors are applying blockchain to cloud accounting data, and most of the cloud accounting data are structured or semistructured with characteristics of continuity, periodicity, and multiple utilization. Only a few people do this in this way.

RELATED KNOWLEDGE

ECC

Compared to traditional cryptography methods, such as the Digital Signature Algorithm (DSA), Rivest-Shamir-Adleman (RSA), and Diffie-Hellman, the ECC adopted in this paper provides higher encryption efficiency at the same security level, as it can achieve the same security level with smaller key sizes (Yang et al., 2022). The elliptic curve equation is defined as follows:

$$y^2 = x^3 + ax + b \bmod p\,, \tag{1}$$

where p is a large prime number. Let Z_p be the finite field of modulo p, then $a,b\hat{I}Z_p$, and $4a^3 + 27b^2 \bmod p \neq 0$. All points on the elliptic curve form an additive group denoted as $E(F_p)$, where points $(x,y\hat{I}Z_p)$ satisfy Equation (1). In addition, the infinitely distant point O is also on the elliptic curve.

Given a point Q on the elliptic curve and a positive integer $t \in Z_p^*$, the addition of points on the elliptic curve is defined as $tQ = Q + Q + ... + Q$, and $E(F_p)$ is an Abelian group.

Given two points P and P_1 on an elliptic curve, where $P_1 = xP$, the Elliptic Curve Discrete Logarithm Problem (ECDLP) is to find $x\hat{I}Z_p$ such that the equation $P_1 = xP$ is satisfied. Given three points P, P_1, and P_2 on an elliptic curve, $x, y \in Z_p^*$, the Elliptic Curve Diffie-Hellman Problem (ECDHP) is to find xyP given P, xP, and yP.

Blockchain

Blockchain technology, as the underlying technology of Bitcoin, is essentially a distributed database that is maintained by multiple nodes and supports read and write operations that are immutable. In a point to point (p2p) network composed of untrusted nodes, blockchain technology, cryptography knowledge, and a consensus algorithm create an open, transparent, traceable, and tamper-proof security system to ensure data consistency among nodes (Gad et al., 2022). In a blockchain system, all participating nodes use a consensus algorithm to package a set of consensus-derived results into blocks, which are connected in a chain-like structure. Blocks that have been added to the blockchain cannot be modified, and each participating node in the blockchain system locally stores a completely consistent data chain.

Figure 1 shows the data structure of a block in a blockchain, which consists of two parts: the block header and block body. Typically, the block header (version 4Byte) consists of a total of 80 bytes and contains three sets of information. The first set is data that references the hash value of the parent block (32 bytes), which is used to connect the block with the previous block in the blockchain. Among them, the blockchain body hash value can uniquely and clearly identify a block, but it is not actually included in the data structure of the block. The second set of information includes difficulty (4 bytes), timestamp (4 bytes), and nonce (4 bytes), which are related to mining competition. The third set of information contains the Merkle tree root (32 bytes), which is the data structure used to summarize all transactions in the blockchain. The blockchain body records the number of transactions and transaction data stored in the blockchain. On average, each transaction contains at least 250 bytes, and each block contains at least 500 bytes of transaction information. The block body below Figure 1 shows the Merkle tree structure, which includes leaf nodes tx1, tx2, tx3, and tx4, as well as nonleaf nodes hash1, hash2, hash3, and so on.

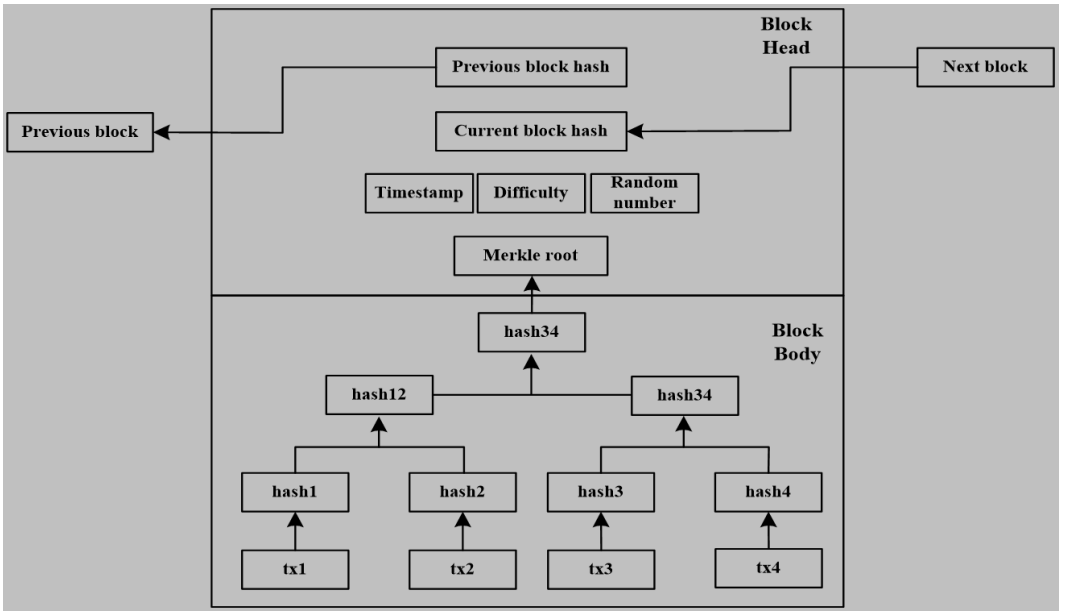
Blockchain-Based Cloud Accounting System

The proposed solution is a cloud accounting data security model based on blockchain encryption technology, as shown in Figure 2. Different from most existing cloud accounting service models, the proposed model uses blockchain to store and back up user accounting data and the corresponding hash value. The ECC technology and evidence chain are used to ensure the security and integrity of user cloud data, thereby providing users with a more secure online accounting service.

