Digital Communications and Networks 8 (2022) 591–603

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

A survey on blockchain-based integrity auditing for cloud data

Haoxiang Han [a], Shufan Fei [a], Zheng Yan [a,*], Xiaokang Zhou [b,c]

[a] The State Key Lab of ISN, School of Cyber Engineering, Xidian University, 266 Xinglong Section of Xifeng Road, Xi'an, Shaanxi, 710126, China
[b] The Faculty of Data Science, Shiga University, Hikone, 522-8522, Japan
[c] The RIKEN Center for Advanced Intelligence Project, RIKEN, Tokyo, 103-0027, Japan

ARTICLE INFO

Keywords:
Blockchain
Cloud storage
Integrity auditing
Decentralization

ABSTRACT

With the rapid advancement of cloud computing, cloud storage services have developed rapidly. One issue that has attracted particular attention in such remote storage services is that cloud storage servers are not enough to reliably save and maintain data, which greatly affects users' confidence in purchasing and consuming cloud storage services. Traditional data integrity auditing techniques for cloud data storage are centralized, which faces huge security risks due to single-point-of-failure and vulnerabilities of central auditing servers. Blockchain technology offers a new approach to this problem. Many researchers have endeavored to employ the blockchain for data integrity auditing. Based on the search of relevant papers, we found that existing literature lacks a thorough survey of blockchain-based integrity auditing for cloud data. In this paper, we make an in-depth survey on cloud data integrity auditing based on blockchain. Firstly, we cover essential basic knowledge of integrity auditing for cloud data and blockchain techniques. Then, we propose a series of requirements for evaluating existing Blockchain-based Data Integrity Auditing (BDIA) schemes. Furthermore, we provide a comprehensive review of existing BDIA schemes and evaluate them based on our proposed criteria. Finally, according to our completed review and analysis, we explore some open issues and suggest research directions worthy of further efforts in the future.

1. Introduction

The explosive growth of data has not only spawned a large number of new applications based on Information Communication Technology (ICT), but also plagued users with limited resources at the same time. Fortunately, cloud storage and cloud computing can be used to liberate individuals and enterprises with limited resources. The cloud service can provide users with enough space to store their data [1,2], i.e., users can use cloud servers to store and manage their data efficiently and conveniently without paying high data storage costs. However, outsourcing local data to the cloud means that users lose direct control of their data. In reality, cloud storage service is not completely reliable, considering the security risks brought by natural hazards, external attacks, disk damage, internal attacks and even human error accidents. In addition, compromised or malicious Cloud Service Providers (CSPs) can also bring huge security and privacy infringement risks to users. For example, a profit-driven CSP may delete outsourcing data that is rarely accessed without obtaining users' permission [3,4]. A malicious CSP may even tamper with some data to obtain economic benefits. Therefore, it is urgent and essential to design a scheme to verify whether the data stored in the cloud is intact.

To ensure the integrity of remote data, many researchers have proposed a series of Remote Data Integrity Auditing (RDIA) schemes, in which a Data Owner (DO) can check the integrity of remote data. In these RDIA schemes, a mechanism named challenge-proof-verify is designed to achieve private auditing between CSP and DO, where a DO generates a challenge to a CSP and verifies the corresponding proof from the CSP to check data integrity. However, such solutions have some limitations.

- Firstly, the verification procedure is only performed by the DO in private auditing, which means that DOs will bear a large computational burdens due to the increase of data volume and auditing requests.
- Secondly, CSPs are usually assumed to be fully trustworthy for DOs. Unfortunately, the DO's full trust in CSP is unrealistic. For example, a CSP may conceal data corruption incidents to maintain its reputation.
- Thirdly, from the perspective of CSP, the DO is also not fully trustworthy because DO may maliciously forge the result of data integrity auditing to claim unreasonable indemnity.

* Corresponding author.
E-mail addresses: haoxianghan0228@gmail.com (H. Han), shufanfei@gmail.com (S. Fei), zyan@xidian.edu.cn (Z. Yan), zhou@biwako.shiga-u.ac.jp (X. Zhou).

https://doi.org/10.1016/j.dcan.2022.04.036
Received 1 August 2021; Received in revised form 28 April 2022; Accepted 29 April 2022
Available online 5 May 2022

To solve the above limitations, many researchers have introduced a Third-Party-Auditor (TPA) to achieve public auditing [16–18]. It is assumed that TPA has sufficient computing power to accept delegations from resource-constraint users and execute outsourced computation according to user needs. In addition, the TPA is regarded as an honest and trustworthy entity, which is unreasonable in practical application scenarios. In fact, a TPA may collude with a CSP or DO. Besides, such a centralized structure has two disadvantages: (1) Single point of failure: auditing systems cannot work normally once the TPA is hacked or broken down. (2) Performance limitation: a TPA cannot handle large-scale auditing requests from multiple users simultaneously.

In order to solve the problems of the single-point-of-failure and trust issues, some researchers suggested employing blockchain in cloud data auditing schemes. Blockchain technology was first proposed by Nakamoto [19]. The specific characteristics of blockchain, such as decentralization, tamper-proof, immutability, traceability have brought great inspiration to RDIA schemes [20–23]. Many researchers endeavored to utilize the blockchain to strengthen the security and trustworthiness of RDIA [24].

There have been some surveys on RDIA techniques, but they mainly focus on centralized TPA-based data integrity auditing. According to our review of existing RDIA-related literature, a blockchain-based approach is currently the only way to achieve distributed RDIA. However, a comprehensive survey of Blockchain-based Data Integrity Auditing(BDIA) is still lacking. Thus, we are motivated to give a thorough survey on blockchain-based integrity auditing techniques for cloud data. Table 1 shows the comparison of our survey with other existing ones. Zafar et al. [7] and Sookhak et al. [8] focused on the centralized data integrity auditing, but ignored the decentralized data integrity auditing, which was addressed in our survey. Li et al. [5] and Zhou et al. [6] conducted in-depth survey on TPA-based data integrity auditing, and they pointed out that blockchain can be used for decentralized data integrity auditing, but they did not give a review on BDIA. Our work gives a thorough review of BDIA schemes. Gangadevi et al. [15] investigated BDIA schemes, but they did not compare the BDIA schemes with centralized TPA-based data integrity auditing schemes to show the advantages of the BDIA schemes, which are handled in our survey.

In this paper, we first introduce essential basic knowledge of cloud data integrity auditing and blockchain techniques. Then, we propose a series of requirements to evaluate existing BDIA schemes. Furthermore, we review existing works and evaluate them with the criteria we propose. Finally, we propose some open issues and suggest future research directions based on our review and analysis. In particular, the main contributions of our survey can be summarized below:

(i) We propose a taxonomy of existing BDIA schemes.
(ii) We propose a set of criteria for evaluating the BDIA schemes.
(iii) We give a review of existing BDIA schemes and utilize the proposed criteria to evaluate them and compare their advantages and disadvantages.
(iv) Based on our review and evaluation, we explore a number of open issues and further propose several future research directions.

The remainder of the paper is organized as follows. In the next section, we give a brief introduction to some basic concepts and background knowledge, which can help readers understand this paper. In Section 3, we propose a set of metrics for evaluating existing BDIA schemes. In Section 4, we give a comprehensive review of existing BDIA schemes and utilize the proposed metrics to evaluate them. In Section 5, we explore unsolved open issues and propose promising research directions in the future. Finally, we conclude this paper in the last section.

2. Background knowledge

In this section, we give a brief introduction to some basic concepts and background knowledge, which can help readers understand our paper. We first introduce blockchain technology, including its origin, development, application scenarios, characteristics, taxonomy and consensus mechanism. Then, we introduce homomorphic signature techniques, which are often applied for data integrity auditing. What is more, we overview TPA-based data integrity auditing and BDIA techniques. Finally, we introduce the security threats of RDIA, which paves the way for the analysis of the security of BDIA schemes.

2.1. Blockchain

Blockchain was first proposed by Satoshi Nakamoto [19], a cryptocurrency named after Bitcoin, which combines decentralized consensus and only additional data structure. Bitcoin could be regarded as a distributed public ledger, recording and verifying transactions in an open Peer-to-Peer (P2P) network. Bitcoin is resistant to double-spending attacks in a completely decentralized P2P network without any trusted centralized authority [25].

Following Bitcoin, an emerging technique of blockchain is smart contract [26], which is a program deployed on the blockchain and run by miner nodes. Based on predefined rules, the smart contract can automatically perform some operations (e.g., verify the proof information from the CSP), which is one of the advantages of blockchain for RDIA. It is reasonable to assume that with the continuous evolution of blockchain, the BDIA technology could be continually improved.

Till now, blockchain technology has been widely used in various scenarios, including industry, government and finance [27–30]. In data auditing applications, most BDIA schemes are based on public blockchain and consortium blockchain. In the past few years, blockchain has evolved from digital currency (Blockchain 1.0) to smart contract (Blockchain 2.0), and to various forms of decentralized collaborations with high security and trust (Blockchain 3.0) [31–33].

In a blockchain network, it is ensured that once a transaction is packaged into a block successfully (i.e., the block containing the transaction has been successfully generated), the transactions in the generated block can no longer be modified [34,35]. In general, blockchain is regarded as a one-way list of data blocks. As shown in Fig. 1, in addition to the transaction information, each block includes a hash pointer that returns to its previous block. The blocks linked could be regarded as a one-way hash chain. The salient features of blockchain can be listed and explained as follows:

Table 1
Comparison of our survey with existing surveys.

Topic	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	Our Survey
Give a review of Homomorphic Signature Techniques	×	✓	✓	✓	✓	×	✓	✓	✓	✓	×	✓
Summarize possible attacks on data integrity auditing schemes	×	✓	✓	×	×	✓	✓	×	✓	×	×	✓
Give a review of the performance and security of centralized TPA-based data integrity auditing schemes	×	×	×	×	×	×	×	×	×	×	×	✓
Give a review of BDIA	✓	✓	×	×	×	×	×	×	×	×	✓	✓
Compare BDIA and TPA-based data integrity auditing	×	×	×	×	×	×	×	×	×	×	×	✓
Focus on auditing schemes based on blockchain	×	×	×	×	×	×	×	×	×	×	✓	✓

✓: discussed; × : not discussed.

