Received July 1, 2020, accepted July 24, 2020, date of publication July 30, 2020, date of current version August 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013153

Decentralized and Privacy-Preserving Public Auditing for Cloud Storage Based on Blockchain

YING MIAO[1], QIONG HUANG[1,2], MEIYAN XIAO[1], AND HONGBO LI[1]
[1]College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China
[2]Guangzhou Key Laboratory of Intelligent Agriculture, Guangzhou 510642, China

Corresponding author: Qiong Huang (qhuang@scau.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872152, in part by the Major Program of Guangdong Basic and Applied Research under Grant 2019B030302008, and in part by the Science and Technology Program of Guangzhou under Grant 201902010081.

ABSTRACT Cloud storage systems provide a flexible, convenient and friendly way for users to outsource data. However, users lose control of their data once outsourcing them to the cloud. *Public auditing* was introduced to ensure data integrity, in which a third-party auditor (TPA) is delegated to execute auditing tasks. In general, TPA generates and sends challenge information to the cloud server (CS), which proves data possession accordingly. However, the TPA may not perform public auditing protocol honestly or may even collude with CS to deceive users. Some existing public auditing schemes utilize blockchain to resist against the malicious TPA. However, the CS may guess the challenge messages and there is a risk that users' information may be leaked to the TPA during the process of auditing. In this paper, we propose a decentralized and privacy-preserving public auditing scheme based on blockchain (DBPA), in which a blockchain is utilized as an unpredictable source for the generation of (*random*) challenge information, and the auditor is required to record the audit process onto the blockchain. Due to the characteristics of blockchain, users can check the audit results publicly. Moreover, zero-knowledge proof is used in DBPA to protect user's privacy during the audit process so that the response information returned by the CS does not leak information about user's data. Security analysis and performance evaluation show that DBPA is secure and efficient.

INDEX TERMS Decentralization, privacy preserving, public auditing, cloud storage, blockchain.

I. INTRODUCTION

As valuable resources, data are generated in various of ways whenever and wherever. Massive data at local storage cause a series of difficulty in management. To reduce heavy burden of data storage and maintenance in local storage, many users choose to outsource their data into cloud [1]. As an excellent tool, cloud brings tremendous benefits and convenience to our life. At the same time, concerns about data security emerge [2]–[4]. After outsourcing to the cloud, users lose control of their data, and data on the cloud may not be secure and may suffer from a various of attacks [5], [6]. On one hand, the cloud server (CS) may behave illegally on the outsourced data, e.g. retrieve or steal user data to make profit. On the other hand, the CS might corrupt or delete user data to save storage space and reduce maintenance expense. Thus, data confidentiality, integrity and availability are violated. Furthermore, the cloud may suffer from single point of failure when hardware fails. Unfortunately, the CS may try to hide data accidents in order to maintain its good reputation. According to [7], the most critical threats of cloud storage is data integrity and privacy leakage. In recent years, a series of cloud storage security incidents have drawn highly attention of the public.[1] Take *Under Armour* data breach as an example. Their health and fitness tracking App ''MyFitnessPal'' was attacked by hackers, affecting about 150 million users at the end of February, 2018. The leaked information includes usernames, email addresses, passwords and etc. Therefore, it is of great importance to guarantee the integrity and privacy of cloud data.

In recent years, many works on cloud data integrity and privacy protection have been reported. Firstly, a bunch of public verification schemes have been proposed in order to improve the integrity of cloud data [8]–[17]. Public

The associate editor coordinating the review of this manuscript and approving it for publication was Junggab Son.

[1] https://blog.360totalsecurity.com/en/2018-cybersecurity-report/

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

verification enables a user or a delegated TPA to check the data integrity [18]. An auditor usually checks the data on schedule, and informs the user of data exception if the check fails. Secondly, as a third party, the auditor should not know extra information about user data in order to protect their privacy. Thus, many privacy-preserving public auditing schemes have been proposed, such as [19]–[22] and etc. Meanwhile, with the development of blockchain technology and its advantage in decentralization, trustless consensus, tamper proof and traceability, many researchers have studied decentralized public auditing schemes against the malicious auditor. A series of literatures can be found in [23]–[28].

In most public verification schemes, the auditor is generally assumed to be honest and reliable. However, it is a strong assumption, as the auditor may not be so reliable as expected, i.e. it may compromise and collude with the CS to hide data corruption incidents. However, few recent literatures take a malicious auditor into consideration. In addition, most recent schemes secure against the malicious auditor are based on a centralized and trustworthy third party [29]–[31]. Blockchain-based public auditing schemes provide a good solution to the problem of resisting against the malicious auditor. But the consensus mechanism brings some concerns as well, since a malicious cloud server could take use of public messages to infer auditing information before the auditor sends challenge messages. However, recent literatures do not consider the issue. In this work we try to solve the problem that the cloud server may guess challenge messages ahead of time in decentralized public auditing schemes, and in the meanwhile, to guarantee that the TPA does not know extra information of user data for the sake of privacy protection.

A. RELATED WORK

1) BLOCKCHAIN

Blockchain is increasingly recognized as an outstanding tool in designing decentralized protocols. The concept traces back to the original whitepaper of Nakamoto [32] published in 2008, in which he applied blockchain as the core component of the famous cryptocurrency named Bitcoin. Roughly, blockchain is a distributed database that is maintained by multiple nodes and increases a list of ordered records in the shape of blocks without requiring trust among nodes [33]. There are many mature blockchain systems, such as Ethereum [34], Litecoin [35] and etc.

As a decentralized system, blockchain adopts the decentralized consensus mechanism without a third-party trusted authority. There are four major consensus mechanisms [36], *Proof of Work* (PoW), *Proof of Stake* (PoS), *Practical Byzantine Fault Tolerance* (PBFT) and *Delegated Proof of Stake* (DPoS). The two popular cryptocurrencies, Bitcoin and Ethereum, use PoW mechanism, which aims to prove the credibility of data by solving puzzles computationally hard to compute but easy to verify. A blockchain system includes miners whose task is to compute a nonce satisfying the following relation:

$$\begin{aligned}&\mathrm{SHA256}(\mathrm{PrevBlockHash}\|\mathrm{Nonce}\\&\qquad\qquad\|tx_1\|tx_2\|\cdots\|tx_n) < \mathrm{target},\end{aligned}\tag{1}$$

where the target can be adjusted to change the difficulty of PoW puzzles.

Blockchain systems can be classified into three types: *public blockchain*, *consortium blockchain* and *private blockchain* [37], according to the managed data, availability of data and actions performed by a user. A private blockchain is authorized by an owner, while a consortium blockchain is authorized by a consortium organization in which all participants do not necessarily trust each other. A public blockchain has no threshold for users, and anyone can join or leave the blockchain without getting permission from centralized or distributed authorities. Furthermore, different blockchains have their advantages in different applications. Private blockchain is faster, and public blockchain is more open and transparent. In general, blockchain has its characteristics and advantages in *decentralization and anonymity*, *non-modifiability and unforgeability* and *traceability and irreversibility* [38]. Blockchain has been successfully applied in various of areas, such as electronic medical records [39], public auditing [27], energy tracing [40], decentralized supply chain management [41] and etc.

2) PUBLIC VERIFICATION

In order to ensure the integrity of data stored on an untrusted cloud server, Juels *et al.* [8] proposed the notion of *Proof of Retrievability* (POR), which relies on indistinguishable blocks as sentinels to detect data corruption. However, their scheme does not support dynamic numbers of POR queries, nor consider the public auditing model. Ateniese *et al.* [9] firstly proposed the *Provable Data Possession* (PDP) model which utilizes homomorphically verifiable tags and a kind of challenge-response protocol. However, they did not provide a security proof of their protocol in the paper. Following the work of POR and PDP, many extended public auditing schemes have been proposed for catering to different requirements, such as [10], [11], [11], [20] and etc. However, these schemes are mainly based on public key infrastructure (PKI). Due to the limitation of communication resources and large amount of data, an auditor is delegated to audit the integrity of outsourced data. Key management including revocation, storage, distribution and verification is cumbersome and costly in PKI-based auditing systems.

To avoid heavy computation and communication cost of managing certificates in public auditing schemes, Zhao *et al.* [21] proposed the first identity-based public auditing (IBPA) scheme. After that, a series of IBPA schemes were proposed, such as [14], [15], [22], [31], [42] and etc. These schemes assumes the existence of a fully trusted TPA, which is somewhat strong. If the auditor is dishonest or even malicious, it may collude with the cloud server to cover data loss and may not perform the auditing honestly, which could not be detected by users.

3) DECENTRALIZED PUBLIC AUDITING

How to improve the credibility of TPA is increasingly attracting attentions in recent literatures [43]. Especially, thanks to its outstanding properties of decentralization, openness and non-modifiability, blockchain technology provides a good solution to deal with the aforementioned problems [44], [45].

In 2014, Armknecht *et al.* [23] firstly proposed a public verification scheme secure against the malicious auditor, which uses Bitcoin blockchain as a secure source of time-dependent pseudorandomness provider and uses the hash of the latest block based on the time t and security parameters to generate challenge messages. Owing to the unique and unpredictable bits extracted from Bitcoin blocks, Armknecht's scheme avoids to generate biased challenge messages to deceive the user. However, a new block is generated in 10 minutes on average in Bitcoin, and the cloud server may know the challenge information ahead of time.

Following the work of Armknecht *et al.*, a series of decentralized public auditing schemes secure against the malicious auditor were proposed. To name a few, Zhang *et al.* [24] proposed an identity-based public integrity-verification scheme which uses the latest Bitcoin block hash based on the time t to generate challenge messages.

Besides, Zhang *et al.* did not take the user privacy into consideration. Afterwards, Zhang *et al.* [25] proposed another public verification scheme. The new scheme adopts a random masking technique to hide linear relationship between proof information and data blocks, which resists against external adversaries and protects privacy information of users.