In the layer of "Software as a Service," there are mainly three aspects: financial accounting, managerial accounting, and operating decision. Financial accounting includes voucher processing, ledger accounting, account closing, and statement accounting. Managerial accounting includes budget management, cost management, performance management, and so on. All these are presented to accounting users in the form of software. The platform layer is "Platform as a Service," which contains basic service, accounting service, and decision support. The technical support of the platform layer is blockchain, which contains an access layer, security mechanism, and data layer. The data layer is very important and includes financial accounting, sales, expense, performance assessment, and so on. And blockchain is supported by an infrastructure layer and hardware layer.

Additionally, the blockchain characteristics of immutability, traceability, and transparency can enable enterprises and cloud accounting service providers to conduct cloud accounting business in

Figure 1. Block structure of the blockchain



a trusted and transparent environment, helping to establish a new trust model for enterprises and accountants.

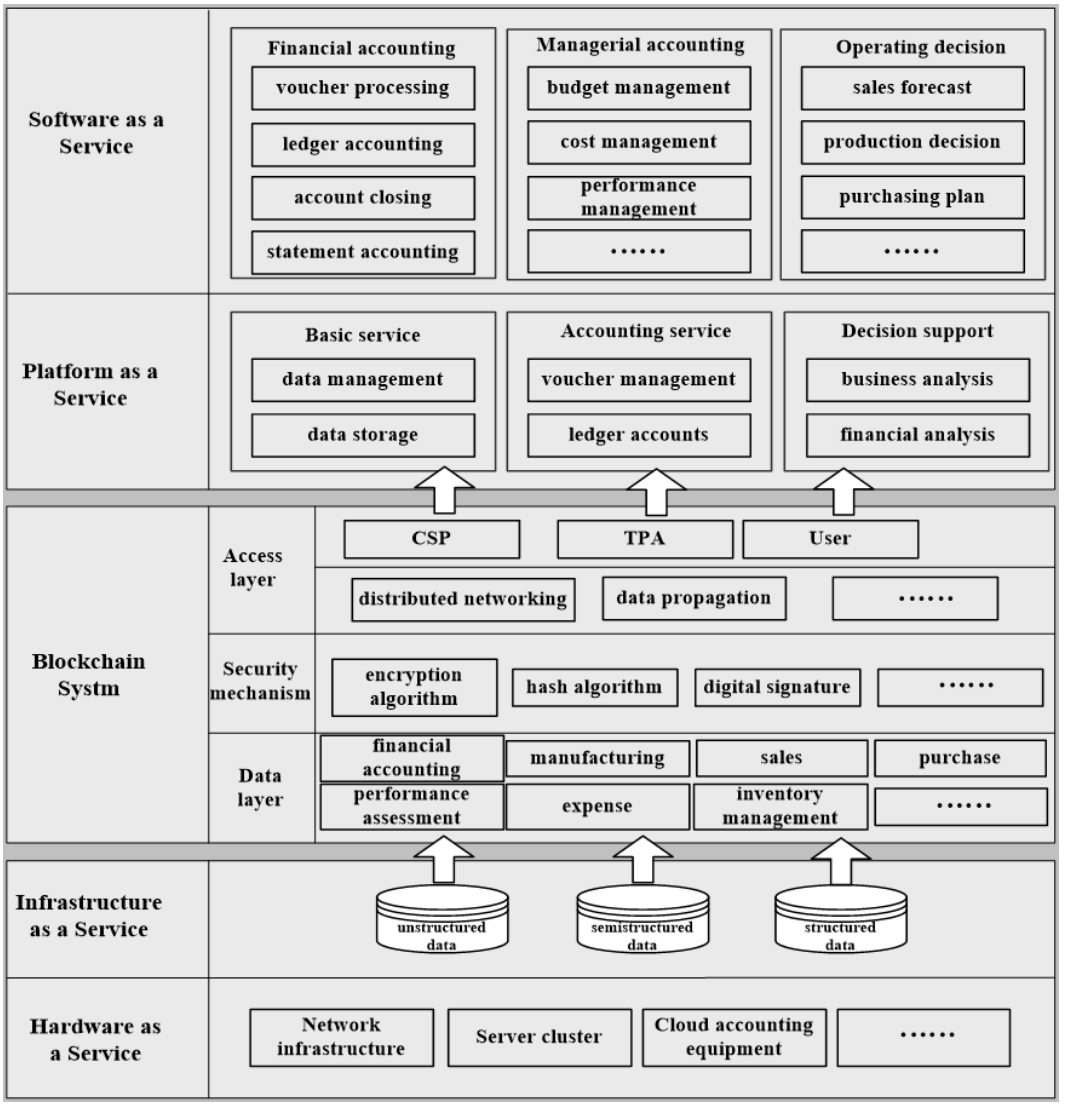
Security Assumptions

In the proposed scheme, it is assumed that the CSP is an untrusted entity, meaning that the CSP may delete data without the authorization of users or may not delete data in accordance with deletion requests from users. It is assumed that a user is a semihonest entity, meaning that the data owner may deny the data deletion requests they previously issued and falsely accuse the cloud server of deleting data without authorization. This solution allows for passive attacks, meaning that an adversary can eavesdrop on all communications in the system, and unauthorized users may collude to obtain plaintext information from each other. Considering the application environment of the proposed solution and the security goals proposed by Ramokapane et al. (2016), the security goals of the cloud accounting system are set to meet requirements such as correctness, integrity, deterministic data deletion, secure access control, and accountability tracking.