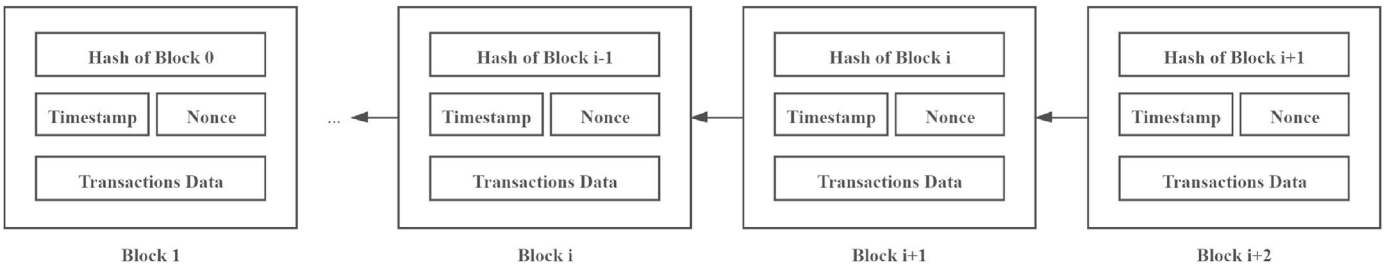

Fig. 1. The structure of blockchain.

(1) Decentralization: Blockchain does not rely on any third-party management institution or hardware facilities. Each node realizes the self verification, transmission and management of information through distributed accounting and storage.
(2) Openness: The openness of blockchain means that anyone who complies with a set of rules can be given permission to join the system and process transactions.
(3) Tamper-resistance: Each block of the blockchain stores the hash value of its previous block. Users can publicly validate the hash value to verify whether the stored data has been modified.
(4) Anonymity: Anonymity is the main characteristic of public blockchain [36]. The real-world identity of users cannot be directly obtained from any transaction information. For example, when a user issues a public transaction, its transaction address will be recorded in the blockchain without leaking its identity information.

Blockchain can be divided into three different types according to the different licensing mechanisms for miners' participation.

(1) Public blockchain is accessible to anyone with connections to the blockchain network [37]. The participants can send, receive and verify transactions by participating in the consensus procedure. Usually, economic incentives is offered for those involved in consensus mechanism and the maintenance of the ledger.
(2) Consortium blockchain is a semi-private system with a permissioned user group that can work across different organizations [38]. Its write permissions are constrained so that only a pre-selected entity can participate in the maintenance of the ledger. Meanwhile, read permission is open to anyone in the network.
(3) Private blockchain is an invitation-only blockchain managed by an administrator [39]. Only those who get permission from the network administrator can read, write or operate the blockchain. This blockchain has multiple layers of data access to ensure data confidentiality.

Consensus mechanism in blockchain is a fault-tolerant mechanism for achieving an agreement on the same network state. As a publicly shared ledger, blockchain needs an efficient, fair, real-time and secure mechanism to authenticate each transaction that occurs on blockchain, and all nodes agree to the same status of the ledger. Such an important task can be performed through the consensus mechanism, which is a set of rules used to verify the validity of the contributions of all nodes in the blockchain system. Popular consensus mechanisms that have been applied in BDIA schemes are Proof-of-Work (PoW), Proof-of-Stack (PoS), Practical Byzantine Fault Tolerance (PBFT). For example, Yu et al. [40] utilized a consensus algorithm based on PBFT to implement BDIA. In PoW, miners compete with each other to compute a hash value using different nonce values. The miner who successfully computes the relevant hash value obtains priority to create a new block and gets the payment of a specific cryptocurrency. In PoS, a new block is generated by validators randomly selected by the Committee rather than miners, and the probability of being selected as a validator is proportional to the amount of assets it holds. Compared with PoW, PoS consumes less energy. PBFT aims to solve the general Byzantine problem [41], and it can tolerate 2/3 malicious Byzantine nodes; that is, the consensus can be achieved correctly even if only a small percentage of nodes work honestly.

2.2. Homomorphic signature techniques

The Homomorphic Verification Tag (HVT) plays an important role in BDIA because it can help verify data integrity without retrieving the data. For a file block m, we define T_m as its HVT. The HVT will be stored on the CSP together with the file F and used as unforgeable verification metadata for the file block. In addition to the unforgeability, HVT also meets another two properties as follows:

(1) Block-less Verifiability: By using HVT, a CSP can generate a proof, with which users can verify whether the CSP possesses intact file blocks, even when the user does not have permission to access the actual file blocks.
(2) Homomorphism: Given two values T_{m_i} and T_{m_j}, anyone can combine them into a value $T_{m_i+m_j}$ corresponding to the sum of the messages $m_i + m_j$.

In order to improve auditing efficiency, existing BDIA schemes combine signature technology with HVTs to develop homomorphic signature techniques. The most important feature of homomorphic signature techniques is the property of block-less verifiability. With block-less verifiability, the verifier can check data integrity by retrieving a single block without having to download the whole data. According to different types of signature schemes, the signature-based auditing techniques can be divided into four types as follows.

2.2.1. RSA-based homomorphic methods

Zhang et al. [42] proposed a provable data possession scheme that combines the Rivest-Shamir-Adleman (RSA) signature with HVTs, which can be simplified as follows.

In the key generation phase, the client chooses two prime numbers p and q and computes $N = pq$. Then, it selects a large secret prime e and computes $d \equiv e^{-1} (\mathrm{mod}\ \varphi(N))$. Let $H : \{0,1\}^* \rightarrow Z_p^*$ be a full-domain hash function, and h: $\{0,1\}^* \rightarrow \{0,1\}^{256}$ is a hash function. The public key is (e, N), where n is the number of file blocks. The private key is (d, p, q).

In the tag generation phase, users divide the file F into n blocks $F = (m_1, m_2, \ldots, m_n)$. Then, users compute a tag $\sigma_i = (h(Fname) \cdot \mu^{m_i})^d \bmod N$ for each block m_i, where μ is randomly chosen from Z_p^*, $Fname$ denotes the file name. The CSP stores file and corresponding tags.

In the challenge phase, a user firstly selects the file block to be verified. Then, the user generates a random challenge and sends it to the CSP.

In the verification phase, after verifying the proof, the user can judge whether the data is stored correctly without retrieving the whole data block.

2.2.2. BLS-based homomorphic methods

Compared with the RSA-based homomorphic method, the

Boneh–Lynn–Shacham (BLS)-based homomorphic method has less storage overhead as it offers shorter homomorphic signatures. Besides, the request and response are shortest (20 and 40 bytes) in BLS signature. Wang et al. [43] proposed a public auditing scheme based on BLS.

In the key generation phase, a bilinear pairing is a map e: $G \times G \rightarrow G_T$, and h: $\{0,1\}^* \rightarrow G$ is a cryptographic hash function that maps strings uniformly to G. The public key pk is $y = g^x \in G$, and the private key sk is $x \in Z_p$.

In the tag generation phase, a user divides file F into each block m_i and chooses a random element $\beta \in G$. Then, the user computes the tag $\sigma_i = (h(m_i) \cdot \beta^{m_i})^x$ of all blocks and denotes the set of signatures by $\varphi = \{\sigma_i\}$, $1 \le i \le n$. Later, the user generates a root R based on the Merkle Hash Tree, where the leaf nodes of the tree are ordered set hashes of file tags: $H(m_i)$, $(i = 1, 2, \ldots, n)$. Next, a user signs the root R with the private key x: $sig_{sk}(H(R)) \leftarrow (H(R))^x$. A user sends $\{F, \varphi, sig_{sk}(H(R))\}$ to the server and deletes it from the local.

In the challenge phase, a verifier (the user or TPA) selects a random c-element subset $I = \{s_1, s_2, \ldots, s_c\}$ of set $[1, n]$, where $s_1 \le \cdots \le s_c$. For each $i \in I$, the verifier chooses a random element $v_i \in Z_p$. Then, the verifier generates a challenge message $chal = \{(i, v_i)\}_{s_1 \le i \le s_c}$, which specifies the location of the blocks to be checked in this challenging phase. Next, the verifier sends $chal$ to the server.

In the verification phase, after receiving $chal$, the server computes $\mu = \sum_{i=s_1}^{s_c} v_i m_i \in Z_p$ and $\sigma = \prod_{i=s_1}^{s_c} \sigma_i^{v_i} \in G$, where both the data blocks and the corresponding signature tags are aggregated into a single block, respectively. Then, the server returns proof information $P = \{\mu, \sigma, sig_{sk}(H(R))\}$ with some auxiliary information $\{\Omega_i\}_{s_1 \le i \le s_c}$ (the node siblings on the path from the leaf to the root R of MHT) to the verifier. After receiving the proof information, the verifier generates MHT root R by using $\{H(m_i), \Omega_i\}_{s_1 \le i \le s_c}$ and checks whether $e(sig_{sk}(H(R)), g) = e(H(R), g^x)$. If they are not equal, the verifier outputs *FALSE*. Otherwise, the verifier checks whether $e(\sigma, g) = e(\prod_{i=s_1}^{s_c} H(m_i)^{v_i} \cdot \mu^{\mu}, v)$. If they are equal, the verifier outputs *TRUE* and the data is intact. Otherwise, *FALSE* is output.

2.2.3. Identity-based(ID-based) homomorphic methods

ID-based signature schemes do not depend on any certificate compared with RSA signature and BLS signature. Wang et al. [44] proposed ID-based data possession schemes.

In the key generation phase, let e be a bilinear map: e: $G \times G \rightarrow G_T$, let $f : Z_q^* \times \{0,1\}^{\log_2 |n|} \rightarrow Z_q^*$ be a pseudo-random function, and let $\pi : Z_q^* \times \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$ be a pseudo-random permutation. The private key generator selects $x \in Z_p^*$ and computes $Y = g^x$ as a public key. After receiving ID from users, the private key generator picks $r \in Z_p^*$, computes $R = g^r$ and $\sigma = r + xh(ID, R)$ sends the private key (R, σ) to a user through a secure channel.

In the tag generation phase, for each file block m_i, a user generates tags $T_i = (h(i) \cdot \mu^{m_i})^{\sigma}$ and sends (m_i, T_i) to the server.

In the challenge phase, a verifier chooses random keys $k_1, k_2 \in Z_q^*$ of the pseudo-random permutation π and a pseudo-random function f. Then, the verifier generates a challenge message $chal = (c, k_1, k_2)$ to check the integrity of c $(1 \le c \le n)$ file block. Next, the verifier sends $chal$ to the cloud server.