In order to solve this problem, Zhang *et al.* [26] proposed a blockchain-based public integrity verification scheme which uses a series of successive Ethereum block hashes based on the timestamp t instead of the latest block hash to generate challenge messages. Their core technique has been applied in another scheme [46] which aims to add an accurate time-stamp for outsourced data. However, Zhang *et al.*'s scheme [26] does not take the protection of user privacy into consideration either. Xue *et al.* [27] proposed an identity-based public auditing scheme which uses the latest Bitcoin block nonce to generate challenge messages. Their scheme prevents a malicious auditor from generating specified challenge messages. Yu *et al.* [28] proposed a decentralized data auditing scheme which uses a series of successive blocks in consortium blockchain to generate challenge messages. Their scheme could prevent a malicious auditor from colluding with the cloud server to generate some specified challenge messages and thus deceiving users. However, the block numbers used in consortium blockchain is controlled by the auditor, which means the challenge messages are still controlled by auditor to some extent.

All the schemes above take use of blockchain as the pesudorandom seed to generate challenge messages. However, they failed to consider the issue of challenge messages guessing attacks launched by the cloud server. According to the PoW mechanism, a new block is generated every 10 minutes on average, which gives the cloud server a chance to guess the challenge messages ahead of time and tries to prepare for covering data loss during the period.

B. OUR CONTRIBUTIONS

In this paper, we propose a decentralized public auditing solution targeting specifically to provide security against challenge messages guessing attacks and privacy protection for users during the process of auditing. Our contributions in the paper can be summarized as follows.

- We propose a decentralized privacy-preserving public data integrity auditing scheme based on blockchain, named DBPA, in which the challenge message is generated based on the latest successive block hashes and a random seed chosen by the TPA. Therefore, a malicious cloud server is unable to guess the challenge message ahead of time any more.
- We utilize zero-knowledge proof (ZKP) to protect user privacy in DBPA. Concretely, instead of returning the aggregated tag (computed according to the challenge message), the cloud server returns a blinded version of the tag and provides a ZKP to show the correctness of the tag. If the proof passes the verification, the TPA learns nothing else but the correctness of user data. Thus, privacy of user data is guaranteed.
- Our DBPA scheme employs the PoW consensus mechanism and utilizes blockchain to record the audit results, which is public, decentralized and unforgeable. Any malicious behaviors and incorrect results can be easily detected. Therefore, the audit results could be trusted.
- We show that our DBPA scheme is secure in the random oracle model based on the intractability of Computational Diffie-Hellman problem and Discrete Logarithm problem. Experimental results show that our scheme is efficient and performs well.

C. ORGANISATION

The remainder of the paper is organized as follows. We introduce the preliminaries and definitions in Sections II and III, respectively. In Section IV, we describe the construction of our DBPA scheme. Then, we analyze the security of our scheme in Section V. We provide a performance evaluation of our scheme in Section VI. Finally, we summarize the work in Section VII.

II. PRELIMINARIES

A. BASIC TOOLS AND HARD PROBLEMS

1) BILINEAR MAPS

Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p, respectively, g be a generater of $\mathbb{G}_1$. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ be a bilinear map with the following properties: (1) *Bilinearity*: for all $U, V \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$, $e(U^a, V^b) = e(U, V)^{ab}$; (2) *Computability*: for any

$U, V \in \mathbb{G}_1$, $e(U, V)$ could be efficiently computed; and (3) *Non-degeneracy*: $e(g, g) \neq 1_T$, where 1_T is the identity element of $\mathbb{G}_T$.

2) DISCRETE LOGARITHM(DL) ASSUMPTION

Given $g, g^a \in \mathbb{G}_1$ as input, where a is a random element of $\mathbb{Z}_p$, there is no probabilistic polynomial-time (PPT) adversary $\mathcal{A}_{\mathcal{DL}}$ which could output a with non-negligible probability. We denote it as

$$\Pr[\mathcal{A}_{DC}(g, g^a) = a : g \leftarrow \mathbb{G}_1, a \leftarrow \mathbb{Z}_p] \leq \epsilon,$$

where ϵ is a negligible function.

3) COMPUTATIONAL DIFFIE-HELLMAN(CDH) ASSUMPTION

Given $g, g^a, g^b \in \mathbb{G}_1$, where a, b are randomly chosen from $\mathbb{Z}_p$, no PPT adversary could calculate g^{ab} with non-negligible probability. We denote it as

$$\Pr[\mathcal{A}_{CDH}(g, g^a, g^b) = g^{ab} : g \leftarrow \mathbb{G}_1, a, b \leftarrow \mathbb{Z}_p] \leq \epsilon.$$

B. BLOCKCHAIN STRUCTURE

Figure 1 shows the structure of blockchain [34]. Each block contains a hash pointer that points to its previous block. *BlockHash* denotes the hash value of current block. *PrevBlockHash* denotes the hash value of the previous block. *Nonce* denotes the solution to the PoW puzzle shown in Eq. 1. *Timestamp* denotes the generation time of the block. *Tx* denotes the transaction, and all the transactions are authenticated by Merkel tree root (denoted by *MerkelRoot*). A transaction contains a payer's account address, a payee's account address, data and the payer's signature.

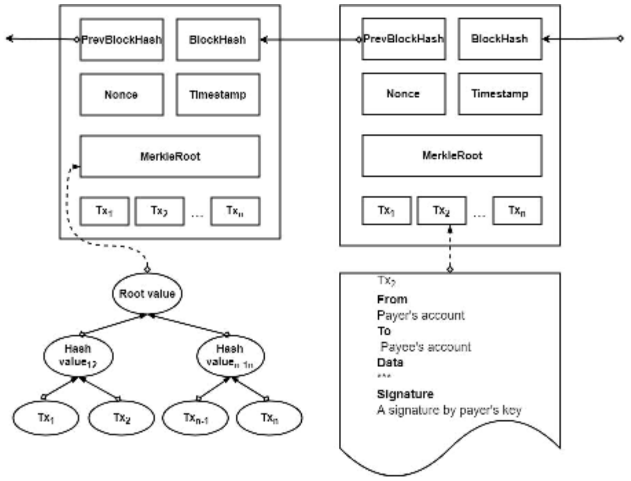


FIGURE 1. Data structure of blockchain.

It is commonly believed that *BlockHash* is random and unpredictable. In our scheme, a series of successive *BlockHash*, e.g. $\{Bl_{t-\varphi+1}, Bl_{t-\varphi+2}, \cdots, Bl_t\}$, are used to generate an unpredictable challenge message, where φ is the number of blocks used to confirm a transaction. For instance, we set $\varphi = 6$ in Bitcoin, and set $\varphi = 12$ in Ethereum. In addition, t denotes the agreed verification time, and Bl_t denotes the hash of the latest block generated at or before time t, since the latest block may not appear exactly at time t.

III. DECENTRALIZED AND PRIVACY-PRESERVING PUBLIC AUDITING SCHEME

A. SYSTEM MODEL

Figure 2 shows the architecture of our decentralized and privacy-preserving public auditing scheme based on blockchain. In the scheme, there are four different entities, i.e., key generation center (KGC), cloud server (CS), data user ($\mathcal{U}$) and a third-party auditor (TPA).

- *Key generation center* is an authority, whose task is to generate system parameters and partial private key for users according to their identity.
- *Cloud server* provides cloud storage services. It not only has enough storage space, but also possesses amount of computing power.
- *Data user* is the data owner, who outsources data to the cloud and delegates the TPA to check the data integrity. He checks the auditor's behavior via the blockchain.
- *Third-party auditor* detects the data integrity periodically and checks if there is any data corruption. TPA uploads the verification results to the blockchain after verifying the proof information from the CS.

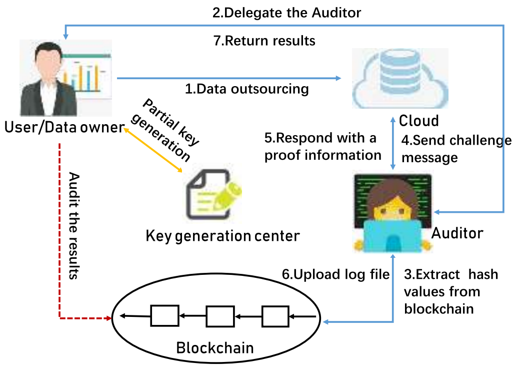


FIGURE 2. System model of DBPA.

DBPA system works as follows. Firstly, $\mathcal{U}$ outsources his data into cloud, and delegates a TPA to help him execute the auditing tasks. After receiving a delegation, the TPA utilizes the latest public blockchain information to generate a challenge message and sends it to CS, which then generates a proof accordingly to confirm the data possession. If the proof from CS passes the verification, the TPA generates a log file to record the audit result, and uploads the file to the blockchain. Finally, $\mathcal{U}$ checks the auditing results according to the log file on the blockchain.

B. DEFINITION

DBPA consists of six algorithms, **Setup**, **Extract**, **Store**, **Audit**, **LogGen** and **CheckLog**, defined as below.

Setup is run by the KGC to generate a master secret key α and public parameters which are used in the following algorithms.

Extract is run by the KGC to generate secret key of a user according to its identity $\mathcal{ID}_{\mathcal{U}}$.

Store is run by $\mathcal{U}$ to outsource its data to the CS. The user needs generate verification tags that enable a TPA to check the data integrity. Furthermore, the CS needs confirm that the data is uploaded correctly.

Audit is run between the TPA and CS to check the data integrity. It consists of three sub-algorithms, including challenge generation (*ChaGen*), proof generation (*ProGen*) and proof verification (*ProVer*).

LogGen is run by the TPA to generate a log file to record the auditing result. The log file will be uploaded to the blockchain.