Algorithm Procedure

The proposed architecture includes four interacting entities: 1) CSP, which provides data storage services to users but is untrusted and may delete cloud data for profit or steal user data; 2) user, who uploads data to the cloud and wants to protect data privacy. The user is the owner of the data; 3) TPA, a semihonest third-party auditor appointed by the user to perform the task of auditing the integrity of cloud data; and 4) evidence chain, an arbiter that remains impartial and under no control of any party. The interaction process among the entities is as follows: the user uploads accounting data to the CSP. When data integrity needs to be verified, the user sends an audit request to the CSP and TPA. The CSP and TPA generate the same challenge parameters noninteractively based on the public parameter timestamp on the blockchain. The CSP generates a data possession proof based on the challenge parameters and sends it to the TPA. The TPA verifies the integrity of the cloud data using the evidence chain and sends the verification results to the user and CSP.

Figure 2. Cloud accounting data security architecture based on blockchain encryption(Wang et al., 2020)



In a cloud accounting system, a data security solution needs to meet the following requirements:

- Storage correctness: The verification of the audit scheme can only be passed if the CSP stores the user's files in their entirety.
- Publicly auditable: Users can delegate a TPA to perform audits on their behalf to reduce their own expenses.
- Comprehensive data privacy protection: The CSP cannot know the content of cloud data; when the TPA verifies the integrity of cloud data on behalf of the user, they cannot know the content of the data.
- Security: The solution can resist forgery attacks and replay attacks from cloud servers, and the CSP and TPA cannot obtain the user's private key during the audit process.

- Efficiency: The audit process of the solution does not require bilinear pairing operations, exponentiation operations, or hash operations mapped to points.
- Noninteractivity: When a challenge message is generated in the solution, a pseudo-random function is used to input the timestamp on the blockchain and output the index of the challenge block. On the one hand, the randomness of the index can be ensured because the timestamp is not under the control of both parties. On the other hand, the CSP and TPA do not need to exchange information, reducing the possibility of collusion.

The specific process of the proposed solution is as follows. In the setup phase, the CSP inputs the security parameter λ and outputs public parameters $\{E, G, p, g, H, st\}$. E represents an elliptic curve, p is a large prime number, G is defined as a cyclic group of order p on E, g is a generator element of group G, and H is a secure hash function mapped to Z_p^*, defined as:

$$st = 2^{\delta}, \tag{2}$$

where *st* is a parameter required for generating virtual indices, which can be changed according to different application scenarios, $\delta \in N^*$. The more frequently it is dynamically updated, the larger the value of *st*.

In the key generation phase, the user selects a random number $d \in Z_p^*$ as the private key *pk* and calculates the public key $pk = P \in G$ as follows:

$$P = d \cdot g. \tag{3}$$

In the tag generation (TG) phase, the user divides data to be stored into *n* blocks, and the true index of each data block is $i \in I = \{1, 2, ..., n\}$. The virtual index $\mu_i = i \cdot \mathrm{st}(1 \le i \le n)$ is calculated, and a conversion table is generated to maintain the correspondence between the true index and the virtual index (Yu et al., 2020). Symmetric encryption and decryption algorithms Enc, Dec are selected, and key_1 and key_2 are chosen. The encryption algorithm Enc and key_1 are used for encryption, denoted as (Yu et al., 2020):

$$M = m_1 \,||\, m_2 \,||\, ... \,||\, m_n. \tag{4}$$

Blocked processing can improve computational efficiency and enable sampling verification. Afterwards, the user selects a random number k and calculates $K = k \cdot g \in G$. Let *R* be the x-coordinate of *K*. Then, for each data block m_i, the user uses the virtual index η_i to calculate the following equation (Yu et al., 2020):

$$S_i = k^{-1} m_i + k^{-1} dRH(\eta_i), \tag{5}$$

where *i* is the unique index of each file block, and the user can calculate the tag offline during this process. The encryption algorithm Enc and key_2 are used to encrypt $M = m_1 \,||\, m_2 \,||\, ... \,||\, m_n$:

$$M' = \mathrm{Enc}_{key_1}(M) = m_1' \,||\, m_2' \,||\, ... \,||\, m_n'. \tag{6}$$

Afterward, $\{i, m_i', R, S_i\}_{(1\le i\le n)}$ is sent to the cloud server, $\{i, R, \mathrm{Enc}, key_2\}$ is sent to TPA, and $\{i, m_i, m_i', R, S_i\}$ is deleted locally.

In the challenge generation (CG) phase, when TPA and CSP receive an audit request from the user, they use a pseudo-random function, input the current timestamp *t* of the blockchain, output *c* random numbers from *I* as the index of the audited data block, and search for the corresponding virtual index to form the challenge message $chal = \{i_j, \eta_j\}_{(1\le j\le c)}$. The timestamp *t* used in this process is not controlled by CSP or TPA, and no interaction is required between the two parties, achieving fairness and randomness.

In the proof generation (PG) phase, after receiving the audit request and generating *chal*, CSP calculates $pr = \mid \rho', S \mid$ as the data possession proof and sends it to TPA. ρ' and *S* are calculated as follows (Wang et al., 2019):

$$\rho' = (m_{i_1}' \mid\mid m_{i_2}' \mid\mid ... \mid\mid m_{i_c}') \tag{7}$$

$$S = \sum_{j=1}^{c} S_{i_j} . \tag{8}$$

During the proof verification (PV) phase, TPA receives $pr = \mid \rho', S \mid$, decrypts ρ' using Dec and key_2 to obtain $m_1 \mid\mid m_2 \mid\mid ... \mid\mid m_c$. Then, TPA calculates $\rho = \sum_{j=1}^{c} m_{i_j}$, and performs the following verification (Sun et al., 2020):

$$K = S^{-1} \cdot [\rho \cdot g + \sum_{j=1}^{c} H(\eta_j) \cdot R \cdot P] . \tag{9}$$

After calculating *K*, TPA verifies whether the x-coordinate of *K* is equal to *R*. If so, TPA informs the user that the data integrity has not been compromised. Finally, TPA sends the integrity verification result to the user and CSP.