In the verification phase, after receiving $chal$, the cloud server generates the indices and coefficients: $i_j = \pi_{k_1}(j), a_j = f_{k_2}(j)$. The challenge message $chal$ is defined as an ordered set $\{c, i_1, \ldots, i_c, a_1, \ldots, a_c\}$. Then, the cloud server computes $T = \prod_{j=1}^{c} T_{i_j}^{a_j}$ and $\hat{m} = \sum_{j=1}^{c} a_j m_j$. Next, the cloud server returns proof information $V = (T, \hat{m})$ to the verifier. Finally, the verifier checks whether $e(T, g) = e(\prod_{i=1}^{c} h(\pi_{k_1}(i))^{f_{k_2}(i)} \mu \hat{m}, R \cdot Y^{H(ID,R)})$. If they are equal, the verifier outputs *TRUE* and the data is intact. Otherwise, *FALSE* is output.

2.2.4. Short signature homomorphic methods

The short signature is proposed by Zhang et al. [45], which is based on a bilinear pairing and has less overhead than BLS signature. We need to use the bilinearity property of bilinear mapping: $e(P^x, P^y) = e(P,P)^{xy}$.

In the key generation phase, a user selects a random integer $\alpha \in Z_p$ as its private key sk and αP as its public key pk. Others cannot calculate α from pk.

In the tag generation phase, for each file block m_i, a user generates a tag $\sigma_i = \frac{1}{h(m_i)+\alpha} P$, where h: $\{0,1\}^* \rightarrow G$ is a cryptographic hash function.

In the challenge phase, a verifier randomly chooses $c \in [1, n]$ to construct a data block index set $I = \{s_1, s_2, \ldots, s_c\}$ and generates a pseudo-random number for each $i \in I$. Then, the verifier sends a challenge message set $chal = \{(i, v_i)\}$ to the cloud server.

In the verification phase, after receiving $chal$, the cloud server computes $R = \sum_{i=s_1}^{s_c} v_i Y, \mu = \sum_{i=s_1}^{s_c} v_i H(m_i) P, \eta = P - P^2 \sum_{i=s_1}^{s_c} \frac{v_i}{\sigma_i}$ and returns proof information $\{R, \mu, \eta\}$ to the verifier. The verifier checks whether $e(n, P) \cdot e(\mu + R, P) = e(P, P)$. If they are equal, the verifier outputs *TRUE* and the data is intact. Otherwise, *FALSE* is output.

2.3. Overview of TPA-based data integrity auditing

The architecture of a TPA-based data integrity auditing scheme is shown in Fig. 2. The following are three entities in the architecture of TPA-based data integrity auditing schemes [46].

(1) Users: Users can be classified into two types: DOs and Authorized Users (AUs). DOs have a large amount of data to store and can perform dynamic data operations in a CSP. An AU is the entity authorized to access data stored by DOs.
(2) TPA: A TPA can accept auditing delegations from DOs and perform auditing tasks on behalf of DOs.
(3) CSP: A CSP provides DOs with adequate storage space and computing resources.

The working procedure of the TPA-based data integrity auditing scheme can be summarized as follows.

(1) DOs firstly encrypt their data and then generate the metadata of the encrypted data. After that, DOs upload the encrypted data as well as the corresponding metadata to CSPs. If a DO wants to check the integrity of outsourced data, it can delegate an auditing task to a TPA and wait for the auditing results. (2) After receiving the auditing tasks from the DO, the TPA generates a random challenge and sends it to the CSP. (3) After receiving the challenge from the TPA, the CSP generates the proof corresponding to the challenge and sends it to the TPA. (4) The TPA verifies the proof and returns auditing results to DOs.

Although TPA-based data integrity schemes can reduce the verification burden of DOs, it brings a series of problems. Firstly, A TPA is a centralized entity, so once the TPA is compromised or broken down, the entire auditing system will collapse [47]. Secondly, in these schemes, it is assumed that TPAs are credible. However, TPAs may collude with CSPs to conceal auditing results in reality. Therefore, the credibility of TPA-based schemes is hard to be guaranteed. It is difficult for this scheme to provide users with a trustworthy auditing arbitration.

2.4. Overview of BDIA

The architecture of BDIA schemes is shown in Fig. 3. There are dedicated nodes in the blockchain network corresponding to DOs and CSPs [40]. A DO who needs to check data integrity firstly generates a challenge and sends it to a CSP. Then, the CSP generates an auditing proof according to the received challenge and broadcasts the proof to the blockchain network. After that, the representative nodes package the

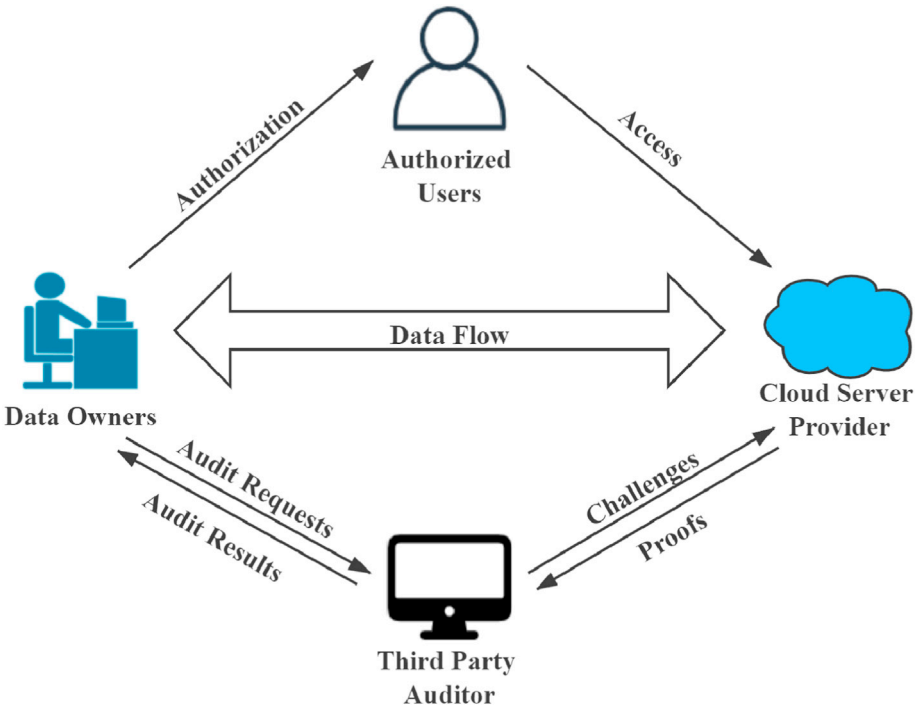

Fig. 2. The architecture of TPA-based data integrity auditing.

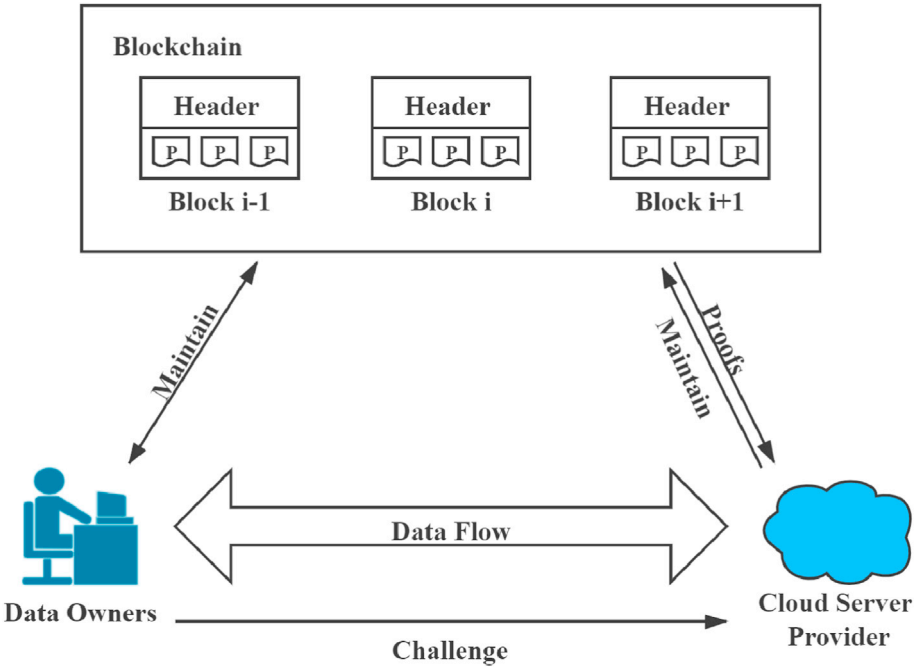

Fig. 3. The architecture of a BDIA scheme.

auditing proof into a new block and communicate with other nodes to publish the new block. Finally, the DO can obtain the auditing proof from blockchain and verify the auditing proof.

Compared with the TPA-based data integrity auditing schemes, the advantages of using blockchain in integrity auditing can be summarized as follows: (1) Decentralization: Blockchain is a distributed digital ledger maintained by miner nodes rather than a central authority. DOs are able to delegate RDIA tasks to any miners without looking for a trustworthy TPA. After receiving auditing requests from DOs, a group of miners instead of a TPA perform auditing tasks, which greatly mitigates the single point of failure threat of the single TPA. (2) Transparency: The records on blockchain are publicly visible, which means that all miners and authorized parties can access and verify the validity of auditing results. (3) Tamper-resistance: Each block of the blockchain stores the hash value of its previous block. DOs can publicly validate the hash value to verify whether the outsourced data has been tampered with, which can prevent CSP and TPA from conspiring to tamper with auditing results. (4) Non-repudiation: Blockchain can record operation logs of the data or auditing history. DOs are able to analyze the logs to uncover potential malicious CSPs or auditors, and it is hard for these malicious entities to

deny their malicious operations. However, there are also some limitations that should not be ignored in the BDIA schemes. For example, the latency of auditing is corresponding as each transaction needs to be verified by all minor nodes with consensus, which actually reduces auditing efficiency.

2.5. Security threats of remote data integrity auditing

Data integrity auditing schemes are vulnerable to some possible attacks [48,49], which can be divided into internal attacks and external attacks. The former is initiated by a CSP or an auditor, and the latter by an outside attacker. The following are the potential attacks against a data integrity auditing scheme:

(1) Replacement attack: The CSP may attempt to pass auditing verification by replacing the corrupted pair of file blocks and corresponding tags with another uncorrupted pair of file blocks and corresponding tags.
(2) Forge attack: The CSP may attempt to hide the user's data damage and deceive verifiers by forging proofs.
(3) Replay attack: The CSP tries to pass current verification by replaying previous proofs which have passed verification.
(4) Collusion attack: The CSP may collude with auditors to change challenges or proofs to mislead users.
(5) Recovery attack: An outside attacker attempts to reveal outsourced data by acting as a public auditor.