CheckLog is run by the user to audit the TPA's behavior by checking the validity of auditing records stored in the log file on the blockchain.

C. SECURITY THREATS

We consider threats from two entities, e.g. cloud server and TPA.

- **Semi-trusted cloud server**. The CS is assumed to be semi-trusted. It may be dishonest and hide the incident of data corruption by forging a proof to deceive the TPA. It may also try to predict the challenge message ahead of the audit.
- **Misbehaving third-party auditor**. The TPA is assumed to be semi-trusted. It will fulfill its obligation of data audit for users, but may try to infer information about user data from the response information returned by the CS.

We also consider the case in which the CS and TPA may collude together to generate false audit results to deceive the data user.

D. DESIGN GOALS

In this paper we target to design a secure and privacy-preserving public auditing scheme for cloud data storage. Namely, our scheme should achieve the following goals.

Authenticity. Data corruption could be detected with overwhelming probability. That is, the CS could not pass the auditing if there is any data loss or modification. We follow the model in [13], [47], and consider the following game in which the data owner is viewed as a challenger C and the CS is viewed as an adversary $\mathcal{A}$.

1) **Setup phase**. C generates the master secret key and system public parameters pp, and sends pp to $\mathcal{A}$.
2) **Query phase**. $\mathcal{A}$ makes the following queries to C.
    a) *Extract Queries:* $\mathcal{A}$ queries for the private key of user with identity ID_U. C runs the Extract algorithm to generate the private key $sk_{\mathcal{U}}$, and returns it to $\mathcal{A}$.
    b) *Store Queries:* $\mathcal{A}$ queries for the tags of a file M of a user ID_U. C uses the private key $sk_{\mathcal{U}}$ to run the Store algorithm to generate file tags, and returns the tags to $\mathcal{A}$.
3) **Challenge phase**. In this phase, $\mathcal{A}$ submits an identity ID_U which has not appeared in extract queries before. C generates a challenge message *chal* to $\mathcal{A}$, which refers to at least one data block whose tag has not been given to $\mathcal{A}$.
4) **Forgery phase**. $\mathcal{A}$ generates a data possession proof *proof* for the data blocks indicated by *chal*. If *proof* can pass the verification with non-negligible probability, we say that the adversary $\mathcal{A}$ succeeds in the game.

The security model above indicates that, if the cloud server does not keep all the data blocks challenged by C, it is unable to generate a valid proof *proof* to pass the verification.

1) DECENTRALIZED CHALLENGE MESSAGES GENERATION

In order to prohibit a misbehaving auditor from colluding with the CS and generating an audit result ahead of time schedule, the challenge message should not depend solely on either the user or the auditor. Furthermore, the auditor should provide incontrovertible evidence which should not be pre-defined or predicted but can be checked and verified publicly.

2) PRIVACY PRESERVATION

Except the verification result of data audit, the TPA should be unable to infer any other information about user data from the proofs collected during the auditing process.

3) TRACEABILITY

In order to ensure the correctness and integrity of the outsourced data, the audit process should be traceable so that any malicious behavior of the TPA could be detected.

IV. OUR DBPA SCHEME

In this section, we describe our DBPA scheme, which utilizes a blockchain. Assume that a user $\mathcal{U}$ has an identity $\mathcal{ID}_{\mathcal{U}}$ and that φ new blocks are needed to confirm a transaction in the blockchain (see Section II-B). Our scheme works as below.

A. SETUP

Given security parameter 1^{ℓ}, the KGC generates system parameters as follows:

- choose a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$ and $\mathbb{G}_T$ are multiplicative groups with the same prime order p, and g is the generator of $\mathbb{G}_1$;
- choose a random $\alpha \in Z_p$ as the master key and set $P_M = g^{\alpha}$;
- choose a pseudorandom function π_1: $\mathcal{K}_1 \times [1, n] \rightarrow [1, n]$, and a pseudorandom permutation π_2: $\mathcal{K}_2 \times [1, n] \rightarrow Z_p$, where n is the (maximal) number of file blocks, $[1, n]$ is the set $\{1, 2, \cdots, n\}$, and $\mathcal{K}_1, \mathcal{K}_2$ are the key spaces of π_1 and π_2, respectively;

- choose cryptographic hash functions $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p$, $H_i : \{0, 1\}^* \rightarrow \mathbb{G}_1$ for $i = 1$ to 4, $h_1 : \{0, 1\}^* \rightarrow \mathcal{K}_1$, and $h_2 : \{0, 1\}^* \rightarrow \mathcal{K}_2$;
- output the system public parameter $pp = \{\mathbb{G}_1, \mathbb{G}_T, e, g, P_M, \varphi, H, H_1 \sim H_4, h_1(\cdot), h_2(\cdot), \pi_1, \pi_2\}$, and keep α secret.

B. EXTRACT

The KGC generates private key for $\mathcal{U}$ as follows:

- compute $Q_{U,0} = H_1(\mathcal{ID}_{\mathcal{U}}, 0)$ and $Q_{U,1} = H_1(\mathcal{ID}_{\mathcal{U}}, 1)$;
- compute $D_{U,0} = Q_{U,0}^{\alpha}$ and $D_{U,1} = Q_{U,1}^{\alpha}$.

The KGC sends $D_{U,0}, D_{U,1}$ to $\mathcal{U}$, which checks if $e(D_{U,0}, g) = e(Q_{U,0}, P_M)$ and $e(D_{U,1}, g) = e(Q_{U,1}, P_M)$ hold. If not, $\mathcal{U}$ rejects; otherwise, it chooses a random $x_u \in Z_p$ and computes $PK_U = g^{x_u}$. The private key of $\mathcal{U}$ is $sk_u = \{x_u, D_{u,0}, D_{u,1}\}$, and the public key is $\{PK_U, \mathcal{ID}_{\mathcal{U}}\}$.

C. STORE

$\mathcal{U}$ divides its data file M into n blocks, e.g. $M = \{m_i\}_{1 \le i \le n}$, randomly chooses an element $name \in Z_p$ for file naming and a one-time number $r_1 \in Z_p$, and computes $\tau = H(name \| n \| r_1 \| PK_U)$. $\mathcal{U}$ then generates file tags as follows:

- randomly choose $r_2 \in Z_p$, and compute $R = g^{r_2}$, $V = H_3(r_1)$ and $W = H_4(r_1)$;
- for each $i \in [1, n]$, compute $T_i = H_2(i \| \tau \| R)$, and $S_i = (D_{U,0} \cdot V^{x_u})^{m_i} \cdot (D_{U,1}, W^{x_u})^{H(i \| \tau \| R)} \cdot T_i^{r_2}$, where S_i is the file tag for data block m_i;
- upload $F = \{M, \{S_i\}_{i=1}^{n}, R, r_1\}$ to the CS.

After receiving F, the CS computes $\tau = H(name \| n \| r_1 \| PK_U)$, and verifies the correctness of the data by checking if

$$e(\prod_{i=1}^{n} S_i, g) = e(\prod_{i=1}^{n} (Q_{U,0}^{m_i} Q_{U,1}^{h_i}), P_M) \cdot e(\prod_{i=1}^{n} (V^{m_i} W^{h_i}), PK_U) \cdot e(\prod_{i=1}^{n} T_i, R), \quad (2)$$

where $h_i = H(i \| \tau \| R)$. The CS accepts F if the equation holds, and rejects otherwise.

D. AUDIT

This algorithm consists of the following sub-algorithms.

1) *ChalGen*. The TPA chooses a random $r_3 \in Z_p$ and $c \leftarrow [1, n]$, and sends the challenge message $chal = (\varphi, t, r_3, c)$ to the CS, where t is the current timestamp.
2) *ProGen*. After receiving $chal$ from the TPA, the CS works as follows:
   - extract $\{Bl_{t-\varphi+1}, Bl_{t-\varphi+2}, \cdots, Bl_t\}$ from the blockchain based on t and φ, and compute
   $$k_1 = h_1(Bl_{t-\varphi+1} \| Bl_{t-\varphi+2} \| \cdots \| Bl_t \| r_3)$$
   and
   $$k_2 = h_2(Bl_{t-\varphi+1} \| Bl_{t-\varphi+2} \| \cdots \| Bl_t \| r_3);$$
   - compute $i_\xi = \pi_1(k_1, \xi)$ and $v_{i_\xi} = \pi_2(k_2, \xi)$ for $\xi = 1, 2, \cdots, c$;
   - compute $S = \prod_{\xi=1}^{c} S_{i_\xi}^{v_{i_\varsigma}}$ and $\mu = \sum_{\xi=1}^{c} v_{i_\xi} m_{i_\xi}$;
   - randomly select $\rho \in Z_p$, compute
   $$T_M = P_M^{\rho}, T_U = PK_U^{\rho}, T_R = R^{\rho},$$
   and $A = e(S^{\rho}, g)/(e(Q_{U,0}^{\mu}, T_M) \cdot e(V^{\mu}, T_U))$;
   - randomly select $\theta \in Z_p$, set $W = S^{\theta}$, and provide the following zero-knowledge proof (ZKP):
   $$\pi = \mathsf{ZKP} \times \Big\{(\rho, \mu, \theta) | T_M = P_M^{\rho} \wedge T_U = PK_U^{\rho} \wedge T_R = R^{\rho} \wedge e(W^{\rho}, g^{1/\theta}) e(Q_{U,0}^{-\mu}, T_M) e(V^{-\mu}, T_U) = A\Big\}.$$
   Concretely, the proof π is generated as follows:
     - randomly select $r_{\rho}, r_{\theta}, r_{\mu} \in Z_p$, and compute
     $$R_W = W^{r_\rho}, \quad R_\theta = g^{1/\theta},$$
     $$R_Q = Q_{U,0}^{-r_\mu}, \quad R_V = V^{-r_\mu},$$
     $$R_M = P_M^{r_\rho}, \quad R_U = PK_U^{r_\rho}, \quad R_r = R^{r_\rho};$$
     - compute
     $$c = H(R_W, R_\theta, R_Q, R_V, R_M, R_U, R_r);$$
     - compute $z_\rho = r_\rho + \rho c$, $z_\theta = c$, $z_\mu = r_\mu + \mu c$;
     - output $\pi = (c, z_\rho, z_\theta, z_\mu)$.