During the dynamic updating (DU) phase, when inserting less than 2δ data blocks between adjacent data blocks, the virtual indices of other data blocks will not change, and there is no need to recalculate the tags. When inserting a new data block m_j encrypted using Enc and key_1 after data block m_i, the user calculates the virtual index of m_j as follows:

$$\eta_j = (\eta_i + \eta_{i+1}) / 2 . \tag{10}$$

Afterward, the label S_j of m_j is computed and encrypted using the encryption algorithm Enc and key_2 to obtain m_j'. The user sends $\{\mathrm{insert}, \eta_j, m_j', S_j\}$ to CSP. Finally, $i+1$ is taken as the true index of m_j, and the true index of each data block is incremented by 1 to update the index conversion table. The CSP receives the data and label, and based on the virtual index η_j, finds the corresponding position to insert m_j' and S_j. Finally, the index conversion table is updated in the same manner.

When a data block m_j' is deleted, the user sends $\{\mathrm{delete}, \eta_i\}$ to CSP. The virtual index η_i is deleted, and each true index after *i* is decremented by 1 to update the index conversion table. The

CSP receives the deletion command and, based on η_i, finds the corresponding position to delete m_j' and its corresponding label S_i. The virtual index table is then updated in the same manner.

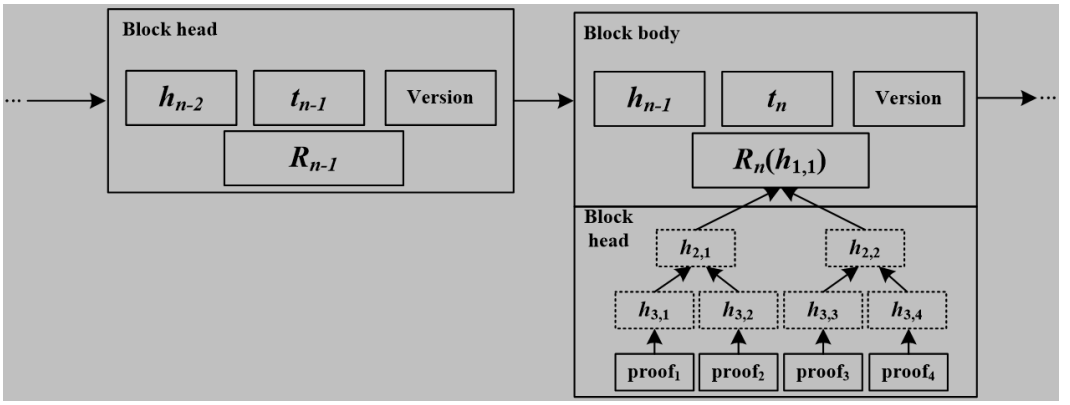
When a data block m_i' is modified to be m_j', the user sends $\{\text{change}, \eta_i\}$ to CSP. The CSP returns the corresponding data block m_i' to the user. The user decrypts m_i' using Dec and key_2 to obtain m_i, then replaces m_i with the encrypted data block m_j using the encryption algorithm Enc and key_1, and computes the label S_j. m_j is then encrypted using Enc and key_2 to obtain m_j', and $\{\eta_i, m_i', S_{m_i}\}$ is sent to the CSP. The CSP receives the data and label, and based on the virtual index η_i, finds the corresponding position to modify the data block and its corresponding label from m_i' and S_i to m_j' and S_{m_j}.

After data deletion or modification, evidence is generated and delivered to the TPA, which adds the evidence to the evidence chain. The data block of the evidence chain contains two parts: the block header and the block body. The block header mainly includes the previous block hash value, the Merkle root value, the timestamp, and the version information. The block body contains the generated evidence information. The specific structure of the evidence chain is shown in Figure 3.

In Figure 3, h_{n-2} denotes the hash value of the pre-previous blockchain. t_{n-1} is the timestamp of the previous block. R_{n-1} represents the root value of the previous block. $h_{2,1}$, $h_{2,2}$, $h_{3,1}$, $h_{3,2}$, $h_{3,3}$ and $h_{3,4}$ are all non-leaf nodes. TPA calculates the hash value H(proof$_j$) of proof$_j$ as the leaf node to generate the evidence of Merkle hash tree (MHT), and the root value of MHT is R_n. The hash value h_n of the data block to be processed, block$_n$, is calculated as $h_n = H(h_{n-1} \,||\, t_n \,||\, R_n)$, where t_n represents the timestamp of the current block and h_{n-1} represents the previous hash value in the hash chain. The proof$_j$ is then inserted into the processing block using the MHT structure, and the root node R_n is generated. After successful Practical Byzantine Fault Tolerance (PBFT) consensus, the block will be linked to the evidence chain.

The user verifies whether the deletion request has been completed as requested by using the deletion information provided by the CSP. If the user finds that the requested data have been leaked, they can request the TPA to trace responsibility. The TPA calculates the root value by using the deletion evidence proof$_j$ and the auxiliary nodes associated with it on the evidence MHT and requests the evidence chain data. If the data deletion on the evidence chain is valid, it indicates that the CSP has promised to delete the data, but there has been a data leakage. Otherwise, the user did not submit a deletion request for the data, and the CSP is not responsible for the data leakage.

Figure 3. Structure of the evidence chain



EXPERIMENT AND ANALYSIS

Accounting data in our paper were collected from the financial data of several large- and medium-sized companies, most of which are structured data. Due to the commercial privacy nature of these data, they are not publicly available. We only used it under supervision.

We used the Java Pairing-Based Cryptography (JPBC) library for experiments and selected Type A prime order elliptic curves from the JPBC library. Under the conditions of the Windows 10 operating system, 2.50 GHz, i5 processor, and 4 GB memory, we performed various operations that occurred during the audit process of this scheme for 10,000 times and took the average time cost.