3. Evaluation criteria

In this section, we list a series of requirements that can serve as the criteria for evaluating existing BDIA schemes.

(1) Traceability (TR)

Traceability means that logs of operations on the data can be quickly retrieved by any legitimate entity and cannot be tampered with. By this way, DOs can track the change of outsourced data by analyzing the operation logs. If a BDIA scheme satisfies traceability, DOs can find some malicious behaviors by analyzing the operation logs of the data (i.e., any malicious data operation can be detected by DOs, which is non-repudiation). Therefore, it is necessary to seriously consider traceability.

(2) Provable Security (PS)

Provable security is an important property of some cryptographic constructions, especially the asymmetric cryptography-based schemes. The paradigm of provable security is as follows. The first step is usually to determine the security goal. The second step is to establish a formal adversarial model according to the ability of the adversary and define what it does mean for a secure cryptographic scheme. The third is to analyze security goal based on the formal adversarial model. Finally, the security analysis can be reduced to a mathematical problem such as Integer Factorization Problem (IFP) and Discrete Logarithm Problem (DLP). It can be reasonably assumed that it is impossible to solve those mathematical problems (e.g., IFP and DLP) considering the current development level of computing equipment.

The result of provable security is to avoid heavy burdens of searching for specific attacks against a cryptographic scheme. Because if the underlying cryptographic primitives are regarded secure and the computational problems are hard to be solved, the security risk is small. Therefore, provable security is of great significance for BDIA schemes [5].

(3) Dynamic Data Operation (DD)

Dynamic data operation refers to the property that a BDIA scheme can support dynamic operations of the data, including modification, insertion and deletion. DOs can perform dynamic operations of the data stored on the cloud without having to download the whole file and re-uploading it after modification. If a BDIA scheme supports dynamic data operation, DOs only need to download file blocks that need to be updated, instead of the whole file blocks. Therefore, the dynamic data operation is an important requirement of BDIA schemes [5,6,9].

(4) Batch Auditing (BA)

Batch auditing ensures that a group of auditing nodes in the blockchain can handle multiple auditing tasks of different users simultaneously instead of auditing separately. A BDIA scheme that supports batch auditing can greatly improve auditing efficiency and reduce its computation overhead. Therefore, batch auditing is an important indicator of the efficiency of BDIA schemes [5–7,9].

(5) Privacy Preservation (PP)

Privacy preservation means that the privacy of DOs should not be exposed during the entire auditing process. Once a malicious entity obtains the origin data, the private information (e.g., identity privacy and data privacy) may be disclosed. Therefore, a data integrity auditing scheme should take privacy preservation into account [5,7,9].

(6) Auditing Delegation (AD)

Auditing delegation enables a group of auditing nodes with high computing power to check data integrity without compromising the privacy of DO during the auditing process. The auditing nodes can be entities in the blockchain network. With auditing delegation, DOs could delegate the auditing tasks to the auditing nodes, which can greatly improve auditing capacity of BDIA schemes and reduce the burdens of DOs. Therefore, it is necessary to consider auditing delegation in the design of BDIA schemes [7,9].

(7) Incentive Mechanism (IM)

Incentive mechanism is to promote the entities in the auditing system to complete their own work honestly and efficiently to obtain remuneration. Moreover, a subtle incentive mechanism can attract more nodes to participate in the consensus and improve network reliability and consensus quality. Therefore, it is necessary to employ incentive mechanisms in BDIA schemes.

(8) Reputation Evaluation (RE)

Reputation evaluation is to quantify reputation based on the experiences of earlier interactions of entities. Reputation management system can be employed to assess the risk of outsourcing data. High reputation means high security level and low security risks. Therefore, an efficient and reliable BDIA scheme should support reputation evaluation.

(9) Data Recovery (DR)

Data recovery enables DOs to recover corrupted outsourced data. A qualified BDIA scheme can not only process integrity verification but also provide relevant measures to recover corrupted data if any corruption is identified. Therefore, data recovery is one of the important properties of a BDIA scheme [5,6].

(10) Computational Cost (Cp)

Computational Cost is one of the most important indexes to evaluate the performance of BDIA schemes. Different BDIA schemes often require different computational capabilities of various involved entities.

Therefore, from a practical point of view, it is necessary to evaluate the computational cost of a BDIA scheme [5]. In order to provide a unified method to evaluate the computational cost of different BDIA schemes, we select some most time-consuming cryptographic operations as the benchmark. The cryptographic operations include group multiplication, group exponentiation, bilinear pairing and the hash function that maps a string to a point in the group. Usually, these operations are executed in three phases of BDIA schemes as follows: (1) **TagGen**: A DO first pre-processes outsourced files to generate corresponding tags for each file block, which will incur a computational cost on the user side; (2) **ProofGen**: A CSP needs to generate corresponding proof information after receiving an auditing challenge from the DO, which incurs a computational cost on the cloud side; (3) **Verification**: Auditors verify the integrity of data by checking the proof information, which will incur a computational cost on the auditor side.

(11) Communication Cost (Cm)

Communication Cost refers to the size of data transferred during the process of data integrity auditing process, and it is related to the amount of file blocks and random challenged blocks. In a BDIA scheme, the transmission of the challenge and the proof information inevitably causes expensive communication costs. These communication costs are one of the important benchmarks to evaluate the performance of a BDIA scheme [5].

4. Literature review

In this section, we first provide a taxonomy of existing BDIA schemes. Then, we give a review of existing works and evaluate them by utilizing the proposed metrics, where n represents the number of outsourced file blocks and z represents the number of challenged file blocks. Table 2 presents a summary of our review results.

4.1. Taxonomy of BDIA schemes

We classify the BDIA schemes into tag-based and signature-based schemes according to different metadata generation methods, as shown in Fig. 4.

The tag-based auditing scheme generally generates tags for each file block using hash function. The tag can be used to create an aggregated value in the verification process. We further divide tag-based schemes into tag-based schemes based on a static model and tag-based schemes based on a dynamic model. In the static model, a DO has to download the entire file and then re-upload when any modification occurs, which will bring heavy communication overhead for the DO. On the other hand, in

Table 2
Summary and comparison of BDIA schemes.

		Ref	TR	PS	DD	BA	PP	AD	IM	RE	DR	Computational Cost			Cm	Thwarted attacks
												TagGen	**ProofGen**	**Verification**		
Tag-based Schemes	Static Model	[50]	×	✓	×	×	✓	✓	✓	×	×	$n(M+2E+H)$	$P+(z-1)M+(z+1)E$	$2P+(z+1)M+(z+3)E+zH$	$O(1)$	\
		[51]	×	×	×	×	✓	✓	×	×	×	–	–	–	–	\
		[52]	×	×	×	×	×	✓	×	×	×	–	–	–	–	\
	Dynamic Model	[53]	×	×	✓	✓	✓	✓	×	×	×	$n(M+2E+H)$	$(n-1)M+nE$	$2P+nM+(n+1)E+nH$	$O(n)$	Recovery attack
		[54]	×	×	✓	✓	✓	✓	×	×	×	–	–	–	–	\
Signature-based Schemes	RSA-based Homomorphic Methods	[40]	✓	✓	✓	✓	✓	✓	×	×	×	$n(M+2E+H)$	$(z-1)M+(z+1)E$	$3P+(z+1)M+(z+3)E+zH$	$O(1)$	Recovery attack Forge attack
	BLS-based Homomorphic Methods	[55]	×	✓	✓	✓	✓	✓	✓	×	×	$n(M+2E+H)$	$(2z+2)E+zH$	$2P+(z-1)M+(z+1)E$	$O(z)$	Replacement attack Forge attack Replay attack
		[56]	✓	✓	✓	✓	✓	✓	✓	✓	×	$n(M+2E+H)$	$(2z-1)M+zE$	$3P+zM+(z+1)E+zH$	$O(z)$	Replacement attack Forge attack Replay attack Collusion attack
		[57]	×	×	×	×	✓	✓	✓	×	×	$n(M+2E+H)$	zM	$2P+(2z-1)M+(2z+1)E+zH$	$O(z)$	Forge attack
		[58]	×	✓	✓	×	×	✓	✓	×	✓	$n(nM+(n+1)E+H)$	$z((2z-2)M+zE)$	$2P+z(zM+(z+1)E+H)$	$O(zlogn)$	Forge attack
	ID-based Homomorphic Methods	[59]	×	✓	×	×	×	✓	×	×	×	4 nM + $(5n+1)E+(n+2)H$	$(z-1)M+zE$	$4P+(3z+1)M+(3z+2)E$	$O(1)$	Forge attack
		[60]	×	✓	✓	✓	✓	✓	✓	×	×	$n(5M+H)$	$z(2M+H)$	$2P+M+zH$	$O(1)$	Collusion attack Forge attack
	Short Signature Homomorphic Methods	[61]	×	✓	✓	×	×	✓	×	×	×	$n(M+H)$	$(4z+1)M+zH$	$3P$	$O(n)$	Forge attack

✓: It is considered in this work; × : It is not considered in this work; -: It is not applicable in this work; \: It is not analyzed in this work; TR: Traceability; PS: Provable Security.
DD: Dynamic Data Operation; BA: Batch Auditing; PP: Privacy Preservation; AD: Auditing Delegation; IM: Incentive Mechanism; RE: Reputation Evaluation.
DR: Data Recovery; Cm: Communication Cost; n: The number of outsourced file blocks; z: The number of challenged file blocks; M: The multiplication operation on a group.
E: The exponentiation operation on a group; P: The bilinear pairing operation; H: The hash function mapping a string to a point in the group.