   The CS sends $proof = \{A, r_1, W, \pi\}$ to the TPA.
3) *ProVer*. Upon receiving $proof$, the TPA checks the data integrity as follows:
   - reject if either of the following equations fails to hold:
   $$e(\frac{W^{z_\rho}}{R_W}, R_\theta) e(\frac{Q_{U,0}^{-z_\mu}}{R_Q}, T_M) e(\frac{V^{-z_\mu}}{R_V}, T_U) = A^c,$$
   $$P_M^{z_\rho}/R_M = T_M^c,$$
   $$PK_U^{z_\rho}/R_U = T_U^c,$$
   $$R^{z_\rho}/R_r = T_R^c;$$
   - compute $\tau = H(name \| n \| r_1 \| PK_U)$, and
   $$k_1 = h_1(Bl_{t-\varphi+1} \| Bl_{t-\varphi+2} \| \cdots \| Bl_t \| r_3),$$
   $$k_2 = h_2(Bl_{t-\varphi+1} \| Bl_{t-\varphi+2} \| \cdots \| Bl_t \| r_3);$$
   - compute $i_\xi = \pi_1(k_1, \xi)$ and $v_{i_\xi} = \pi_2(k_2, \xi)$ for all $\xi = 1, 2, \cdots, c$;
   - check whether
   $$A = e(\prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi} h_{i_\xi}}, T_M) e(\prod_{\xi=1}^{c} W^{v_{i_\xi} h_{i_\xi}}, T_U) \cdot e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, T_R), \quad (3)$$
   where $h_{i_\xi} = H(i_\xi \| \tau \| R)$ and $T_{i_\xi} = H_2(i_\xi \| \tau \| R)$. The TPA rejects if Eq. (3) does not hold, and accepts otherwise.

TABLE 1. The log file f.

Challenge				Proof				Position in Blockchain	
$t^{(1)}$	$r_3^{(1)}$	$c^{(1)}$	$A^{(1)}$	$T_M^{(1)}$	$T_U^{(1)}$	$T_R^{(1)}$	$r_1^{(1)}$	$BlockHeight^{(1)}$	$TxID^{(1)}$
$t^{(2)}$	$r_3^{(2)}$	$c^{(2)}$	$A^{(2)}$	$T_M^{(2)}$	$T_U^{(2)}$	$T_R^{(2)}$	$r_1^{(2)}$	$BlockHeight^{(2)}$	$TxID^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t^{(l)}$	$r_3^{(l)}$	$c^{(l)}$	$A^{(l)}$	$T_M^{(l)}$	$T_U^{(l)}$	$T_R^{(l)}$	$r_1^{(l)}$	$BlockHeight^{(l)}$	$TxID^{(l)}$

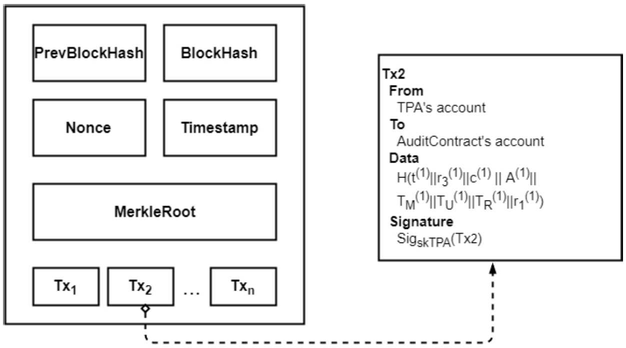


FIGURE 3. Public auditing data structure of a transaction.

E. USER CHECK

The user checks the TPA's behavior as follows.

1) *LogGen*. The TPA generates an auditing log as below:

a) for each verification task, generate an record as

$$\{t, r_3, c, A, T_M, T_U, T_R, r_1\};$$

b) store the record to a log file f in chronological order as shown in Table 1, where *TxID* denotes the transaction ID;

c) compute the hash value

$$\hbar_{t_1} = H(Bl_{t-\varphi+1}^{(1)} \| Bl_{t-\varphi+1}^{(1)} \| \cdots \| Bl_t^{(1)} \\ \| t^{(1)} \| r_3^{(1)} \| A^{(1)} \| T_M^{(1)} \| T_U^{(1)} \| T_R^{(1)} \| r_1^{(1)});$$

d) generate a transaction Tx_1 as shown in Figure 3, where the data field is set to $\hbar_t$. If the transaction is successfully recorded into the blockchain, add *BlockHeight* and *TxID* in the log file f, as shown in Table 1.

2) *CheckLog*. $\mathcal{U}$ checks the validity of the auditing results as follows:

a) acquire $t^{(1)}$, $t^{(1)} + \varphi + 1$, derive the actual time when the audit was performed from $t^{(1)}$ and $t^{(1)} + \varphi + 1$, and reject if the time does not match the agreed one;

b) extract $\hbar_{t_1}$ from the blockchain, and reject if the extraction fails;

c) check whether $\hbar_{t_1}$ matches the entry in the first row of f, and rejects if $\hbar_{t_1}$ does not match the agreed one;

d) compute $\tau = H(name \| n \| r_1 \| PK_U)$, $i_\xi^{(1)} = \pi_1(k_1, \xi)$ and $v_{i_\xi}^{(1)} = \pi_2(k_2, \xi)$, where

$$k_1 = h_1(Bl_{t-\varphi+1}^{(1)} \| Bl_{t-\varphi+2}^{(1)} \| \cdots \| Bl_t^{(1)} \| r_3^{(1)}),$$
$$k_2 = h_2(Bl_{t-\varphi+1}^{(1)} \| Bl_{t-\varphi+2}^{(1)} \| \cdots \| Bl_t^{(1)} \| r_3^{(1)});$$

e) accept if

$$A^{(1)} = e(\prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi}^{(1)} h_{i_\xi}^{(1)}}, T_M^{(1)}) \\ \cdot e(\prod_{\xi=1}^{c} W^{v_{i_\xi}^{(1)} h_{i_\xi}^{(1)}}, T_U^{(1)}) e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}^{(1)}}, T_R^{(1)}), \quad (4)$$

where $h_{i_\xi}^{(1)} = H(i_\xi^{(1)} \| \tau \| R)$, and $T_{i_\xi}^{(1)} = H_2(i_\xi^{(1)} \| \tau \| R)$, and reject otherwise.

V. SECURITY ANALYSIS

A. CORRECTNESS

Assume that the user generates file tags $\sigma = \{\{S_i\}_{i=1}^n, R, r_1\}$ honestly and the TPA and CS follow the scheme to audit the data and generate *proof* $= \{A, r_1, W, \pi\}$. Correctness can be verified as follows. Regarding Eq. (2), we have:

$$e(\prod_{i=1}^{n} S_i, g)$$
$$= e(\prod_{i=1}^{n} (D_{U,0}^{m_i} V^{x_u m_i} D_{U,1}^{H(i\|T\|R)} W^{x_u H(i\|T\|R)} T_i^{r_2}), g)$$
$$= e(\prod_{i=1}^{n} (D_{U,0}^{m_i} D_{U,1}^{H(i\|T\|R)}), g)$$
$$\cdot e(\prod_{i=1}^{n} (V^{x_u m_i} W^{x_u H(i\|T\|R)}), g) e(\prod_{i=1}^{n} T_i^{r_2}, g)$$
$$= e(\prod_{i=1}^{n} (Q_{U,0}^{m_i} Q_{U,1}^{hi}), P_M) e(\prod_{i=1}^{n} (V^{m_i} W^{hi}), PK_U) e(\prod_{i=1}^{n} T_i, R).$$

Regarding Eqs. (3) and (4), we have:

$$A = e(S, T_G)/(e(Q_{U,0}^{\mu}, T_M) \cdot e(V^{\mu}, T_U))$$
$$= e(\prod_{\xi=1}^{c} S_{i_\xi}^{v_{i_\xi}}, g^{\rho})/(e(Q_{U,0}^{\mu}, P_M^{\rho}) \cdot e(V^{\mu}, PK_U^{\rho}))$$
$$= e(Q_{U,0}^{\mu} \prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi} h_{i_\xi}}, P_M^{\rho}) e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, R^{\rho})$$

$$\times e(V^{\mu}\prod_{\xi=1}^{c} W^{v_{i_\xi}h_{i_\xi}}, PK_U^{\rho})/(e(Q_{U,0}^{\mu}, P_M^{\rho})e(V^{\mu}, PK_U^{\rho}))$$
$$= e(Q_{U,0}^{\mu}, P_M^{\rho})e(\prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi}h_{i_\xi}}, P_M^{\rho})e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, R^{\rho})$$
$$e(V^{\mu}, PK_U^{\rho})e(\prod_{\xi=1}^{c} W^{v_{i_\xi}h_{i_\xi}}, PK_U^{\rho})$$
$$/(e(Q_{U,0}^{\mu}, P_M^{\rho})e(V^{\mu}, PK_U^{\rho}))$$
$$= \times e(\prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi}h_{i_\xi}}, P_M^{\rho})e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, R^{\rho})e(\prod_{\xi=1}^{c} W^{v_{i_\xi}h_{i_\xi}}, PK_U^{\rho})$$
$$= e(\prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi}h_{i_\xi}}, T_M)e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, T_R)e(\prod_{\xi=1}^{c} W^{v_{i_\xi}h_{i_\xi}}, T_U).$$

B. SECURITY ANALYSIS

Lemma 1: If the CDH problem is hard, the user's file tags are unforgeable under adaptively chosen-message attacks.