Security Analysis

In traditional cloud accounting service models, the users upload their core data to the accounting CSP for storage, which results in the CSP having factual management and control over the user's core data, and the user effectively loses control over their core data. Core accounting data are the user's trade secret, and in the traditional accounting CSP model, the user core data managed and controlled by the CSP is at risk of leakage and destruction. For example, an unscrupulous CSP may sell a large amount of user data they possess for profit, or a CSP may trade with the users' competitors for huge profits, resulting in the leakage of user core data.

However, in the proposed accounting data security model based on blockchain encryption technology, the accounting data uploaded by the users to the blockchain network were encrypted, and the CSP needed to upload modification or deletion evidence to the evidence chain and accept the TPA's audit. The effectiveness of the proposed method in terms of security issues can be discussed from the following three aspects.

1) CSP uses an ECC encryption algorithm to encrypt the cloud user data before uploading it to the blockchain network. Due to the fact that ciphertext can only be decrypted through private keys, consensus nodes in blockchain networks cannot obtain data information from open blockchain networks.
2) During the accounting data audit process, the cloud users need to generate a one-time blockchain account address for this audit process. Assuming that the same user audits their own data multiple times, others cannot connect data ciphertext, user public key, and other information with actual blockchain users through the content of the audit contract. This measure can effectively protect user privacy and prevent other users from analyzing their identities through behavior analysis.
3) Moreover, even if the accounting data of the user are leaked, the thief cannot view the user's plaintext data because they do not have the user's private key, which effectively ensures the security and integrity of the accounting data.

As for the data transmission stage, in a traditional cloud accounting model, the users often overlook data encryption or only perform simple data encryption when transmitting data to the cloud. This makes it easy for malicious attackers to intercept, tamper with, or delete user data during the upload process, and the security and integrity of the data cannot be effectively guaranteed. However, in the proposed model, because the attacker lacked the user private key during the transmission process, they cannot obtain the plaintext information of the transmitted data and cannot tamper with the data during transmission. In addition, because the users store hash tags in both the blockchain network and their local computers and can entrust a TPA to check the data integrity of their cloud data based on the evidence chain, it can effectively guarantee the security of the user data during transmission over the Internet.

In the proposed cloud accounting data security model, both the users and accounting CSP can entrust the TPA to check the security and integrity of user data. When a user modifies data information, the CSP will also recalculate the updated data block tag to keep in sync with user data. Therefore, in the proposed security model, there was a higher level of mutual trust between users and accounting CSP than in the traditional cloud accounting model.

Efficiency Analysis

Firstly, the functions of our proposed scheme were compared with existing schemes (Wang et al., 2020; Wei et al., 2020; Sun et al., 2020; Mishra et al., 2022). Then, the computational overhead of different schemes was analyzed. Finally, the efficiency of different schemes was verified through simulation experiments. The comparison of the functions of each scheme is shown in Table 1. Among them, the methods in Wang et al. (2020) and Wei et al. (2020) use bilinear mapping technology to verify evidence from the CSP, but only the method in Wei et al. (2020) describes the dynamic update function in detail. Compared with our proposed method, only the method in Mishra et al. (2022) implements privacy protection for the CSP.

Currently, public dynamic auditing schemes all transfer audit proofs to other entities, such as TPA, to alleviate the users' auditing burden. However, this process may involve collusion between the CSP and TPA to falsify audit results. Since existing schemes cannot fully guarantee the integrity and correctness of data stored on the CSP, our proposed scheme aimed to resist substitution, forgery, and deletion attacks while also resisting collusion and replay attacks, enhancing the security of the auditing verification mechanism. As shown in Table 1, since methods in Wei et al. (2020), Sun et al. (2020), and Mishra et al. (2022) all introduce semitrusted TPA, although blind audit proofs are used to prevent the TPA from stealing data privacy, there is still a security risk of collusion between the CSP and TPA. Compared with other schemes, our proposed scheme can achieve dynamic update operations, fair arbitration of audit results, and data privacy protection for the CSP, and had a more comprehensive set of functions.

Table 2 provides the definitions of various operation types. The computational complexity of the proposed method was calculated as follows: let n denote the total number of data blocks, c denote the number of challenged data blocks, and select a multiplication cyclic group $G1$ and an additive cyclic group $G2$ on elliptic curves. In the TG phase, the user calculates $K = k \cdot g \in G$, $S_i = k^{-1}m_i + k^{-1}dR\mathrm{H}(\eta_i)$, with a total computational overhead of $nH + nA_{Z_p} + 3nM_{Z_p} + M_{G2}$ (Wei et al., 2020). In the PG phase, the CSP calculates $S = \sum_{j=1}^{c} S_{i_j}$, with a computational overhead of cM_{G2}. In the PV phase, the TPA calculates $\rho = \sum_{j=1}^{c} m_{i_j}$, $K = S^{-1} \cdot [\rho \cdot g + \sum_{j=1}^{c} \mathrm{H}(\eta_j) \cdot R \cdot P]$, with a computational overhead of $cH + 2cA_{Z_p} + M_{Z_p} + 3M_{G2}$. We compare the computational complexity of this scheme with that of other schemes in Table 3. It can be seen that in the TG and PG phases, the computational overhead of the proposed scheme was only slightly higher than that of reference Wang et al. (2020), and much lower than that of the other schemes. In the PV phase, the computational overhead of the proposed scheme was lower than that of other schemes.

Table 1. Function provided comparison

Schemes	ECC	Bilinear Map	Dynamic Updating	CSP Privacy	TPA Privacy	Arbitration Fairness
Wang et al. (2020)	×	✓	×	×	—	×
Wei et al. (2020)	×	✓	✓	×	×	✓
Sun et al. (2020)	✓	×	×	✓	×	×
Mishra et al. (2022)	✓	×	✓	✓	×	✓
Proposed	✓	×	✓	✓	✓	✓

ECC: Ellipse Curve Cryptography; CSP: Content Security Policy ; TPA: Third Party Auditor
"Ö"means this scheme use the function; "×" means this scheme doesn't use the function. "—" means this function doesn't exist.