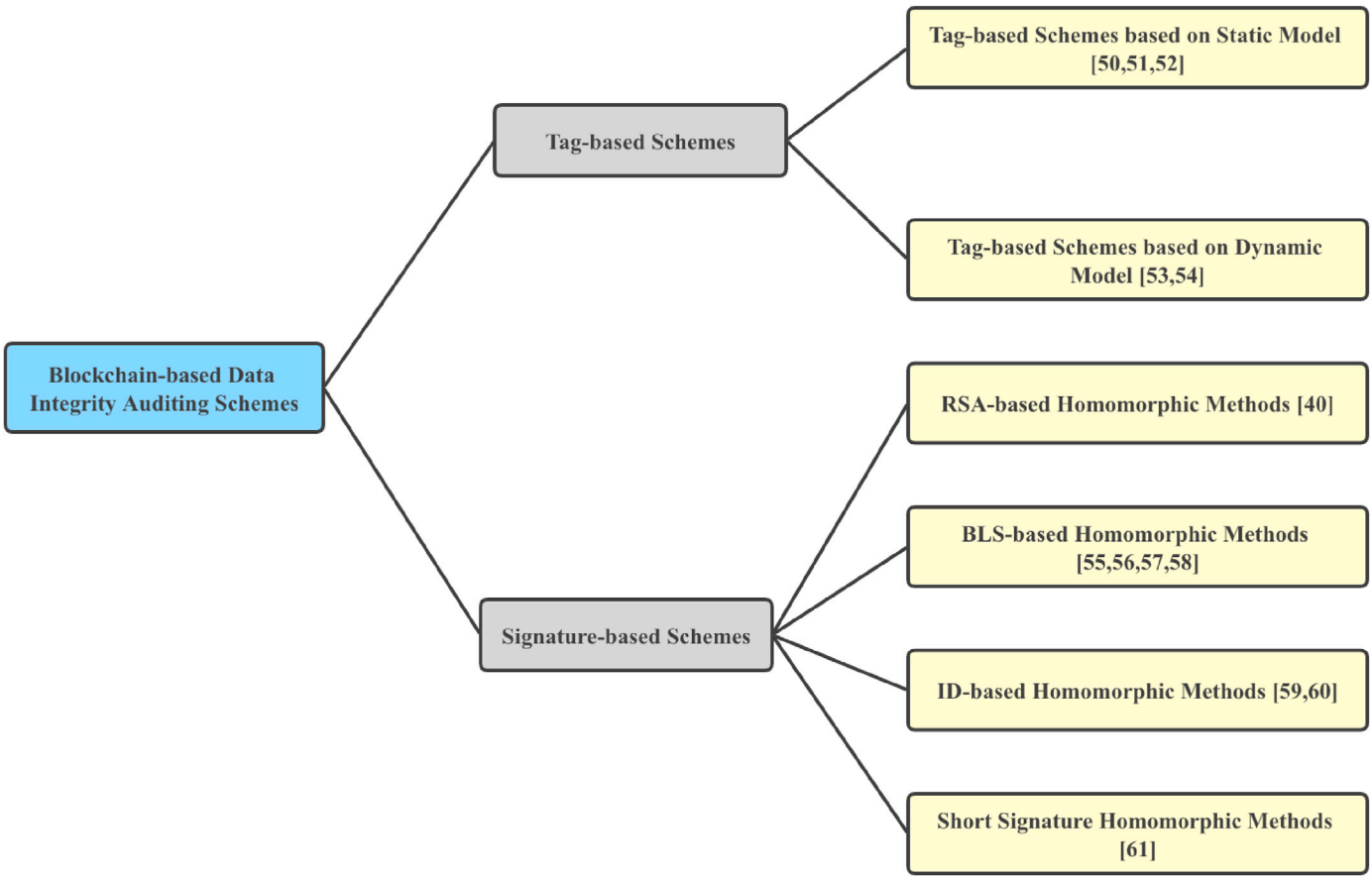

Fig. 4. The taxonomy of BDIA schemes.

the dynamic model, a DO can perform dynamic operation on data without having to download the whole file. This advantage is based on a verifiable structure (e.g., the Merkle Hash Tree (MHT) structure) that supports dynamic data updates.

Signature-based schemes combine signature technology with HVTs to construct homomorphic signature technique. According to different types of signature technology, existing signature-based schemes can be divided into four types: RSA-based homomorphic methods, BLS-based homomorphic methods, ID-based homomorphic methods, short signature homomorphic methods.

4.2. Tag-based schemes

4.2.1. Tag-based schemes based on static model

Wang et al. [50] proposed the notion of Non-Interactive Public Provable Data Possession (NI-PPDP). Based on NI-PPDP, they further designed a BDIA scheme. There are two phases in this scheme: setup phase and auditing phase. In the setup phase, a DO first generates tags for all file blocks. Then, the DO uploads file blocks and the corresponding tags to a CSP. In the auditing phase, the CSP generates proofs according to the data tags. Then, a TPA can verify whether the CSP stores complete data or not. In this work, a formalized security proof of the proposed scheme is given under a random oracle model. The security problem of this scheme can be reduced to a Computation Diffie-Hellman (CDH) problem. Moreover, the authors also give the proof that auditors cannot obtain any privacy information of DOs in the verification process. In addition, a DO can delegate auditing tasks to other entities in the blockchain (i.e., this scheme supports auditing delegation). The CSP will receive remuneration only if the verification is passed; otherwise, it will be punished to pay the penalties. In terms of computational cost, the DO performs n multiplication operations, $2n$ exponentiation operations, n mapping-to-point hash operations in the **TagGen** phase. The CSP performs one bilinear pairing operation, $z-1$ multiplication operations, $z+1$ exponentiation operations in the **ProofGen** phase. The auditor performs two bilinear pairing operations, $z+1$ multiplication operations, $z+3$ exponentiation operations, z mapping-to-point hash operations in the **Verification** phase. In addition, the transmission of the challenge information and the proof information brings $O(1)$ communication cost. Despite those benefits above, this work still suffers from some limitations. Firstly, traceability is not satisfied since this work does not store the operation logs of the data on blockchain. Secondly, this scheme is based on static model (i.e., dynamic data operation is not supported in this work). Thirdly, this work does not support batch auditing as auditing nodes can perform only one auditing operation at a time. Fourthly, this work does not employ reputation evaluation mechanisms. Fifthly, data recovery is not supported in this work because it cannot provide relevant measures to recover corrupted data once any corruption is identified.

Yue et al. [51] proposed a BDIA framework for decentralized Edge-Cloud Storage (ECS), which eliminates the semi-honest TPA. In this work, the MHT with random challenge number for data integrity verification is employed. In order to ensure timeliness, the researchers also proposed sampling verification and formulated rational sampling strategies to make the sampling verification more effective. In this work, DOs can delegate auditing tasks to the miner nodes acting as auditors (i.e., this work supports auditing delegation). Meanwhile, the random number corresponding to each shard (file block) can be used to get the leaf node of MHT. And the random number is exclusively stored by the DO, thus, only the DO can generate the corrected leaf node of MHT, which enhances privacy preservation. Despite those advantages above, this work still has some limitations. Firstly, traceability cannot be satisfied as relevant operation logs of the data are not stored on the blockchain. Secondly, a formalized security proof of the scheme is not given in this

work. Thirdly, the dynamic operations of the data are not supported in this scheme. Fourthly, this work does not support batch auditing as only one auditing operation can be executed at a time. Fifthly, this work does not employ incentive mechanisms and reputation evaluation mechanisms. Sixthly, this work cannot provide relevant measures to recover corrupted data once any corruption happens. In addition, the computational cost and the communication cost cannot be evaluated as this work does not provide specific details of each phase.

Liu et al. [52] proposed a BDIA framework for Internet of Things (IoT). This framework can provide a reliable data integrity auditing service for DOs without relying on any semi-trusted TPA. Based on this framework, the authors designed relevant protocols and implemented a prototype system to evaluate the feasibility of the protocols. In this work, a DO can delegate other DOs to perform auditing tasks (i.e., this work supports auditing delegation). Although this work does not rely on any single TPA, it has some limitations that should not be ignored. Firstly, this work does not satisfy traceability as relevant operation logs of data are not stored on the blockchain. Secondly, this work does not give the formalized security proof of the proposed scheme. Thirdly, this framework is based on static model (i.e., dynamic data operation is not supported in this work). Fourthly, this work does not support batch auditing since one auditing operation can be executed at a time. Fifthly, this work does not employ incentive mechanisms and reputation evaluation mechanisms. Sixthly, the privacy protection is not considered in this paper. Seventhly, this work does not provide relevant measures to recover corrupted data in case of any corruption. In addition, the computational cost and the communication cost cannot be measured because this work does not provide specific details of each phase.

4.2.2. Tag-based schemes based on dynamic model

Li et al. [53] proposed a BDIA scheme for cloud storage, which stores the hashtags of encrypted file blocks on the blockchain and utilizes MHT to verify data integrity. In this work, the blockchain stores all the hashtags of data blocks rather than the root of the MHT. In this case, a DO only needs to generate the hashtags of the file block to be updated instead of calculating the hashtags of all file blocks. Moreover, the DO can delegate auditing tasks to other DOs instead of a centralized TPA (i.e., this work supports auditing delegation). In addition, the auditor can perform multiple auditing tasks for different DOs simultaneously (i.e., this work supports batch auditing). The verification process does not leak any privacy information as auditors only verify the hashtag instead of original files. In terms of computational cost, the DO performs n multiplication operations, $2n$ exponentiation operations, n mapping-to-point hash operations in the **TagGen** phase. The CSP performs $n-1$ multiplication operations, n exponentiation operations in the **ProofGen** phase. The auditor performs two bilinear pairing operations, n multiplication operations, $n+1$ exponentiation operations, n mapping-to-point hash operations in the **Verification** phase. The communication cost of this scheme is $O(n)$. Despite the above benefits, this work still has some limitations. Firstly, traceability is not satisfied since this work does not store the operation logs of the data on the blockchain. Secondly, this work does not give formalized security proof of the proposed scheme. Thirdly, this work does not offer any incentive mechanisms and reputation evaluation mechanisms. Fourthly, this work cannot provide relevant measures to recover corrupted data once any corruption occurs.

El et al. [54] proposed a scheme for managing data deduplication and ensuring data integrity on the cloud server side based on the multi-agent system and the blockchain. In this work, the authors utilize MHT to support dynamic operations of the data. A DO can delegate auditing tasks to a TPA(i.e., this work supports auditing delegation). The authors give formalized proof that the verification process does not leak any privacy information of DOs. In addition, the TPA could perform multiple auditing tasks for different DOs simultaneously (i.e., this work supports batch auditing). Despite the above benefits, there are also some limitations that should not be ignored. Firstly, traceability is not satisfied as relevant operation logs of the data are not stored on the blockchain. Secondly, a formalized security proof of the scheme is not given in this work. Thirdly, this work does not utilize incentive mechanisms and reputation evaluation mechanisms. Fourthly, this work cannot provide relevant measures to recover corrupted data once any corruption is identified. In addition, the computational cost and the communication cost cannot be measured because this work does not provide specific details of each phase.

