IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON AI-DRIVEN BIG DATA PROCESSING: THEORY, METHODOLOGY, AND APPLICATIONS

Received November 29, 2018, accepted December 10, 2018, date of publication December 20, 2018, date of current version January 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2888940

Decentralized Big Data Auditing for Smart City Environments Leveraging Blockchain Technology

HAIYANG YU[1,2], ZHEN YANG[1], (Member, IEEE), AND RICHARD O. SINNOTT[2]

[1]College of Computer Science, Beijing University of Technology, Beijing 100124, China
[2]School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC 3010, Australia

Corresponding author: Zhen Yang (yangzhen@bjut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61671030, in part by the National Key R&D Program of China under Grant 2016YFB0801100 and Grant 2018YFB0803600, and in part by the Industrial Internet Innovation Development Project.

ABSTRACT The idea of big data has gained extensive attention from governments and academia all over the world. It is especially relevant for the establishment of a smart city environment combining complex heterogeneous data with data analytics and artificial intelligence (AI) technology. Big data is generated from many facilities and sensor networks in smart cities and often streamed and stored in the cloud storage platform. Ensuring the integrity and subsequent auditability of such big data is essential for the performance of AI-driven data analysis. Recent years has witnessed the emergence of many big data auditing schemes that are often characterized by third party auditors (TPAs). However, the TPA is a centralized entity, which is vulnerable to many security threats from both inside and outside the cloud. To avoid this centralized dependency, we propose a decentralized big data auditing scheme for smart city environments featuring blockchain capabilities supporting improved reliability and stability without the need for a centralized TPA in auditing schemes. To support this, we have designed an optimized blockchain instantiation and conducted a comprehensive comparison between the existing schemes and the proposed scheme through both theoretical analysis and experimental evaluation. The comparison shows that lower communication and computation costs are incurred with our scheme than with existing schemes.

INDEX TERMS Big data, smart city, data auditing, blockchain.

I. INTRODUCTION

With the explosion of the population, increasing urbanization is taking place globally for the last century. As one example, Beijing only had a population of 1.5 million in 1915 but has since grown to over 20 million in 2017. Indeed there were only 20 cities that had over one million people a century ago, but now the number has surpassed 450, and it seems likely that this trend will remain so for the foreseeable future. With the continued increase of the urban population, many challenges and problems arise, such as traffic congestion, waste pollution and energy and water challenges. To tackle these new problems, the concept of smart cities has been proposed. A smart city refers to the intelligent collection and analysis of all kinds of data created in cities through increasing digitization that is occurring. Such data can be used to provide better public services and build more sustainable urban environments. However, since the volume of data can be extremely large and it is often created in real-time, it is impossible for humans to process them. Scalable technical solutions and architectures for data acquisition and analysis are required. Thus, as a powerful technology of processing and analyzing urban data, artificial intelligence (AI) is integrated into smart city to help produce analysis results and make conclusions for improving public services. Such conclusions and results are all based on the collected big data, and can be used to influence polices and decisions that have a direct consequence on societies living within the cities. As a result, any incorrect or incomplete data may affect the analysis made by AI and produce results that negatively impact the cities in which we live. Figure 1 shows one such the 'Big Data Reference Architecture' from the National Institute of Standards and Technology (NIST) [5]. From this figure, we can see that without the guarantee of big data integrity, both data providers and big data framework providers may offer incorrect or corrupted data to big data application providers, which will affect the results of all AI-based big data applications [10], [13]. As one example, in the transport domain, intelligent transportation services with incorrect car locations

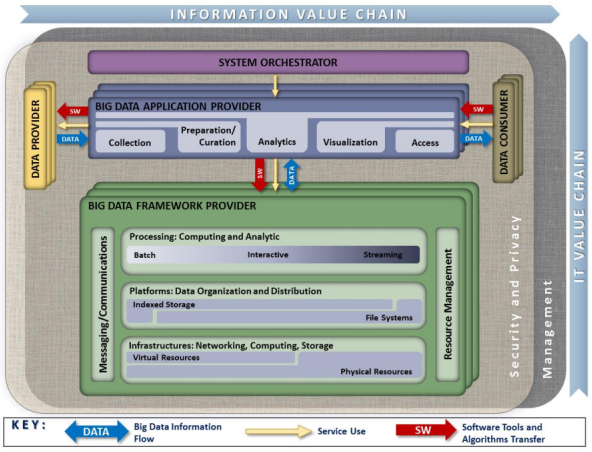

FIGURE 1. NIST Big Data Reference Architecture [5].

provided by smart cars will make erroneous judgement on traffic condition and offer wrong instructions to smart cars, which may lead to serious car accidents. Thus, guaranteeing the integrity of big data in smart city is of critical importance to this scenario.