Table 2. Description of each operation

Symbol	Description	Symbol	Description
A_{Z_p}	Addition operation on Z_p field	M_{G2}	Multiply operation on $G2$
M_{Z_p}	Multiply operation on Z_p field	A_{G2}	Addition operation on $G2$
E_{Z_p}	Power operation on Z_p field	H	Hash operation mapping to Z_p
E_{G1}	Power operation on $G1$	H'	Hash operation mapping to $G1$
M_{G1}	Multiply operation on $G1$	P	bilinear map

Table 3. Overhead comparison during different phases

Schemes	TG (Tag Generation)	PG (Proof Generation)	PV (Proof Verify)
Wang et al. (2020)	$nH' + nM_{G1} + 2nE_{G1}$	$(c-1)M_{G1} + (c+1)E_{G1} + cH + (2c-1)M_{Z_p}$	$cM_{G1} + (2+c)E_{G1} + cH' + 2P$
Wei et al. (2020)	$nH' + 2nE_{G1} + nM_{G1} + nsM_{Z_p}$	$cE_{G1} + (c-1)M_{G1} + sA_{Z_p} + (2c-1)M_{Z_p}$	$cH' + cE_{G1} + (c-1)M_{G1} + 3P + sE_{Z_p} + (s-1)M_{Z_p}$
Sun et al. (2020)	$nH' + 4nM_{G2} + nH$	$(c+1)M_{Z_p} + (2c-2)A_{G2} + 2cM_{G2} + cA_{Z_p}$	$CH' + (c+3)M_{G2} + cM_{Z_p} + cA_{G2} + cA_{Z_p} + cH$
Mishra et al. (2022)	$nH + nA_{Z_p} + nM_{Z_p}$	$2cM_{Z_p} + (2c-2)A_{Z_p} + cE_{Z_p}$	$(2c+1)H + (2c-2)A_{Z_p} + 2cM_{Z_p} + 3M_{G2} + 3A_{G2}$
Proposed	$nH + nA_{Z_p} + 3nM_{Z_p} + M_{G2}$	cM_{G2}	$cH + 2cA_{Z_p} + M_{Z_p} + 3M_{G2}$

Simulation Results

To analyze the proposed cloud data auditing scheme and ensure the randomness of the experimental results, random data were generated as input for dynamic operations. The communication between the user and the CSP was established using the JXTA (Juxtapose) technique (He et al., 2020). The basic functions of the scheme were implemented using the Java programming language. It was assumed that the user initially stores 5,000 files on the CSP and performs 10,000 insertions, modifications, and deletions on the data. The average time overhead of each operation for different numbers of operations was calculated and analyzed as the final experimental result. All the results are shown in Figure 4.

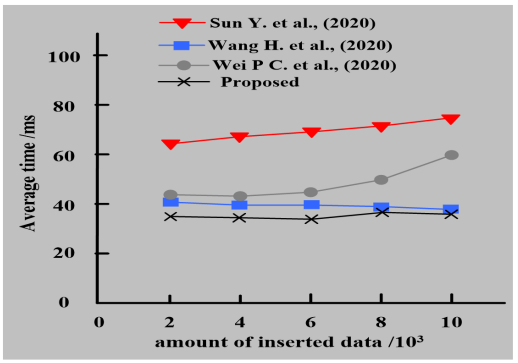
As shown in Figure 4(a), we can see that the proposed scheme and scheme in Wang et al. (2020) show a smaller increase in average time overhead for data insertion operations, while schemes in Wei et al. (2020) and Sun et al. (2020) show an increase in average time overhead with an increase in the amount of inserted data. Schemes in Wei et al. (2020) and Sun et al. (2020) achieve bad results because tags were generated with the true index of data blocks, which can lead to additional computational overhead if the true index changes. So, the proposed scheme was much better. This is

because the proposed approach used virtual indexing technology to avoid additional computational overhead caused by changes in the real index and improve update efficiency.

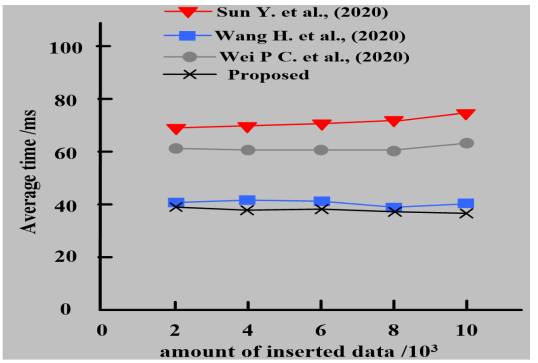
For deleting and modifying data, see Figure 4(b) and (c), the same approach as for the modification operation experiment was used, where data were deleted or modified after 10,000 insertion operations. The efficiency of data retrieval was affected by the different data structures used in the proposed scheme and the three comparison schemes. However, as the overall data volume decreased, the average time overhead showed a decreasing trend for all schemes. The reason why the proposed scheme achieved better results was during the deleting and modifying phase, only the data owner and cloud server needed to interact without introducing a trusted third party, further reducing the system communication and computing costs.

To better simulate real-life scenarios, it is assumed that the user has already stored 10,000 data records and randomly performs the same number of insertions, deletions, modifications, and audits. The average time overhead for each operation of every 1,000 operations was calculated to study the auditing time overhead required by the user in practical situations. The result is shown in Figure 5, and it can be observed that the scheme in Sun et al. (2020) has the highest time overhead for different user operation requests. The proposed scheme responded to a single user request within 30–45 ms and handled a thousand user requests within 3–5 s, which was an acceptable time overhead. Therefore, in real-life scenarios, the proposed scheme can reduce time overhead and efficiently handle various situations with irregular and varying user operations compared to other schemes, enhancing its practicality. Overall, the use of virtual indexing technology only requires interaction between the data owner and the cloud server during the deletion and validation stages without the need to introduce trusted third parties. These operations reduce time overhead and enhance practicality.