4.3. Signature-based schemes

4.3.1. RSA-based homomorphic methods

Yu et al. [40] proposed a BDIA scheme for cloud storage, which removes the semi-honest TPA. The researchers designed a Data Auditing Blockchain (DAB) to collect auditing proofs to verify data integrity. In this work, a DO firstly constructs HVTs for each file block. Then, the DO sends file blocks and corresponding RSA-based HVT to a CSP. After that, the DO generates challenge information and sends it to the CSP. After receiving the challenge information, the CSP generates an auditing proof and stores it on the blockchain. Finally, the DO can access and verify the auditing proof stored on blockchain. All auditing proofs are stored on blockchain (i.e., all operation logs on data are traceable). A formalized security proof of this scheme is given with a formalized method under a random oracle model. The security problem of this scheme can be reduced to a CDH problem. Moreover, this scheme utilizes MHT to support dynamic operations of the data. A DO can delegate auditing tasks to other DOs (i.e., this work supports auditing delegation), and this scheme supports batch auditing as auditors can execute multiple auditing tasks for multiple DOs simultaneously. The researchers also prove that auditors cannot obtain any privacy information as auditors cannot retrieve any raw data from the auditing proof. In terms of computational cost, the DO performs n multiplication operations, $2n$ exponentiation operations, n mapping-to-point hash operations in the **TagGen** phase. The CSP performs $z-1$ multiplication operations, $z+1$ exponentiation operations in the **ProofGen** phase. The auditor performs three bilinear pairing operations, $z+1$ multiplication operations, $z+3$ exponentiation operations, and z mapping-to-point hash operations in the **Verification** phase. In addition, the transmission of the challenge information and the proof information brings $O(1)$ communication cost. Although this work has the advantages mentioned above, there are some limitations that should not be ignored. Firstly, this work does not use incentive mechanisms. Secondly, it does not employ reputation evaluation mechanisms. In addition, it does not provide relevant measures to recover corrupted data once any corruption happens.

4.3.2. BLS-based homomorphic methods

Fan et al. [55] proposed a BDIA scheme called Dredas, in which a DO is able to obtain auditing results from the Ethereum without relying on the assumption of semi-honest TPA. In Dredas, the smart contract is deployed on the blockchain to perform auditing tasks. Such an auto-execution auditing manner is more secure than traditional methods. Moreover, a DO and a CSP must pay some ether for the smart contract as a deposit, which can be used to pay service fees. In this work, a DO can delegate auditing work to other miner nodes (i.e., this work supports auditing delegation). The security proof of this scheme is given through a formalized method under a standard model. The security problem of this scheme is based on the complexity of the CDH problem and a Discrete Log (DL) problem. The experiments indicate that an auditor can simultaneously handle 460 auditing tasks of challenge data blocks from approximately 500 DOs. In addition, a DO can utilize MHT to perform dynamic operations of outsourced data blocks. In terms of computational cost, the DO performs n multiplication operations, $2n$ exponentiation operations, n mapping-to-point hash operations in the **TagGen** phase. The CSP performs $2z+2$ exponentiation operations, and z mapping-to-point hash operations in the **ProofGen** phase. The auditor performs two bilinear pairing operations, $z-1$ multiplication operations, $z+1$ exponentiation operations in the **Verification** phase. The communication cost of this scheme is $O(z)$. Despite the above benefits,

this work still has some limitations. Firstly, traceability is not satisfied since the operation logs of the data are not stored on the blockchain. Moreover, this work does not employ reputation evaluation mechanisms. In addition, no measures are proposed to recover corrupted data once needed.

In the scheme of Yu et al. [40], a DO must traverse the blockchain to get the auditing proof, which will bring additional communication and computation overhead. To solve this problem, Huang et al. [56] proposed a Collaborative Auditing Blockchain (CAB) framework for cloud data storage, in which a DO can get auditing results from group managers without traversing the whole blockchain. In this work, all operation logs of a file block are traceable. A formalized security proof of this scheme is given with a formalized method under a standard model. The security of this scheme is based on the complexity of the CDH problem. The authors designed an Auxiliary Chain Table (ACT) to support dynamic operations of the data. Meanwhile, this work aggregates multiple challenges from various DOs to support batch auditing. In addition, this scheme can protect DO's identity privacy as auditors cannot distinguish which DO sends the request. A DO can delegate auditing tasks to other DOs (i.e., this work supports auditing delegation). Moreover, the researchers designed a series of feasible incentive mechanisms to encourage consensus nodes to perform tasks honestly. With the incentive mechanism, credit score can be used to evaluate reputation of CSPs; thus, DOs can choose the CSPs with high credit score to outsource their data to obtain high security guarantee. In terms of computational cost, the DO performs n multiplication operations, $2n$ exponentiation operations, n mapping-to-point hash operations in the **TagGen** phase. The CSP performs $2z-1$ multiplication operations, z exponentiation operations in the **ProofGen** phase. The auditor performs three bilinear pairing operations, and z multiplication operations, $z+1$ exponentiation operations, and z mapping-to-point hash operations in the **Verification** phase. The communication cost of this scheme is $O(z)$. However, this work cannot provide relevant measures to recover corrupted data once any corruption is detected.

Dong et al. [57] proposed a consortium BDIA scheme for IoT data, which utilizes the anonymity of the blockchain to achieve identity privacy protection. In this work, a DO generates BLS-based HVTs for each data block and sends data blocks with the corresponding HVTs to a CSP. Then, the DO publishes an auditing task through a smart contract. The CSP will return the corresponding proof to a TPA. Finally, the TPA utilizes a smart contract to verify the proof and stores auditing results of the blockchain. The smart contract in this scheme can be used to issue auditing tasks with corresponding point rewards. Each auditing task is assigned a corresponding completion time. The TPA who receives the auditing assignment must complete it within a fixed time interval to get rewards. Otherwise, it will be punished. A DO can delegate auditing tasks to a TPA (i.e., this work supports auditing delegation). In terms of computational cost, the DO performs n multiplication operations, $2n$ exponentiation operations, n mapping-to-point hash operations in the **TagGen** phase. The CSP performs z multiplication operations in the **ProofGen** phase. The auditor performs two bilinear pairing operations, $2z-1$ multiplication operations, $2z+1$ exponentiation operations, z mapping-to-point hash operations in the **Verification** phase. In addition, the transmission of the challenge information and the proof information brings $O(z)$ communication cost. Despite those benefits above, this work still has some limitations. Firstly, traceability is not satisfied as relevant operation logs of the data are not stored on blockchain. Secondly, a formalized security proof of the scheme is not given. Thirdly, the dynamic operations of the data are not supported in this scheme. Fourthly, this work does not support batch auditing since only one auditing operation can be executed at a time. Fifthly, this work cannot provide relevant measures to recover corrupted data once any corruption happens. At last, reputation evaluation mechanisms are not employed, which leads to some risks.

Li et al. [58] proposed a BDIA framework for smart cities. A formalized proof of this framework is given with a formalized method under a random oracle model. The security of this framework is based on the complexity of the CDH problem and a DL problem. An authenticated data structure based on a rank-based Merkle hash tree is utilized to support data dynamic operations. This framework is publicly verifiable for the DO who stores the file or a miner. Furthermore, the researchers designed a series of feasible incentive mechanisms. With the incentive mechanisms, only the honest CSP can get the rental fee from DOs. In addition, the framework implements multiple replicas storage, which means that the DO who can retrieve the redundant replicas is able to recover the original file. In terms of computational cost, the DO performs n^2 multiplication operations, $n(n+1)$ exponentiation operations, n mapping-to-point hash operations in the **TagGen** phase. The CSP performs $z(2z-2)$ multiplication operations, z^2 exponentiation operations in the **ProofGen** phase. The auditor performs two bilinear pairing operations, z^2 multiplication operations, $z(z+1)$ exponentiation operations, z mapping-to-point hash operations in the **Verification** phase. The communication cost of this work is $O(zlogn)$. Although this work has the advantages mentioned above, it suffers from some limitations that should not be ignored. Firstly, traceability is not satisfied as this work does not store the operation logs of the data on the blockchain. Secondly, this work does not support batch auditing since only one auditing operation can be performed at a time. Thirdly, the privacy protection is not considered in this paper. Fourthly, this work does not employ reputation evaluation mechanisms.

4.3.3. ID-based homomorphic methods

Zhang et al. [59] proposed a Certificateless Public Verification scheme against Procrastinating Auditors (CPVPA), which resists malicious or procrastinating auditors without involving any semi-honest TPA. Each verification executed by auditors is time-stamped and recorded in transactions. This mechanism enables DOs to check whether auditors finish auditing works within the specific time. A formalized security proof of the scheme is given through a formalized method under a random oracle model. The security problem of this scheme can be reduced to a CDH problem. A DO can delegate auditing tasks to a TPA (i.e., this work achieves auditing delegation). In terms of computational cost, the DO performs $4n$ multiplication operations, $5n+1$ exponentiation operations, $n+2$ mapping-to-point hash operations in the **TagGen** phase. The CSP performs $z-1$ multiplication operations, z exponentiation operations in the **ProofGen** phase. The auditor performs four bilinear pairing operations, $3z+1$ multiplication operations, and $3z+2$ exponentiation operations in the **Verification** phase. In addition, the transmission of the challenge information and the proof information brings $O(1)$ communication cost. In spite of those advantages above, there are also some limitations that should not be ignored. Firstly, traceability is not satisfied as relevant operation logs on data are not stored on the blockchain. Secondly, the TPA may collude with the CSP to obtain privacy information of DOs. Thirdly, this work does not support batch auditing since only one auditing operation can be executed at a time. Fourthly, dynamic data operation is unfortunately not supported. Fifthly, this work cannot provide relevant measures to recover corrupted data once needed. Finally, incentive mechanisms and reputation evaluation mechanisms are not utilized.

Zhao et al. [60] proposed a BDIA scheme without involving trusted TPAs. The scheme solves the problem of privacy leakage of DOs and is resistant to collusion attacks by CSPs and TPAs. In this work, a formalized security proof of the proposed scheme is given under a standard model. The security problem of this scheme can be reduced to a DL problem. The authors also give proof that auditors cannot obtain any privacy information of DOs in the verification process. Meanwhile, the authors extend the scheme to allow a DO to perform dynamic data operation and support batch auditing. A DO can delegate auditing tasks to other DOs (i.e., this work supports auditing delegation). Moreover, the auditor will get some rewards after the verification work. Regarding computational cost, the DO performs $5n$ multiplication operations, n mapping-to-point hash operations in the **TagGen** phase. The CSP performs $2z$ multiplication

operations, z mapping-to-point hash operations in the **ProofGen** phase. The auditor performs two bilinear pairing operations, one multiplication operation, z mapping-to-point hash operations in the **Verification** phase. In addition, the transmission of the challenge information and the proof information brings $O(1)$ communication cost. Although this work has the advantages mentioned above, it suffers from some limitations that should not be ignored. Firstly, traceability is not satisfied since this work does not store the operation logs of the data on the blockchain. Secondly, this work does not employ reputation evaluation mechanisms. Thirdly, this work cannot provide relevant measures to recover corrupted data.