Similar with [26], we can prove that it is computational infeasible for an adversary who does not own the user's secret key to forge a valid signature $\sigma = \{\{S_i\}_{i=1}^{n}, R, r_1\}$. So we omit the proof here.

Lemma 2: As an inside adversary, the cloud server could not forge μ to pass the verification done by the TPA.

Proof: [Proof Sketch] Assume the CS forges μ to μ' and passes the verification. We know that for a given challenge message, the correct responding should be $A = e(S, T_G)/(e(Q_{U,0}^{\mu}, T_M) \cdot e(V^{\mu}, T_U))$. Suppose that the CS outputs the response $\{A', r_1, W, \pi\}$, where $A' = e(S, T_G)/(e(Q_{U,0}^{\mu'}, T_M) \cdot e(V^{\mu'}, T_U))$, which passes the verification done by the TPA. We have that $A/A' = 1$, therefore,

$$e(Q_{U,0}^{\mu}, T_M) \cdot e(V^{\mu}, T_U) = e(Q_{U,0}^{\mu'}, T_M) \cdot e(V^{\mu'}, T_U).$$

That is,

$$e(Q_{U,0}^{\alpha\rho\mu} \cdot V^{x_u\rho\mu}, g) = e(Q_{U,0}^{\alpha\rho\mu'} \cdot V^{x_u\rho\mu'}, g).$$

We get that

$$(Q_{U,0}^{\alpha\rho} \cdot V^{x_u\rho})^{\mu} = (Q_{U,0}^{\alpha\rho} \cdot V^{x_u\rho})^{\mu'}.$$

Since $\mu \neq \mu'$, we set $\omega = Q_{U,0}^{\alpha\rho} \cdot V^{x_u\rho}$ which can be represented as $\omega = (g')^{\chi^*} \cdot (g'')^{\chi'^*}$, where $\chi^*, \chi'^* \in \mathbb{Z}_p$, $g', g'' \in \mathbb{G}_1$ are randomly chosen. Furthermore, there exists $x \in \mathbb{Z}_p, g'' = (g')^x$. Therefore, the discrete logarithm problem here is that given $g', g'' = (g')^x$, compute $x \in \mathbb{Z}_p$, so the solution of discrete log problem is $x = -(\chi^*/\chi'^*)$. However, χ'^* is zero only with probability $1/p$, which is negligible because p is a large prime. We then get a solution to the DL problem with probability of $1 - 1/p$, which contradicts the assumption that the DL problem in $\mathbb{G}_1$ is computationally infeasible.

Theorem 1: Our DBPA scheme achieves the authenticity. That is, if the cloud server's response passes the TPA's verification, it must possess the specified data truly.

Proof: [Proof Sketch] The proof follows from that in Section 4.2 of [48]. A challenger is used to obtain a valid response $\{A, r_1, W, \pi\}$. In addition, the cloud server is treated as an adversary and the challenger controls the random oracle $H(\cdot)$. If there is a non-negligible probability that adversary wins, we can construct a simulator that solves the DL problem and CDH problem. To prove the authenticity of DBPA, we define a sequence of games with interleaved analysis as follows.

Game 0: This is simply the original authenticity game played between the TPA and the CS defined in Section III-D.

Game 1: It is the same as Game 0, with the exception that the adversary tries to forge a part of the proof information in **Audit**. Since $\sigma_i = \{S_i, R\}$ in DBPA is existentially unforgeable, the challenger records each response generated by the adversary, and declares failure and aborts if

1) the response is valid, and
2) the response $\{A, r_1, W, \pi' = \{A, T_M, T_U, T_R'\}\}$ is different from the expected one $\{A, r_1, W, \pi = \{A, T_M, T_U, T_R\}\}$.

Analysis. Denote the event above by abt_1. Given a challenge message, the expected response $\{A, r_1, W, \pi\}$ should satisfy that

$$A = e(S, T_G)/(e(Q_{U,0}^{\mu}, T_M) \cdot e(V^{\mu}, T_U))$$
$$= e(\prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi}h_{i_\xi}}, T_M)e(\prod_{\xi=1}^{c} W^{v_{i_\xi}h_{i_\xi}}, T_U)e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, T_R).$$

In case that the challenger aborts, the response $\{A, r_1, W, \pi'\}$ generated by the adversary satisfies that

$$A = e(S, T_G)/(e(Q_{U,0}^{\mu}, T_M) \cdot e(V^{\mu}, T_U))$$
$$= e(\prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi}h_{i_\xi}}, T_M)e(\prod_{\xi=1}^{c} W^{v_{i_\xi}h_{i_\xi}}, T_U)e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, T_R').$$

We know that $\Delta_{T_R} = T_R - T_R' \neq 0$ since $T_R \neq T_R'$. We further have $r_2 \neq r_2'$, $\Delta_{r_2} = r_2 - r_2' \neq 0$, and

$$e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, T_R) = e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, T_R'),$$

which is

$$e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}r_2\rho}, g) = e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}r_2'\rho}, g).$$

Equally, we have $\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}r_2\rho} = \prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}r_2'\rho}$, and then $\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}\rho\Delta_{r_2}} = 1$.

Given a discrete logarithm problem $g, h \in \mathbb{G}_1$, if we set $T_{i_\xi} = g^{a_\xi} \cdot h^{b_\xi}$ for some $a_\xi, b_\xi \in \mathbb{Z}_p$ and $\xi \in [1, c]$, the solution to the DL problem could be given as $x = \log_g h = -\sum_{\xi=1}^{c} a_\xi v_{i_\xi}\rho\Delta_{r_2}/\sum_{\xi=1}^{c} b_\xi v_{i_\xi}\rho\Delta_{r_2}$. However, Δ_{r_2} is zero only with the probability $1/p$, which is negligible because p is a large prime. Then we get a solution to the DL problem with a probability of $(1-1/p)\Pr[abt_1]$, which is non-negligible if $\Pr[abt_1]$ is so, contradicting the DL assumption. Therefore, we have that the difference between the adversary's success probabilities in Game 0 and Game 1 is non-negligible.

Game 2: It is the same as Game 1, except that the adversary is trained to be able to forge any part of response information in **Audit**. That is, the challenger records each response information generated by the adversary, declares failure and aborts if the response $\{A', r_1, W, \pi' = \{A', T'_M, T'_U, T'_R\}\}$ is valid and different from the expected one $\{A, r_1, W, \pi = \{A, T_M, T_U, T_R\}\}$.

Analysis. Denote the event above by abt_2. Given a CDH problem instance $(g, g^{\alpha}, g^{\beta})$, the challenger sets $g^* = g^{\alpha}$ and $P_M = g^{\beta}$ at the beginning of the game, sets $Q_{U,0} = g^{b_0} \cdot g^{\alpha b'_0}$, $Q_{U,1} = g^{b_1} \cdot g^{\alpha b'_1}$, $h_i = -m_i^* b'_0/b'_1$ where b_0, b'_0, b_1, b'_1 are randomly chosen from $\mathbb{Z}_p$, and randomly selects $x_u \leftarrow \mathbb{Z}_p$ as (part of) the user's secret key. To generate tags for a file $M^* = \{m_i^*\}$, the challenger randomly chooses $r_2 \leftarrow \mathbb{Z}_p$ and computes $(\{S_i\}, R)$, where $R = g^{r_2}$ and

$$S_i = g^{b_0 \beta m_i^*} \cdot V^{x_u m_i^*} \cdot g^{-b_1 \beta m_i^* b'_0/b'_1} \cdot W^{-x_u m_i^* b'_0/b'_1} \cdot T_i^{r_2}.$$

According to Game 1, we know that $T_R = T'_R$. Besides, we have that

$$\begin{aligned} A &= e(S^{\rho}, g)/(e(Q_{U,0}^{\mu}, T_M) \cdot e(V^{\mu}, T_U)) \\ &= e(\prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi} h_{i_\xi}}, T_M) e(\prod_{\xi=1}^{c} W^{v_{i_\xi} h_{i_\xi}}, T_U) e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, T_R) \end{aligned}$$

and

$$\begin{aligned} A' &= e(S^{\rho}, g)/(e(Q_{U,0}^{\mu}, T'_M) \cdot e(V^{\mu}, T'_U)) \\ &= e(\prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi} h_{i_\xi}}, T'_M) e(\prod_{\xi=1}^{c} W^{v_{i_\xi} h_{i_\xi}}, T'_U) e(\prod_{\xi=1}^{c} T_{i_\xi}^{v_{i_\xi}}, T'_R). \end{aligned}$$

We can get that

$$\begin{aligned} & e(Q_{U,0}^{\mu} \cdot \prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi} h_{i_\xi}}, T_M) e(V_{\mu} \cdot \prod_{\xi=1}^{c} W^{v_{i_\xi} h_{i_\xi}}, T_U) \\ &\quad = e(Q_{U,0}^{\mu} \cdot \prod_{\xi=1}^{c} Q_{U,1}^{v_{i_\xi} h_{i_\xi}}, T'_M) e(V_{\mu} \cdot \prod_{\xi=1}^{c} W^{v_{i_\xi} h_{i_\xi}}, T'_U). \end{aligned}$$

Equally, we have

$$\begin{aligned} & e((g^{b_0 \mu \beta} \cdot g^{\alpha b'_0 \mu \beta} \cdot \prod_{\xi=1}^{c} (g^{-b_1 v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 \beta} \cdot g^{-\alpha v_{i_\xi} m_{i_\xi}^* b'_0 \beta}) \\ &\quad \cdot V^{\mu x_u} \cdot \prod_{\xi=1}^{c} W^{-v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 x_u})^{\rho}, g) \\ &= e((g^{b_0 \mu \beta} \cdot g^{\alpha b'_0 \mu \beta} \cdot \prod_{\xi=1}^{c} (g^{-b_1 v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 \beta} \cdot g^{-\alpha v_{i_\xi} m_{i_\xi}^* b'_0 \beta}) \\ &\quad \cdot V^{\mu x_u} \cdot \prod_{\xi=1}^{c} W^{-v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 x_u})^{\rho'}, g). \end{aligned}$$