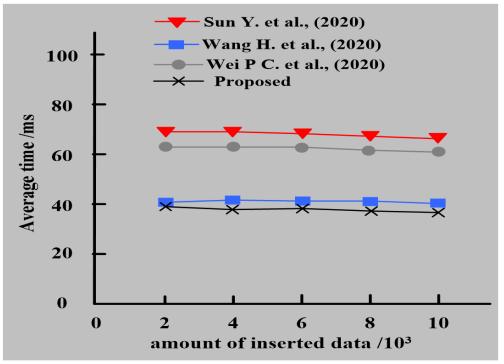
Figure 4. Time cost comparison of different schemes: (a) data insert operation, (b) data modify operation, (c) data delete operation



(a) data insert operation

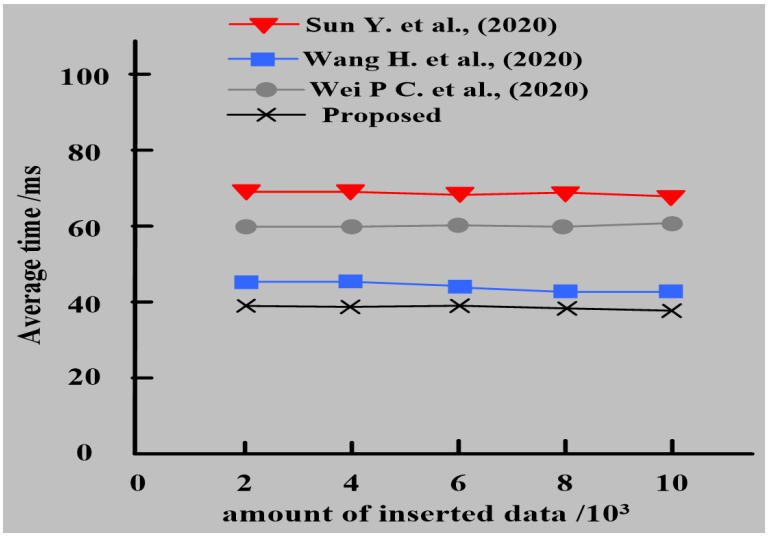


(b) data modify operation



(c) data delete operation

Figure 5. Time cost comparison of hybrid operations



CONCLUSION

A traditional cloud accounting model faces data security and data integrity issues, which seriously hinder the promotion of cloud accounting and the development of accounting digitalization. This paper proposes a cloud accounting data security model based on blockchain encryption technology. In this system, the user uploads encrypted accounting information data to the blockchain and stores the corresponding hash tag of the encrypted data in the blockchain. The experimental results show that the proposed system supports comprehensive privacy protection of data and has low computational overhead. Blockchain technology provides a new development path and idea for cloud accounting. The proposed model combines cloud accounting with blockchain technology for the first time, which helps to solve the data security and integrity issues faced by cloud accounting during its development. It can effectively ensure the security and integrity of the user accounting information and, to some extent, enhance the mutual trust between the accounting CSP and users, which is beneficial for the promotion and development of cloud accounting.

With the rise of Ethereum prices, the cost of blockchain gas is a problem that needs to be addressed. Currently, the transaction cost of each initiation is high, and we do not discuss this issue in this paper because we believe that using blockchain for auditing cloud accounting data is very worthwhile. But how to reduce gas costs while ensuring the security of cloud accounting data is still an important research direction. We will focus on it in the future. In addition, we will also focus more on the structured features of accounting data regarding privacy issues.

AUTHOR NOTE

The data used to support the findings of this study are included within the article.

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding for this paper was provided by the Application of Financial Big Data in the Demonstration Course of Internal Audit (Project Number: 221000512103727).

Correspondence concerning this article should be addressed to Congcong Gou, College of Finance and Economics Management, Sichuan University of Arts and Science, Dazhou, Sichuan, 635000, China. Email: gccghq@126.com.

REFERENCES

Gad, A. G., Mosa, D. T., Abualigah, L., & Abohany, A. A. (2022). Emerging trends in blockchain technology and applications: A review and outlook. *Journal of King Saud University-Computer and Information Sciences*, *34*(9), 6719–6742. doi:10.1016/j.jksuci.2022.03.007

Gudeme, J. R., Pasupuleti, S. K., & Kandukuri, R. (2019). Review of remote data integrity auditing schemes in cloud computing: Taxonomy, analysis, and open issues. *International Journal of Cloud Computing*, *8*(1), 20–49. doi:10.1504/IJCC.2019.097893

He, Q., Jiang, B., Cheng, D., & Liang, R. (2020, August 6–7). *BPS-VSS: A blockchain-based publish/subscribe video surveillance system with fine grained access control* [Conference session]. Blockchain and Trustworthy Systems: Second International Conference, BlockSys 2020, Dali, China. https://link.springer.com/chapter/10.1007/978-981-15-9213-3_20

Huang, L., Zhou, J., Zhang, G., & Zhang, M. (2020). Certificateless public verification for data storage and sharing in the cloud. *Chinese Journal of Electronics*, *29*(4), 639–647. doi:10.1049/cje.2020.05.007

Huttunen, J., Jauhiainen, J., Lehti, L., Nylund, A., Martikainen, M., & Lehner, O. M. (2019). Big data, cloud computing and data science applications in finance and accounting. *ACRN Journal of Finance and Risk Perspectives, 8,* 16–30. http://www.acrn-journals.eu/resources/SI08_2019b.pdf