4.3.4. Short signature homomorphic methods

Wang et al. [61] proposed a Blockchain and Bilinear mapping-based Data Integrity Scheme (BB-DIS) for large-scale IoT data in cloud storage, which solves the trust problem of TPA and reduces large computational and communication overhead. Firstly, they combine smart contracts with bilinear mapping to achieve the verification of data integrity. Secondly, in this work, a DO divides its file into multiple shards and generates HVTs for each shard. Thirdly, a formalized security proof of the scheme is given with a formalized method under a standard model. In addition, the authors designed an update request algorithm and an update execution algorithm to support dynamic data operation. Furthermore, a DO can delegate auditing work to other miner nodes (i.e., this work supports auditing delegation). In terms of computational cost, the DO performs n multiplication operations and n mapping-to-point hash operations in the **TagGen** phase. The CSP performs $4z + 1$ multiplication operations, and z mapping-to-point hash operations in the **ProofGen** phase. The auditor performs three bilinear pairing operations in the **Verification** phase. The communication cost of this scheme is $O(n)$. Despite the merits above, there are also some limitations that should not be ignored. Firstly, traceability is not satisfied as relevant operation logs of the data are not stored on the blockchain. Secondly, this work does not support batch auditing since only one auditing operation can be executed at a time. Thirdly, it cannot provide relevant measures to recover corrupted data once needed. Fourthly, the protection of privacy is not considered. Fifthly, reputation evaluation and incentives are not considered.

5. Open issues and future research directions

Based on the review and evaluation of existing BDIA schemes, we list some open issues, followed by potential future research directions in order to promote future research of BDIA.

5.1. Open issues

Based on the above review and the comparative analysis of the literature, we propose some open issues in the research of BDIA schemes as follows.

Firstly, data recovery has been neglected in most literature. For a data integrity auditing scheme, it is not enough to simply identify the misbehavior of the cloud server. For some DOs, recovering corrupted outsourced data is necessary as they do not back up these outsourced data. Traditional TPA-based data integrity auditing schemes utilize Error Correcting Code (ECC) (e.g., parity check codes) [62] to support data recovery. However, such a method is not applicable to the blockchain as it results in a significant increase in data volume. Therefore, how to support data recovery in a BDIA scheme is an open issue that deserves further effort.

Secondly, the existing works rarely consider reputation evaluation. Reputation evaluation is to quantify reputation based on the experiences of earlier interactions of entities. In our review, the incentive mechanisms in existing BDIA schemes generally reward honest CSPs if they are honest enough to complete their work on time. Otherwise, they will be punished if they work dishonestly maliciously or collude with others. However, if the incentive mechanisms lack reputation evaluation, it is hard for DOs to distinguish between honest and dishonest entities. Entities in the former category have a history of honest behaviors, and entities in the latter category have a history of illegal or dishonest behaviors. How to find an effective and fair standard to evaluate the reputation of CSPs is difficult, but it is worth exploring in the future.

Thirdly, in a BDIA scheme, the cloud should be required to provide the latest data and the historical versions of the data (or data operation logs). With data operation logs, DOs can detect illegal operations of CSPs. However, existing works lack support for traceability as they only store the latest version of the data and the historical versions are abandoned. Although operation logs can be stored on the blockchain to support traceability, they will cause inefficiency in processing transactions due to the increase in operation logs. Therefore, how to design an efficient method to support traceability is an open issue that should be considered seriously.

Fourthly, efficient dynamic operation on large-scale data is not well considered in most existing BDIA schemes. Nowadays, many big data applications utilize CSPs to store large amounts of data, which is updated by DOs frequently. In order to support dynamic data operation, most BDIA schemes utilize a verifiable data structure like MHT. However, MHT must keep balanced after a dynamic data operation. The computation overhead will grow exponentially when DOs update large-scale data frequently. Therefore, how to reduce the computation overhead of MHT updating that happens in the dynamic operation on big data is an open issue that needs to be seriously studied further.

Fifthly, the research on security in BDIA schemes is not thorough. Some potential internal attacks (e.g., replacement, replay, and collusion attacks) that can damage data integrity are still not seriously considered. Although existing BDIA schemes utilize the blockchain technology to eliminate the semi-honest TPA, the CSP is still possible to launch some internal attacks. Therefore, how to detect as many potential internal attacks as possible and deploy corresponding defense strategies is an open issue that deserves more efforts.

5.2. Future research directions

Based on the above review and discussion of open issues, we realize that there is still a long way to go before the blockchain technology can be applied to data integrity auditing of cloud data. In this subsection, we point out some potential future research directions for the research of blockchain-based cloud data integrity auditing technology.

Firstly, data recovery is expected in BDIA schemes. One direct solution is that DOs store multiple copies of a file on multiple CSPs in a blockchain network. By this way, even though a file copy is corrupted in one CSP, DOs are able to retrieve the file from another CSPs. However, this solution requires DOs to generate multiple copies of the same file, which may lead to duplicated storage and storage waste. Therefore, designing an economical BDIA solution that can support data recovery with high-efficiency is a significant future research topic.

Secondly, reputation evaluation is an important mechanism in BDIA schemes. Credit score could be a good standard to evaluate the reputation of CSPs. A CSP will get a specified number of scores if it provides legal cloud service; otherwise, its score will be deducted. By this way, DOs can choose the CSPs with high credit score to outsource their data for a high security guarantee. Therefore, designing a mechanism for evaluating the reputation of CSPs in BDIA schemes is a promising direction.

Thirdly, traceability is essential to research in BDIA schemes. Due to the limited performance of distributed nodes, storing large amounts of operation logs of the blockchain incurs additional storage overhead and reduces the processing speed of transactions. One solution that might be considered is to introduce InterPlanetary File System (IPFS). Such a decentralized file system is efficient and secure for storing large amounts of operation logs as it uses content-addressing to uniquely identify each file and makes full use of the storage space of each node in the network. Therefore, designing a lightweight method to store operation logs is an interesting research topic.

Fourthly, reducing the computation overhead of a dynamic data

operation is essential in BDIA schemes. For existing schemes that support dynamic data operation based on MHT, optimizing updating strategy is an effective solution to reduce computational overhead. Besides, it is better to design an efficient data structure that supports dynamic data operation. A double-linked list could be used to store file block information. This data structure can reduce computational overhead as dynamic operations on a file block, will cause no change in other file blocks but suffers from a high storage overhead. Therefore, how to design an efficient data structure that supports dynamic data operations is a significant research topic.

Fifthly, attack detection and defense are important in BDIA schemes. For attack detection, it might be useful to analyze detection strategies for similar attacks in other scenarios (e.g., P2P) and apply these strategies into BDIA schemes after improvement. For attack defense, increasing randomness in a BDIA scheme could be a good method to prevent some potential internal attacks. Furthermore, some Pseudo-Random Functions could be used for masking the location of entities. In this case, it is difficult for CSPs to collude with other entities. Obviously, detecting and defending against potential internal attacks are essential research topics worthy of special efforts.

6. Conclusions

BDIA schemes are mainly proposed to resist malicious auditors and solve single points and performance limitations of traditional TPA. In this paper, we conducted a thorough survey on BDIA schemes. We first introduced the basic knowledge of BDIA. Then, we proposed a set of criteria for evaluating existing BDIA schemes. After that, we proposed a taxonomy of the existing schemes according to the difference in metadata generation. We thoroughly reviewed, analyzed and compared existing works by employing the proposed criteria. Based on our serious review, we found some open issues and then went ahead to propose future research directions.

Declaration of competing interest

There is no conflict of interest.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62072351; in part by the Academy of Finland under Grant 308087, Grant 335262, Grant 345072, and Grant 350464; in part by the Open Project of Zhejiang Lab under Grant 2021PD0AB01; and in part by the 111 Project under Grant B16037.