To tackle smart city environment challenges, many cities and organizations are adopting cloud platforms as basis for their storage needs. Although almost all urban data can be stored in cloud platforms, the integrity of data cannot always be guaranteed. Any system outage or malicious attack could result in serious data loss, which is a big security threat in smart cities. In order to ensure the integrity of massive cloud-based data, many big data auditing schemes for cloud storage have been proposed. In many cases, a third party auditor (TPA) is introduced into these schemes to perform auditing tasks on behalf of data owners or users. However, there are several problems caused by the introduction of the TPA. Although the assumption is that the TPA is a trusted entity and always behaves in an honest way, in practice, the TPA may not be so reliable as expected. A TPA may even be bribed by cloud service providers (CSPs) to help them hide data corruption incidents. Furthermore, as a centralized entity, there is a single point of failure which could have catastrophic consequences. Both external attacks and internal management faults can give rise to TPA system outages. To address this, our work takes into consideration these concerns and proposes a decentralized auditing framework based on blockchain technology.

Blockchain is a technology enabling a fully decentralized system. Satoshi [14] first proposed blockchain as an underlying technology of Bitcoin and other cryptocurrencies. Its major advantages include its decentralization, trustless consensus, tamper-proof, traceable and collective maintenance. Blockchain has gained extensive attention from financial institutions, enterprises and academia [27]. Some schemes such as Sia [17] and Lambda [1] have been proposed to combine data integrity checking with blockchain-based storage. However, these schemes are not practical until decentralized data storage solutions reach an acceptable level of efficiency.

In this paper, we propose a decentralized big data auditing solution targeted specifically to the needs of smart city environments. The original contributions in our paper can be summarized as the following four aspects:

- We propose a blockchain-based big data integrity auditing scheme, which improves the reliability and stability of auditing schemes through the elimination of the TPA;
- We propose an optimized blockchain instantiation called data auditing blockchain (DAB) that collects auditing proofs instead of bitcoin transactions and uses a consensus algorithm based on a variant of the Practical Byzantine Fault Tolerance (PBFT) algorithm;
- We extend our work to support batch auditing as well as dynamic auditing;
- We show that the security, reliability and efficiency of our scheme are favorable.

The remainder of the paper is structured as follows. Related work is presented in Section II. The problem statement is presented in Section III. We give our detailed solution in Section IV. The security analysis is given in Section V. The evaluation of the proposed scheme is presented in Section VI. Finally, we draw conclusions in Section VII.

II. RELATED WORK

Ateniese *et al.* [2] first presented the notion of provable data possession (PDP), which could check the integrity of data on remote servers. They constructed homomorphic verifiable tags (HVTs) on the basis of Rivest-Shamir-Adleman (RSA)-based signatures. Their scheme also adopted a sampling strategy to check data integrity with high probability. Juels and Kalisk, Jr., [9] first proposed the concept of proof of retrievability (POR). By adopting erasure codes, POR could check the integrity of remote big data whilst retrieving corrupted data files. Yang *et al.* [25] presented an information retrieval framework for the cloud characterized by its retrieval risk formula, which could effectively retrieve keywords [24] from encrypted data in the cloud without undermining key word privacy and retrieval performance. Curtmola *et al.* [6] first presented a multi-replica PDP scheme, which could verify the integrity of multiple replicas on remote servers. This scheme, however, could not support dynamic data operations. Ateniese *et al.* [3] proposed an efficient PDP scheme that supports all dynamic data operations except data insertion. Erway *et al.* [7] first proposed a fully dynamic PDP scheme, which introduces a rank-based authenticated skip list to maintain the dynamic information of data blocks. Their scheme, however, required extra auxiliary authentication information (AAI) during verification. Zhu *et al.* [31] proposed a dynamic auditing scheme associated with index hash tables. Barsoum and Hasan [4] presented a provable multi-copy dynamic data possession approach by introducing a map-version table. In addition, more and more researchers have pointed out that the data distribution characteristics, Gaussian

and Non-Gaussian [10]–[12], [20], should be considered in big data analysis.