Obviously, we can obtain

$$\begin{aligned} & (g^{b_0 \mu \beta} \cdot g^{\alpha b'_0 \mu \beta} \cdot \prod_{\xi=1}^{c} (g^{-b_1 v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 \beta} \cdot g^{-\alpha v_{i_\xi} m_{i_\xi}^* b'_0 \beta}) \\ &\quad \cdot V^{\mu x_u} \cdot \prod_{\xi=1}^{c} W^{-v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 x_u})^{\rho} \\ &= (g^{b_0 \mu \beta} \cdot g^{\alpha b'_0 \mu \beta} \cdot \prod_{\xi=1}^{c} (g^{-b_1 v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 \beta} \cdot g^{-\alpha v_{i_\xi} m_{i_\xi}^* b'_0 \beta}) \\ &\quad \cdot V^{\mu x_u} \cdot \prod_{\xi=1}^{c} W^{-v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 x_u})^{\rho'}. \end{aligned}$$

Since $\rho \neq \rho'$, we can get

$$\begin{aligned} \varpi &= g^{b_0 \mu \beta} g^{\alpha b'_0 \mu \beta} \prod_{\xi=1}^{c} (g^{-b_1 v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 \beta} g^{-\alpha v_{i_\xi} m_{i_\xi}^* b'_0 \beta}) \\ &\quad \cdot V^{\mu x_u} \cdot \prod_{\xi=1}^{c} W^{-v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 x_u} \\ &= 1. \end{aligned}$$

Here, the solution to the given CDH problem is

$$\begin{aligned} g^{\alpha \beta} &= (g^{b_0 \mu \beta} \cdot \prod_{\xi=1}^{c} (g^{-b_1 v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 \beta} \cdot V^{\mu x_u} \\ &\qquad \cdot \prod_{\xi=1}^{c} W^{-v_{i_\xi} m_{i_\xi}^* b'_0/b'_1 x_u}))^{-(b'_0 \mu - \sum_{\xi=1}^{c} (v_{i_\xi} m_{i_\xi}^* b'_0))^{-1}}. \end{aligned}$$

Note that the probability of game failure is the same as that of

$$b'_0 \cdot (\mu - \sum_{\xi=1}^{c} (v_{i_\xi} m_{i_\xi}^*)) = 0 \bmod p,$$

which is $1/p$. Since p is a large prime, it is thus negligible. Therefore, the probability that we solve the CDH problem is $(1 - 1/p) \cdot \Pr[abt_2]$, which is non-negligible if $\Pr[abt_2]$ is so, contradicting the CDH assumption. Hence, the difference between the adversary's success probabilities in Game 1 and Game 2 is negligible.

Theorem 2: The cloud sever's response $proof = \{A, r_1, W, \pi\}$ does not leak any information about μ to the TPA.

Proof: In the response $proof = \{A, r_1, W, \pi\}$ returned by the cloud server, only S and μ may leak information about the user's data. However, S is hidden in W by a random exponent $\theta \in \mathbb{Z}_p$, and both S and μ are hidden in

$$\begin{aligned} A &= e(S^{\rho}, g)/(e(Q_{U,0}^{\mu}, T_M) \cdot e(V^{\mu}, T_U)) \\ &= [e(S, g)/(e(Q_{U,0}^{\mu}, P_M) \cdot e(V^{\mu}, PK_U))]^{\rho} \end{aligned}$$

by a random exponent $\rho \in \mathbb{Z}_p$, where $T_M = P_M^{\rho}$ and $T_U = PK_U^{\rho}$. Furthermore, the zero-knowledge proof π does not leak any information about the witness ρ, μ, θ. To simulate the response, the simulator could randomly select A', r'_1, W' from the corresponding domains, invoke the simulation algorithm of the zero-knowledge proof to produce a simulated proof π', and output $proof' = \{A', r_1', W', \pi'\}$. It is not hard to see that the simulated $proof'$ is indistinguishable from a real response $proof$. Therefore, the response $proof$ does not leak any information about the user's data.

Lemma 3: For a public blockchain whose consensus algorithm is based on proof of work, hash value of the next block that will be generated at a future time is unpredictable.

Proof: Assume that the adversary is a miner, and the cloud server may collude with the miner. We follow the existing threat model of miner in [49]. Suppose that data $M = \{m_i\}_{1 \leq i \leq n}$ consists of n blocks, where κ blocks are not valid, e.g. *corrupted*. The auditor challenges c blocks to check the integrity of file M. Denote the probability of detecting invalid blocks successfully by P_X, where X is the number of invalid blocks being challenged. We have

$$\begin{aligned} P_X &= \Pr[X \geq 1] = 1 - \Pr[X = 0] \\ &= 1 - \frac{n-\kappa}{n} \times \frac{n-\kappa-1}{n-1} \times \cdots \times \frac{n-\kappa-c+1}{n-c+1}. \end{aligned} \tag{5}$$

Since $\frac{n-\kappa-i}{n-i} > \frac{n-\kappa-i-1}{n-i-1}$, we have

$$1 - \left(\frac{n-\kappa}{n}\right)^c \leq P_X \leq 1 - \left(\frac{n-\kappa-c+1}{n-c+1}\right)^c. \tag{6}$$

Denote by $P_{corrupt} = \kappa/n$ the probability of data corruption. We have that

$$P_X = 1 - (1 - P_{corrupt})^c. \tag{7}$$

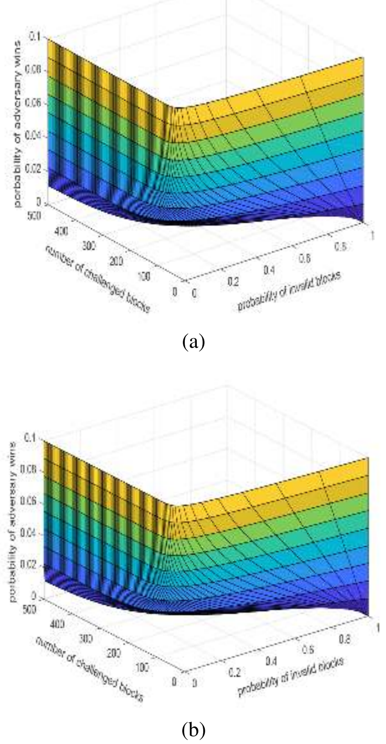
Denote by $P = (1 - P_{corrupt})^c$ the probability that invalid blocks are not detected by the auditor, and by $P_{\mathcal{A}}$ the probability that the adversary $\mathcal{A}$ wins, i.e. successfully cheating the auditor. According to [49], we know that

$$P_{\mathcal{A}} = \frac{P}{1 - \Upsilon(1-P)} = \frac{(1-P_{corrupt})^c}{1 - \Upsilon[1-(1-P_{corrupt})^c]}, \tag{8}$$

where Υ denotes the proportion of $\mathcal{A}$'s mining hashrate. The parameter Υ measures the relative power of $\mathcal{A}$ and can be interpreted as the probability that the next oracle request gives a valid block. For security in blockchain, we assume $\Upsilon < 51\%$. If $\mathcal{A}$ is able to control more than half of computation power of the whole blockchain network, security of the blockchain would be broken. For example, when $\Upsilon = 25\%$, $P_{corrupt} = 10\%$, $c = 500$, the probability that $\mathcal{A}$ wins is 0.013055, indicating that although $\mathcal{A}$ has strong computation power, the probability that it wins is still small.

Theorem 3: The challenge information is unpredictable for the cloud server.

Proof: In our scheme, the challenge message is generated as $k_1 = h_1(Bl_{t-\varphi+1}\| Bl_{t-\varphi+2}\| \cdots \|Bl_t\|r_3)$, $k_2 = h_2(Bl_{t-\varphi+1}\|Bl_{t-\varphi+2}\| \cdots \|Bl_t\|r_3)$, $i_\xi = \pi_1(k_1, \xi)$, $v_{i_\xi} = \pi_2(k_2, \xi)$. As we can see, the challenge message is determined by two parts. One part is r_1 which is generated by the auditor, and the other part $Bl_{t-\varphi+1}\|Bl_{t-\varphi+2}\| \cdots \|Bl_t$ is determined by the public blockchain, which is publicly transparent and tamper-proof. A new block is generated approximately per 15 seconds in Ethereum. The cloud server could not control the generation of a new block, and by Lemma 3 the hash value of the new block is unpredictable for the cloud server. Hence, the challenge information is unpredictable for the cloud server.



(a)

(b)

FIGURE 4. Probability that the adversary wins when (a) Υ is 0.25 and (b) Υ is 0.5.

To ensure the integrity of outsourced data, most existing schemes assume that the cloud server would not collude with the TPA, which is a strong assumption. If the two entities collude, the TPA may send fake audit results to the user, in order to help the CS to cover up a data corruption event and conceal its mistake, without being detected by the user. In our DBPA scheme, the challenge information is generated based on the latest blockchain information and the choices of the TPA. Each audit information generated between the TPA and CS is packed into a transaction and recorded into the blockchain. Due to the characteristics of blockchain, the whole audit process, including challenge information generation, response proof generation, and audit results verification, is thus traceable. Any misbehavior of the TPA could be traced. As long as the blockchain remains tamper-resistant, we can learn from the audit information recorded on the blockchain that the TPA honestly fulfilled its obligation to audit the user's data stored on the CS. Hence, we have the following theorem.