Ionescu, L. (2019). Big data, blockchain, and artificial intelligence in cloud-based accounting information systems. *Analysis and Metaphysics, 18*, 44–49. https://www.proquest.com/openview/168c1444c00ac56315007972e3a4f888/1?pq-origsite=gscholar&cbl=136104

Jayaprakash, J. S., Balasubramanian, K., Sulaiman, R., Hasan, M. K., Parameshachari, B. D., & Iwendi, C. (2022). Cloud data encryption and authentication based on enhanced Merkle hash tree method. *Computers, Materials & Continua*, *72*(1), 519–534. doi:10.32604/cmc.2022.021269

Ming, Y., & Shi, W. (2019). Efficient privacy-preserving certificateless provable data possession scheme for cloud storage. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 122091–122105. doi:10.1109/ACCESS.2019.2938528

Mishra, R., Ramesh, D., Edla, D. R., & Mohammad, N. (2022). Fibonacci tree structure based privacy preserving public auditing for IoT enabled data in cloud environment. *Computers & Electrical Engineering*, *100*, 107890. doi:10.1016/j.compeleceng.2022.107890

Moll, J., & Yigitbasioglu, O. (2019). The role of internet-related technologies in shaping the work of accountants: New directions for accounting research. *The British Accounting Review*, *51*(6), 100833. doi:10.1016/j.bar.2019.04.002

Parast, F. K., Sindhav, C., Nikam, S., Yekta, H. I., Kent, K. B., & Hakak, S. (2022). Cloud computing security: A survey of service-based models. *Computers & Security*, *114*, 102580. doi:10.1016/j.cose.2021.102580

Parmoodeh, A. M., Ndiweni, E., & Barghathi, Y. (2023). An exploratory study of the perceptions of auditors on the impact on Blockchain technology in the United Arab Emirates. *International Journal of Auditing*, *27*(1), 24–44. doi:10.1111/ijau.12299

Rabaninejad, R., Attari, M. A., Asaar, M. R., & Aref, M. R. (2019). Comments on a lightweight cloud auditing scheme: Security analysis and improvement. *Journal of Network and Computer Applications*, *139*, 49–56. doi:10.1016/j.jnca.2019.04.012

Ramokapane, K. M., Rashid, A., & Such, J. M. (2016, October). *Assured deletion in the cloud: Requirements, challenges and future directions* [Conference session]. The 2016 ACM on Cloud Computing Security Workshop, 28 October, 2016, Vienna, Austria. doi:10.1145/2996429.2996434

Razaque, A., Frej, M. B. H., Alotaibi, B., & Alotalbi, M. (2021). Privacy preservation models for third-party auditor over cloud computing: A survey. *Electronics (Basel)*, *10*(21), 2721. doi:10.3390/electronics10212721

Singh, N., Lai, K., Vejvar, M., & Cheng, T. C. E. (2019). Data-driven auditing: A predictive modeling approach to fraud detection and classification. *Journal of Corporate Accounting & Finance*, *30*(3), 64–82. doi:10.1002/jcaf.22389

Sun, Y., Liu, Q., Chen, X., & Du, X. (2020). An adaptive authenticated data structure with privacy-preserving for big data stream in cloud. *IEEE Transactions on Information Forensics and Security*, *15*, 3295–3310. doi:10.1109/TIFS.2020.2986879

Wahhab, A. M. A., Alajeli, E. H. A., & Jawad, B. H. (2022). The role of internal audit in analyzing and auditing big data and its impact on the quality financial reports. *Technium Social Sciences Journal, 32*, 669–679. https://www.researchgate.net/publication/361193706_The_Role_of_Internal_Audit_in_Analyzing_and_Auditing_Big_Data_and_its_Impact_on_the_Quality_Financial_Reports

Wang, F., Xu, L., Choo, K.-K. R., Zhang, Y., Wang, H., & Li, J. (2019). Lightweight certificate-based public/private auditing scheme based on bilinear pairing for cloud storage. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 2258–2271. https://ieeexplore.ieee.org/document/8936952. doi:10.1109/ACCESS.2019.2960853

Wang, H., Qin, H., Zhao, M., Wei, X., Shen, H., & Susilo, W. (2020). Blockchain-based fair payment smart contract for public cloud storage auditing. *Information Sciences*, *519*, 348–362. doi:10.1016/j.ins.2020.01.051

Wei, P. C., Wang, D., Zhao, Y., Tyagi, S. K. S., & Kumar, N. (2020). Blockchain data-based cloud data integrity protection mechanism. *Future Generation Computer Systems*, *102*, 902–911. doi:10.1016/j.future.2019.09.028

Xu, M., Chen, X., & Kou, G. (2019). A systematic review of blockchain. *Financial Innovation*, *5*(1), 1–14. doi:10.1186/s40854-019-0147-z

Xue, J., Xu, C., Zhao, J., & Ma, J. (2019). Identity-based public auditing for cloud storage systems against malicious auditors via blockchain. *Science China. Information Sciences*, *62*(3), 1–16. doi:10.1007/s11432-018-9462-0

Yang, W. (2022). *ECC, RSA, and DSA analogies in applied mathematics* [Conference session]. Proceedings of the International Conference on Statistics, Applied Mathematics, and Computing Science (CSAMCS 2022), 25-27 November 2022, Nanjing, China. doi:10.1117/12.2628013

Yu, B., Li, X., & Zhao, H. (2020). Virtual block group: A scalable blockchain model with partial node storage and distributed hash table. *The Computer Journal*, *63*(10), 1524–1536. doi:10.1093/comjnl/bxaa046

Zhe, J., Kai, Z., Liangliang, W., & Ning, J. (2022). Forward secure public-key authenticated encryption with conjunctive keyword search. *The Computer Journal*, *66*(9), 2265–2278. doi:10.1093/comjnl/bxac075

Congcong Gou (1982-), male, born in Pingchang, Sichuan. Associate professor, whose main research direction is financial accounting and auditing.