Shacham and Waters [15] proposed a Boneh-Lynn-Shacham (BLS) signature-based auditing scheme, which incurred much less communication costs and supported public auditing. Following this idea, Wang *et al.* [19] first introduced a third party auditor (TPA) into the PDP scheme to verify cloud data on behalf of cloud users. The TPA could eliminate the involvement of cloud users through auditing and hence save computational overheads of users. In addition, by leveraging the homomorphic property of HVT, the TPA could support batch auditing, which allowed to perform multiple auditing tasks of different users simultaneously. However, the scheme assumed that the TPA was fully trusted, which cannot always be guaranteed. Furthermore, their scheme violated privacy preservation so that the TPA could derive data owner's data blocks using enough linear combinations of a same set of data blocks. Wang *et al.* [18] extended [19] and proposed a privacy-preserving public auditing scheme that could prevent the TPA from learning the content of cloud user data during the auditing process. This scheme assumed that the TPA was semi-trusted and honestly executed the auditing scheme whilst being curious about cloud user data.

Although many recent remote data integrity auditing schemes [8], [16], [22], [28] have utilized the TPA to support delegated auditing or public auditing, they have a range of drawbacks. First, the assumption of a trusted TPA is an obvious limitation since a third party can never be fully trusted. For instance, the TPA may behave in a dishonest way to retrieve data owner's secret files and may be bribed to hide a CSP's data corruption incidents. Second, in practice, the TPA cannot be completely stable and reliable as expected and they may suffer many security risks such as hacker attacks and internal system failures. Third, the TPA may become the bottleneck in the TPA-based auditing scheme especially when considering that the TPA needs to provide auditing services for potentially millions of data owners and users in the cloud. Too many connections and too many requests can cause performance degradation of the TPA. To address these issues, we present how blockchain technology can be used to build a decentralized auditing architecture. This can be used to build trust between data owners/users and the CSP without any TPA. Meanwhile, this decentralized structure is more robust and reliable than TPA-based schemes. To the best of our knowledge, this is the first work to introduce blockchain technology into cloud data integrity auditing whilst supporting decentralized auditing architecture.

III. PROBLEM STATEMENT

A. SYSTEM MODEL

Figure 2 depicts the decentralized architecture of the proposed scheme. It consists of two kinds of entities: data owners/users and the CSP. Data owners upload their data to the CSP. Data owners/users utilize big data stored in the CSP to conduct data analysis. As part of this, they need to perform integrity checking to ensure the quality and integrity of the data. The CSP provides significant storage space for data owners and should respond to auditing requests of data owners/users. The data auditing blockchain (DAB) consists of nodes of all data owners/users and the CSP. The blocks in the DAB store auditing proofs that can be accessed and verified by any node to check the integrity of data owners' data. A data owner/user first sends an auditing request to the CSP. The CSP then generates an auditing proof and broadcasts it in the DAB system. Finally, the auditing proof is included in a block, which will eventually be added to the DAB.

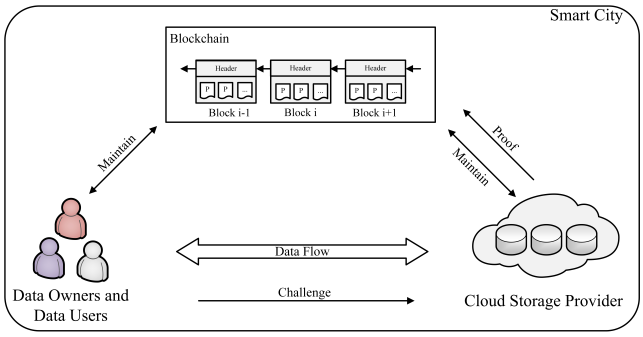

FIGURE 2. Decentralized auditing architecture in smart city.