Theorem 4: Misbehavior of the TPA in auditing the user's data is traceable.

TABLE 2. Property Comparison.

Properties	SWP [48]	YU [28]	SCLPV [24]	CPVPA [26]	IBPA [27]	Ours
Public auditing	✓	✓	✓	✓	✓	✓
Blockchain-based	×	✓	✓	✓	✓	✓
Identity-based	×	×	✓	✓	✓	✓
Privacy Preservation	×	×	×	×	×	✓
Resist challenge message guessing	×	×	×	×	×	✓
Consensus mechanism	×	PBFT	PoW	PoW	PoW	PoW

TABLE 3. Comparison in Computation Overhead.

Computation Cost	Cloud Server	TPA
SWP [48]	$cE_{\mathbb{G}} + (c-1)M_{\mathbb{G}} + cM_{\mathbb{Z}_p}$	$2P + 2cE_{\mathbb{G}} + (2c-1)M_{\mathbb{G}}$
SCLPV [24]	$2cE_{\mathbb{G}} + 2cM_{\mathbb{G}} + cM_{\mathbb{Z}_p}$	$4P + (2c+2)E_{\mathbb{G}} + 2cM_{\mathbb{G}} + 2cM_{\mathbb{Z}_p}$
CPVPA [26]	$cE_{\mathbb{G}} + cM_{\mathbb{G}} + cM_{\mathbb{Z}_p} + 2cC_f$	$4P + (3c+2)E_{\mathbb{G}} + (3c-1)M_{\mathbb{G}}$ $+2cM_{\mathbb{Z}_p} + 2cC_f$
Ours	$4P + (c+16)E_{\mathbb{G}} + cM_{\mathbb{G}}$ $+(c+1)M_{\mathbb{Z}_p} + 2cC_f$	$6P + (3c+7)E_{\mathbb{G}} + (3c-3)M_{\mathbb{G}}$ $+2cM_{\mathbb{Z}_p} + 2cC_f$

VI. PERFORMANCE

In this section, we provide comparisons of our scheme with some other related schemes, in terms of functional features, computational and communicational overhead.

A. PROPERTY COMPARISON

Table 3 shows the comparison of our DBPA scheme with some other schemes in the literature in terms of functional features. As we can see, the proposed scheme supports all the features compared to the existing schemes. Concretely, the proposed scheme supports public auditing, which means the auditing proofs could be verified by any user. Moreover, all the auditing proofs in our scheme are traceable since all the hash values of auditing proofs are stored in the blockchain permanently and cannot be tampered with. In addition, our scheme achieves privacy preservation during the process of auditing, while the other schemes are not. Furthermore, our scheme could prevent the adversary from challenge message guessing, while the other blockchain-based schemes could not.

B. EFFICIENCY COMPARISON

In this part we compare our DBPA scheme with schemes SWP [48], SCLPV [24], and CPVPA [26] in terms of computation overhead and communication overhead. Table 3 provides a comparison in computational efficiency of the cloud server and the TPA, where $M_{\mathbb{G}}$, $M_{\mathbb{Z}_p}$, $E_{\mathbb{G}}$ denote a scalar multiplication in $\mathbb{G}$, a scalar multiplication in $\mathbb{Z}_p$, a modular exponentiation in $\mathbb{G}$, respectively, P denotes a bilinear pairing, C_f denotes the evaluation of a PRF, and c denotes the total number of challenge data blocks. From the table we learn that both the computational overhead of the cloud server and that of the TPA are slightly higher than those of the other three schemes. However, our scheme provides a good solution to privacy protection of the user's data, which is more important for the users if their data is sensitive. Table 4 provides a comparison of our scheme with [24] and [26] in terms of communication overhead. In our scheme, the TPA needs to send the challenge message φ, t, r_3, c to the cloud server in the first move. After receiving the challenge message, the cloud server needs to return a response information $proof = \{A, r_1, W, \pi\}$ to the TPA. As we can see, the communication overhead of our scheme on the TPA side is $|\mathbb{Z}_p|$, and the communication overhead on the cloud server side is $11|\mathbb{G}_1| + |\mathbb{G}_T| + 5|\mathbb{Z}_p|$, where $|\mathbb{G}_1|$, $|\mathbb{G}_T|$ and $|\mathbb{Z}_p|$ denote the length of an element of $\mathbb{G}_1$, $\mathbb{G}_T$ and $\mathbb{Z}_p$, respectively, c denotes the number of challenge blocks, and $|H|$ denotes the hash length a block in the underlying blockchain. From Table 4 we learn that the communication overhead in SCLPV is linear with c on the TPA side, while those in our scheme and CPVPA are independent of c.

TABLE 4. Comparison in Communication Overhead.

Communication overhead	TPA	Cloud Server
SWP [48]	$c\|\mathbb{Z}_p\|$	$\|\mathbb{G}_1\| + \|\mathbb{Z}_p\|$
SCLPV [24]	$c\|\mathbb{Z}_p\|$	$2\|\mathbb{G}_1\| + 2\|\mathbb{Z}_p\|$
CPVPA [26]	$\varphi\|H\|$	$2\|\mathbb{G}_1\| + 2\|\mathbb{Z}_p\|$
Ours	$\|\mathbb{Z}_p\|$	$11\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + 5\|\mathbb{Z}_p\|$

Overall, our DBPA scheme provides better privacy protection and security guarantee, but at the cost of a little higher communication overhead, when compared with SCLPV and CPVPA. Specifically, the extra communication overhead on the cloud server side is $9|\mathbb{G}_1|+|\mathbb{G}_T|+3|\mathbb{Z}_p|$ for the protection of user privacy against the TPA, and that on the TPA side is $|\mathbb{Z}_p|$ for randomizing the challenge message in order to resist against the cloud server.

C. EXPERIMENTAL RESULTS

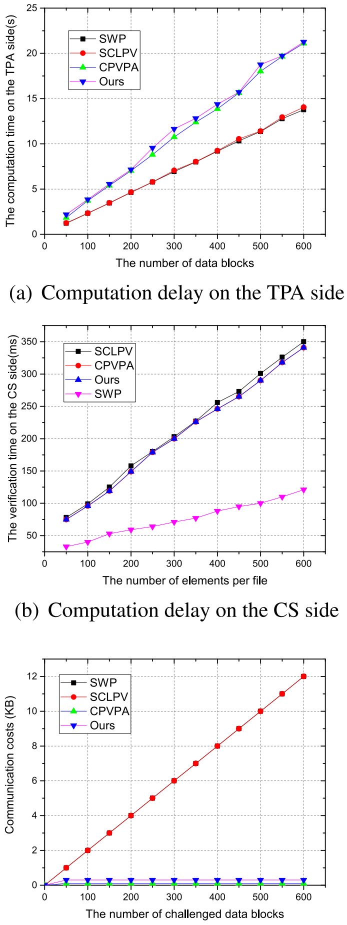
To demonstrate the usability of our DBPA scheme, we implement the scheme in Java. The experiments are conducted on Windows 10 operating system, with Intel(R) Core(TM) i7 CPU, 2.5GHZ and 8GB RAM. We make use of the JPBC library, and utilize Type-A curve in our experiment. To support 80-bit security level, we set the parameter p to be of 160 bits. Figure 5(a) shows the computation delay on the TPA side with different challenge block numbers. As we can see, as the value of number c increases, the auditing delay linearly increases as well, as more exponentiations and multiplications in $\mathbb{G}_1$ are needed. Compared with CPVPA [26], our scheme requires almost the same time to conduct the verification on the same number of challenge data blocks. Figure 5(b) shows the verification time of the CS side, which almost grows linearly with the number of elements per file. The verification time in our DBPA scheme is almost the same as that in SCLPV [24] and CPVPA [26]. Furthermore, We show the communication overhead between the TPA and the CS in Figure 5(c), which is independent of the number of challenge blocks in our scheme and CPVPA [26], while it is linear with the number of challenge data blocks in SWP [48] and SCLPV [24]. Furthermore, we use Ethereum blockchain to examine the efficiency and cost of our DBPA scheme. We use Solidity to create a contract and publish it to Kovan public test network.[2] The current price configuration is 0.0012 Ether per million gas, and the current rate is about 1Ether ≈ 261.8$. Our wallet address is

0x851Ca2C940f1AD6eb10094dC08a37df81B3BE114.

The contract is deployed at block 1846343, and costs 209978 gas. The transaction hash is

0x9c901a6f1b58f381a77da1492f54282
2e61b7435f236b188666c9b373e4c7eb7.

The transaction is confirmed at 22:49 on May 25th, 2020. We first tested how the transaction confirmation time varies in different number of data blocks, and set the number of data blocks from 0 to 100. The results are shown in Figure 6(a). As we can see, the time cost for confirmation has a positive relationship with the transaction numbers. When a transaction has been confirmed by at least 12 nodes on the blockchain network, we consider the transaction is tamper-resistant [50]. Furthermore, we also tested the transaction fee with different transaction numbers in Figure 6(b). We set the transaction number from 0 to 100, and found that the gas cost is linear to the transaction numbers. This is reasonable, because in our scheme the transaction data only includes the hash value, which is of constant size, and thus every transaction costs almost the same fee. From our experiment, the average cost of each transaction is about 0.000035 Ether. More specifically, assume that the number of transactions is 50. Our scheme requires about 0.00175 Ether, which is equal to 0.45815$, which is acceptable.

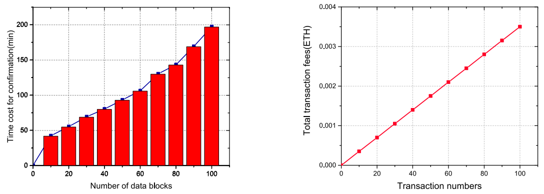


(a) Computation delay on the TPA side

(b) Computation delay on the CS side

(c) **Communication costs between TPA and CS**

FIGURE 5. Efficiency comparison.