References

[1] H. Tabrizchi, M. Kuchaki Rafsanjani, A survey on security challenges in cloud computing: issues, threats, and solutions, J. Supercomput. 76 (12) (2020) 9493–9532.
[2] K. Yang, X. Jia, Data storage auditing service in cloud computing: challenges, methods and opportunities, World Wide Web 15 (4) (2012) 409–428.
[3] D.A. Fernandes, L.F. Soares, J.V. Gomes, M.M. Freire, P.R. Inácio, Security issues in cloud environments: a survey, Int. J. Inf. Secur. 13 (2) (2014) 113–170.
[4] A. Ghobadi, R. Karimi, F. Heidari, M. Samadi, Cloud computing, reliability and security issue, in: 16th International Conference on Advanced Communication Technology, IEEE, 2014, pp. 504–511.
[5] A. Li, Y. Chen, Z. Yan, X. Zhou, S. Shimizu, A survey on integrity auditing for data storage in the cloud: from single copy to multiple replicas, IEEE Trans. Big Data (2020) 1–17, https://doi.org/10.1109/TBDATA.2020.3029209.
[6] L. Zhou, A. Fu, S. Yu, M. Su, B. Kuang, Data integrity verification of the outsourced big data in the cloud environment: a survey, J. Netw. Comput. Appl. 122 (2018) 1–15.
[7] F. Zafar, A. Khan, S.U.R. Malik, M. Ahmed, A. Anjum, M.I. Khan, N. Javed, M. Alam, F. Jamil, A survey of cloud computing data integrity schemes: design challenges, taxonomy and future trends, Comput. Secur. 65 (2017) 29–49.
[8] M. Sookhak, H. Talebian, E. Ahmed, A. Gani, M.K. Khan, A review on remote data auditing in single cloud server: taxonomy and open issues, J. Netw. Comput. Appl. 43 (2014) 121–141.
[9] W.-F. Hsien, C.C. Yang, M.-S. Hwang, A survey of public auditing for secure data storage in cloud computing, Int. J. Netw. Secur. 18 (1) (2016) 133–142.
[10] C.B. Tan, M.H.A. Hijazi, Y. Lim, A. Gani, A survey on proof of retrievability for cloud data integrity and availability: cloud storage state-of-the-art, issues, solutions and future trends, J. Netw. Comput. Appl. 110 (2018) 75–86.
[11] M. Sookhak, A. Gani, H. Talebian, A. Akhunzada, S.U. Khan, R. Buyya, A.Y. Zomaya, Remote data auditing in cloud computing environments: a survey, taxonomy, and open issues, ACM Comput. Surv. 47 (4) (2015) 1–34.
[12] A. Barsoum, Provable data possession in single cloud server: a survey, classification and comparative study, Int. J. Comput. Appl. 123 (9) (2015) 1–10.
[13] M. Thangavel, P. Varalakshmi, R. Sindhuja, S. Sridhar, A survey on provable data possession in cloud storage, in: 2016 Eighth International Conference on Advanced Computing (ICoAC), IEEE, 2017, pp. 25–31.
[14] Y. Dong, L. Sun, D. Liu, M. Feng, T. Miao, A survey on data integrity checking in cloud, in: 2018 1st International Cognitive Cities Conference (IC3), IEEE, 2018, pp. 109–113.
[15] K. Gangadevi, R.R. Devi, A survey on data integrity verification schemes using blockchain technology in cloud computing environment, in: IOP Conference Series: Materials Science and Engineering, vol. 1110, IOP Publishing, 2021, 012011.
[16] A. Li, S. Tan, Y. Jia, A method for achieving provable data integrity in cloud computing, J. Supercomput. 75 (1) (2019) 92–108.
[17] S.K. Nayak, S. Tripathy, Sepdp: secure and efficient privacy preserving provable data possession in cloud storage, IEEE Trans. Serv. Comput. 14 (3) (2021) 876–888.
[18] N.S. Chavan, D. Sharma, Secure proof of retrievability system in cloud for data integrity, in: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), IEEE, 2018, pp. 1–5.
[19] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2008. (Accessed 1 July 2015).
[20] Y. Yu, M.H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, G. Min, Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage, IEEE Trans. Inf. Forensics Secur. 12 (4) (2017) 767–778.
[21] Y. Yu, Y. Li, J. Tian, J. Liu, Blockchain-based solutions to security and privacy issues in the internet of things, IEEE Wireless Commun. 25 (6) (2018) 12–18.
[22] Y. Yu, Y. Li, B. Yang, W. Susilo, G. Yang, J. Bai, Attribute-based cloud data integrity auditing for secure outsourced storage, IEEE Trans. Emerg. Top. Com. 8 (2) (2020) 377–390.
[23] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, K.-K.R. Choo, Fuzzy identity-based data integrity auditing for reliable cloud storage systems, IEEE Trans. Depend. Secure. 16 (1) (2019) 72–83.
[24] Y. Li, Y. Yu, R. Chen, X. Du, M. Guizani, Integritychain: provable data possession for decentralized storage, IEEE J. Sel. Area. Commun. 38 (6) (2020) 1205–1217.
[25] B. Moustapha, The effect of propagation delay on the dynamic evolution of the bitcoin blockchain, Digit. Commun. Netw. 6 (2) (2020) 157–166.
[26] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, F.-Y. Wang, An overview of smart contract: architecture, applications, and future trends, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 108–113.
[27] L. Peng, W. Feng, Z. Yan, Y. Li, X. Zhou, S. Shimizu, Privacy preservation in permissionless blockchain: a survey, Digit. Commun. Netw. 7 (3) (2021) 295–307.
[28] Z. Yan, L. Peng, W. Feng, L.T. Yang, Social-chain: decentralized trust evaluation based on blockchain in pervasive social networking, ACM Trans. Internet Technol. 21 (1) (2021) 1–28.
[29] Y. Wu, Z. Yan, F.R. Yu, R. Deng, V. Varadharajan, W. Chen, Guest editorial: blockchain and healthcare computing, IEEE J. Biomed. Health. 24 (8) (2020) 2144–2145.
[30] W. Feng, Y. Li, X. Yang, Z. Yan, L. Chen, Blockchain-based data transmission control for tactical data link, Digit. Commun. Netw. 7 (3) (2021) 285–294.
[31] D.D.F. Maesa, P. Mori, Blockchain 3.0 applications survey, J. Parallel Distr. Comput. 138 (2020) 99–114.
[32] R. Zhang, R. Xue, L. Liu, Security and privacy on blockchain, ACM Comput. Surv. 52 (3) (2019) 1–34.
[33] W. Feng, Z. Yan, Mcs-chain: decentralized and trustworthy mobile crowdsourcing based on blockchain, Future Generat. Comput. Syst. 95 (2019) 649–666.
[34] G. Liu, H. Dong, Z. Yan, X. Zhou, S. Shimizu, B4sdc: a blockchain system for security data collection in manets, IEEE Trans. Big Data 7 (6) (2020) 5329–5344, https://doi.org/10.1109/TBDATA.2020.2981438.
[35] K.-K.R. Choo, Z. Yan, W. Meng, Blockchain in industrial iot applications: security and privacy advances, challenges, and opportunities, IEEE Trans. Ind. Inf. 16 (6) (2020) 4119–4121.
[36] W. Feng, Z. Yan, L.T. Yang, Q. Zheng, Anonymous authentication on trust in blockchain-based mobile crowdsourcing, IEEE Internet Things J. 9 (16) (2022) 14185–14202, https://doi.org/10.1109/JIOT.2020.3018878.
[37] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: a survey, Int. J. Web Grid Serv. 14 (4) (2018) 352–375.
[38] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, Y. Zhang, Consortium blockchain for secure energy trading in industrial internet of things, IEEE Trans. Ind. Inf. 14 (8) (2017) 3690–3700.
[39] S. Pahlajani, A. Kshirsagar, V. Pachghare, Survey on private blockchain consensus algorithms, in: 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), IEEE, 2019, pp. 1–6.
[40] H. Yu, Z. Yang, R.O. Sinnott, Decentralized big data auditing for smart city environments leveraging blockchain technology, IEEE Access 7 (2018) 6288–6296.
[41] L. Lamport, R. Shostak, M. Pease, The Byzantine Generals Problem, Association for Computing Machinery, New York, NY, USA, 2019, pp. 203–226.
[42] Y. Zhang, J. Ni, X. Tao, Y. Wang, Y. Yu, Provable multiple replication data possession with full dynamics for secure cloud storage, Concurrency Comput. Pract. Ex. 28 (4) (2016) 1161–1173.

[43] Q. Wang, C. Wang, K. Ren, W. Lou, J. Li, Enabling public auditability and data dynamics for storage security in cloud computing, IEEE Trans. Parallel Distr. Syst. 22 (5) (2010) 847–859.
[44] H. Wang, Q. Wu, B. Qin, J. Domingo-Ferrer, Identity-based remote data possession checking in public clouds, IET Inf. Secur. 8 (2) (2014) 114–121.
[45] F. Zhang, R. Safavi-Naini, W. Susilo, An efficient signature scheme from bilinear pairings and its applications, in: International Workshop on Public Key Cryptography, Springer, 2004, pp. 277–290.
[46] Y. Yu, M.H. Au, Y. Mu, S. Tang, J. Ren, W. Susilo, L. Dong, Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage, Int. J. Inf. Secur. 14 (4) (2015) 307–318.
[47] Y. Wu, W. Meng, Z. Yan, V. Varadharajan, Special issue on blockchain and communication networks, Digit. Commun. Netw. 6 (2) (2020) 145–146.
[48] K. Ren, C. Wang, Q. Wang, Security challenges for the public cloud, IEEE Internet Comput. 16 (1) (2012) 69–73.
[49] J. Ryoo, S. Rizvi, W. Aiken, J. Kissell, Cloud security auditing: challenges and emerging approaches, IEEE Secur. Priv. 12 (6) (2013) 68–74.
[50] H. Wang, H. Qin, M. Zhao, X. Wei, H. Shen, W. Susilo, Blockchain-based fair payment smart contract for public cloud storage auditing, Inf. Sci. 519 (2020) 348–362.
[51] D. Yue, R. Li, Y. Zhang, W. Tian, Y. Huang, Blockchain-based verification framework for data integrity in edge-cloud storage, J. Parallel Distr. Comput. 146 (2020) 1–14.
[52] B. Liu, X.L. Yu, S. Chen, X. Xu, L. Zhu, Blockchain based data integrity service framework for iot data, in: 2017 IEEE International Conference on Web Services (ICWS), IEEE, 2017, pp. 468–475.
[53] J. Li, J. Wu, G. Jiang, T. Srikanthan, Blockchain-based public auditing for big data in cloud storage, Inf. Process. Manag. 57 (6) (2020), 102382.
[54] M. El Ghazouani, M.A. El Kiram, L. Er-Rajy, Blockchain & multi-agent system: a new promising approach for cloud data integrity auditing with deduplication, Int. J. Commun. Network. Inf. Secur. 11 (1) (2019) 175–184.
[55] K. Fan, Z. Bao, M. Liu, A.V. Vasilakos, W. Shi, Dredas: decentralized, reliable and efficient remote outsourced data auditing scheme with blockchain smart contract for industrial iot, Future Generat. Comput. Syst. 110 (2020) 665–674.
[56] P. Huang, K. Fan, H. Yang, K. Zhang, H. Li, Y. Yang, A collaborative auditing blockchain for trustworthy data integrity in cloud storage system, IEEE Access 8 (2020) 94780–94794.
[57] G. Dong, X. Wang, A secure iot data integrity auditing scheme based on consortium blockchain, in: 2020 5th IEEE International Conference on Big Data Analytics (ICBDA), IEEE, 2020, pp. 246–250.
[58] R. Chen, Y. Li, Y. Yu, H. Li, X. Chen, W. Susilo, Blockchain-based dynamic provable data possession for smart cities, IEEE Internet Things J. 7 (5) (2020) 4143–4154.
[59] Y. Zhang, C. Xu, X. Lin, X.S. Shen, Blockchain-based public integrity verification for cloud storage against procrastinating auditors, IEEE Trans. Cloud Comput. 9 (3) (2021) 923–937.
[60] Q. Zhao, S. Chen, Z. Liu, T. Baker, Y. Zhang, Blockchain-based privacy-preserving remote data integrity checking scheme for iot information systems, Inf. Process. Manag. 57 (6) (2020), 102355.
[61] H. Wang, J. Zhang, Blockchain based data integrity verification for large-scale iot data, IEEE Access 7 (2019) 164996–165006.
[62] D. Tiwari, G. Gangadharan, A novel secure cloud storage architecture combining proof of retrievability and revocation, in: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2015, pp. 438–445.