(a) Time cost on Blockchain

(b) Total transaction fees

FIGURE 6. Performance in Ethereum blockchain.

VII. CONCLUSION

In this paper, we proposed a decentralized and privacy-preserving public auditing scheme, which is secure against the procrastinating third-party auditor and malicious cloud server. Our scheme utilizes two components to generate unpredicted challenge messages. One is generated by the auditor, and the other is a series of decentralized block hashes. Our scheme could resist against the procrastinating auditor, and a malicious cloud server could not retrieve or guess the challenge message ahead of the audit time. Furthermore, our scheme provides better protection of user privacy during the process of verification of the audit response from the cloud server. We analyzed our scheme to show that it is secure, and

[2] Kovan Testnet: `https://kovan.etherscan.io/`

conducted a comprehensive performance analysis, showing that our scheme has low communication overhead and is efficient in terms of computation overhead. We did experiments on Kovan testnet of Ethereum blockchain to demonstrate the practicability of our scheme.

REFERENCES

[1] E. Azhir, N. J. Navimipour, M. Hosseinzadeh, A. Sharifi, and A. Darwesh, "Query optimization mechanisms in the cloud environments: A systematic study," *Int. J. Commun. Syst.*, vol. 32, no. 8, May 2019, Art. no. e3940.

[2] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," *J. Netw. Comput. Appl.*, vol. 79, pp. 88–115, Feb. 2017.

[3] Y. Shin, D. Koo, and J. Hur, "A survey of secure data deduplication schemes for cloud storage systems," *ACM Comput. Surveys*, vol. 49, no. 4, pp. 1–38, Feb. 2017.

[4] M. Du, Q. Wang, M. He, and J. Weng, "Privacy-preserving indexing and query processing for secure dynamic cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2320–2332, Sep. 2018.

[5] N. Kaaniche and M. Laurent, "Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms," *Comput. Commun.*, vol. 111, pp. 120–141, Oct. 2017.

[6] Y. Li, K. Gai, L. Qiu, M. Qiu, and H. Zhao, "Intelligent cryptography approach for secure distributed big data storage in cloud computing," *Inf. Sci.*, vol. 387, pp. 103–115, May 2017.

[7] N. A. Kofahi and A. R. Al-Rabadi, "Identifying the top threats in cloud computing and its suggested solutions: A survey," *Adv. Netw.*, vol. 6, no. 1, pp. 1–13, 2018.

[8] A. Juels and B. S. Kaliski, "Pors: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur. CCS*, 2007, pp. 584–597.

[9] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur. CCS*, 2007, pp. 598–609.

[10] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," *IEEE Trans. Services Comput.*, vol. 8, no. 1, pp. 92–106, Jan. 2015.

[11] J. Yuan and S. Yu, "Public integrity auditing for dynamic data sharing with multiuser modification," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1717–1726, Aug. 2015.

[12] T. Jiang, X. Chen, and J. Ma, "Public integrity auditing for shared dynamic cloud data with group user revocation," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2363–2373, Aug. 2016.

[13] D. He, B. Huang, and J. Chen, "New certificateless short signature scheme," *IET Inf. Secur.*, vol. 7, no. 2, pp. 113–117, Jun. 2013.

[14] H. Wang, D. He, and S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1165–1176, Jun. 2016.

[15] S. Peng, F. Zhou, Q. Wang, Z. Xu, and J. Xu, "Identity-based public multi-replica provable data possession," *IEEE Access*, vol. 5, pp. 26990–27001, 2017.

[16] W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 331–346, Feb. 2019.

[17] X. Zhang, H. Wang, and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," *Inf. Sci.*, vol. 472, pp. 223–234, Jan. 2019.

[18] L. Zhou, A. Fu, S. Yu, M. Su, and B. Kuang, "Data integrity verification of the outsourced big data in the cloud environment: A survey," *J. Netw. Comput. Appl.*, vol. 122, pp. 1–15, Nov. 2018.

[19] H. Tian, F. Nan, C.-C. Chang, Y. Huang, J. Lu, and Y. Du, "Privacy-preserving public auditing for secure data storage in fog-to-cloud computing," *J. Netw. Comput. Appl.*, vol. 127, pp. 59–69, Feb. 2019.

[20] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 43–56, Jan. 2014.

[21] J. Zhao, C. Xu, F. Li, and W. Zhang, "Identity-based public verification with privacy-preserving for data storage security in cloud computing," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E96.A, no. 12, pp. 2709–2716, 2013.

[22] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, "Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium," *J. Netw. Comput. Appl.*, vol. 82, pp. 56–64, Mar. 2017.

[23] F. Armknecht, J.-M. Bohli, G. O. Karame, Z. Liu, and C. A. Reuter, "Outsourced proofs of retrievability," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. CCS*, 2014, pp. 831–843.

[24] Y. Zhang, C. Xu, S. Yu, H. Li, and X. Zhang, "SCLPV: Secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors," *IEEE Trans. Comput. Social Syst.*, vol. 2, no. 4, pp. 159–170, Dec. 2015.

[25] Y. Zhang, C. Xu, H. Li, and X. Liang, "Cryptographic public verification of data integrity for cloud storage systems," *IEEE Cloud Comput.*, vol. 3, no. 5, pp. 44–52, Sep. 2016.

[26] Y. Zhang, C. Xu, X. Lin, and X. S. Shen, "Blockchain-based public integrity verification for cloud storage against procrastinating auditors," *IEEE Trans. Cloud Comput.*, early access, Mar. 29, 2019, doi: 10.1109/TCC.2019.2908400.

[27] J. Xue, C. Xu, J. Zhao, and J. Ma, "Identity-based public auditing for cloud storage systems against malicious auditors via blockchain," *Sci. China Inf. Sci.*, vol. 62, no. 3, Mar. 2019.

[28] H. Yu, Z. Yang, and R. O. Sinnott, "Decentralized big data auditing for smart city environments leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 6288–6296, 2019.

[29] Y. Wu, X. Lin, X. Lu, J. Su, and P. Chen, "A secure light-weight public auditing scheme in cloud computing with potentially malicious third party auditor," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 10, pp. 2638–2642, 2016.

[30] K. Qian and H. Huang, "A new identity-based public auditing against malicious auditor in the cloud," *Int. J. Embedded Syst.*, vol. 11, no. 4, pp. 452–460, 2019.

[31] X. Zhang, J. Zhao, C. Xu, H. Li, H. Wang, and Y. Zhang, "CIPPPA: Conditional identity privacy-preserving public auditing for cloud-based WBANs against malicious auditors," *IEEE Trans. Cloud Comput.*, early access, Jul. 10, 2019, doi: 10.1109/TCC.2019.2927219.

[32] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[33] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.

[34] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[35] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, vol. 107, pp. 841–853, Jun. 2020.

[36] Y. Yuan and F.-Y. Wang, "Blockchain and cryptocurrencies: Model, techniques, and applications," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 9, pp. 1421–1428, Sep. 2018.

[37] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[38] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for AI: Review and open research challenges," *IEEE Access*, vol. 7, pp. 10127–10149, 2019.

[39] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "MedBlock: Efficient and secure medical data sharing via blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 136, Aug. 2018.

[40] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sep. 2018.

[41] K. Leng, Y. Bi, L. Jing, H.-C. Fu, and I. Van Nieuwenhuyse, "Research on agricultural supply chain system with double chain architecture based on blockchain technology," *Future Gener. Comput. Syst.*, vol. 86, pp. 641–649, Sep. 2018.

[42] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Syst. J.*, vol. 12, no. 1, pp. 64–73, Mar. 2018.

[43] M. Ali, S. U. R. Malik, and S. U. Khan, "DaSCE: Data security for cloud environment with semi-trusted third party," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 642–655, Oct. 2017.

[44] H. Xu, J. Cao, J. Zhang, L. Gong, and Z. Gu, "A survey: Cloud data security based on blockchain technology," in *Proc. IEEE 4th Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2019, pp. 618–624.

[45] N. Ravi and N. R. Sunitha, "Introduction of blockchain to mitigate the trusted third party auditing for cloud security: An overview," in *Proc. 2nd Int. Conf. Emerg. Comput. Inf. Technol. (ICECIT)*, Dec. 2017, pp. 1–6.

[46] Y. Zhang, C. Xu, N. Cheng, H. Li, H. Yang, and X. Shen, "Chronos^{+} +: An accurate blockchain-based time-stamping scheme for cloud storage," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 216–229, Mar./Apr. 2020.

[47] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2003, pp. 452–473.

[48] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2008, pp. 90–107.

[49] C. Pierrot and B. Wesolowski, "Malleability of the blockchain's entropy," *Cryptography Commun.*, vol. 10, no. 1, pp. 211–233, Jan. 2018.

[50] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A taxonomy of blockchain-based systems for architecture design," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Apr. 2017, pp. 243–252.

YING MIAO received the B.S. degree from South China Agricultural University, in 2018, where she is currently pursuing the M.S. degree with the College of Mathematics and Informatics. Her research interests include data security and blockchain.

QIONG HUANG received the Ph.D. degree from the City University of Hong Kong, in 2010. He is currently a Professor with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China. He has published more than 110 research papers in international conferences and journals. His research interests include cryptography and information security, in particular, cryptographic protocols design and analysis. He has served as a Programme Committee Member in many international conferences.

MEIYAN XIAO received the B.S. and M.S. degrees from South China Agricultural University, where she is currently pursuing the Ph.D. degree with the College of Mathematics and Informatics. Her research interests include data security and blockchain.

HONGBO LI received the B.S. and M.S. degrees from South China Agricultural University, Guangzhou, China, where he is currently pursuing the Ph.D. degree with the College of Mathematics and Informatics. His research interests include applied cryptography and cloud security.