B. SYSTEM COMPONENTS

The proposed scheme includes the following algorithms:

1) Representative Selection: nodes in a blockchain network comprise data owners/users and CSPs. All nodes select a master node and some slave nodes as representative nodes;
2) Key Generation: each data owner generates a public-secret key pair and associated public parameters. The data owner makes its public key and associated parameters public;
3) Tag Generation: the data owner generates HVTs for all of its files and uploads all files as well as HVTs to the CSP;
4) Challenge: the data owner/user sends a challenge request to the CSP to randomly check the data blocks in the cloud;
5) Response: when the CSP receives a challenge request, it generates an auditing proof and broadcasts it to all nodes;
6) Consensus: representative nodes store the auditing proofs and communicate with each other following which they publish new blocks and manage the blockchain together;
7) Verification: the data owner or users access the auditing proof from the DAB and verify it to get the auditing results.

C. THREAT MODEL AND DISIGN GOALS

We assume that the CSP is untrusted. That is, it will try not to honestly store data owner's data and may hide data

corruptions or even discard the corrupted data to maintain its reputation. We assume that over 50% of data owners/users are honest and hence can be trusted.

The proposed scheme is intended to meet the following design goals:

- Decentralization: auditing proofs should be stored and managed by all nodes, rather than one node so that they can be accessed by all data owners and users;
- Public auditability: a data owner's data can be checked by any other owner or user;
- Privacy preservation: it is not possible for any owner/user to derive specific information of another owner's data from the auditing proofs stored in the DAB;
- Batch auditing: a data owner or data user can verify multiple auditing proofs simultaneously;
- Traceability: all of the auditing history can be accessed by anyone and cannot be tampered with.

IV. THE PROPOSED SCHEME

A. PRELIMINARIES

Bilinear pairings. Let G_1 and G_T be two multiplicative cyclic groups of large prime order q. Let g_1 and g_2 be the generators of G_1 and G_T, respectively. A bilinear map is a map $e : G_1 \times G_1 \rightarrow G_T$ with three properties:

1) Bilinear. For $u, v \in G_1$ and $a, b \in Z_q^*$, $e(u^a, v^b) = e(u, v)^{ab}$;
2) Non-degenerate. $\exists g_1, g_2 \in G_1$ such that $e(g_1, g_2) \neq 1$;
3) Computable. It is efficient for an algorithm to compute the map e.

B. DATA AUDITING BLOCKCHAIN (DAB)

The DAB is an optimized instantiation designed for remote data integrity auditing. It is based on a growing list of blocks that are connected (chained) to each other. A block consists of a block header and auditing proofs. The block header includes a timestamp, the hash of the previous blocks and the Merkle root. The Merkle hash tree comprises all auditing proofs rather than transactions. In this way, all auditing proofs are stored in blocks and the integrity of them can be guaranteed by the Merkle root. Figure 3 shows an example of a DAB.

C. THE BASIC CONSTRUCTION

The construction of the proposed scheme is as follows:

1) **Representative Selection**: all nodes (data owners/users and the CSP) vote to select a master node and some slave nodes, which comprises the set of representative nodes. The number of representative nodes should be more than $3t$, where t is the maximum number of malicious nodes.
2) **Key Generation**: in this algorithm, G_1 and G_t denote two multiplicative cyclic groups with prime order q. Suppose that $e : G_1 \times G_1 \rightarrow G_T$ is a bilinear map and g is the generator of the group G_1. $h : (0, 1)^* \rightarrow G_1$ denotes the hash function that maps a string to a point. $h' : G_1 \rightarrow Z_q^*$ denotes another hash function. Suppose that f is a pseudo-random function (PRF) and π is a pseudo-random permutation (PRP). Each data owner generates a random secret key $x \in Z_q^*$ and computes the public key $Y = g^x$. It keeps the secret key x and publishes the parameters $\{Y, f, \pi, e, H, h, h'\}$.
3) **Tag Generation**: a data owner generates HVTs for all of its files. It splits a file into data blocks $F = \{b_1, b_2, \cdots, b_n\}$ and randomly chooses an element $u \in G_1$. It then calculates a HVT for each b_i through $\sigma_i = (h(ID_F||i) \cdot u^{b_i})^x$, where ID_F is an unique identifier of F.
4) **Challenge**: the data owner/user determines the amount of challenged data blocks z and randomly chooses two keys $k_1, k_2 \in Z_q^*$. It utilizes the secret key x to sign $(z, k_1, k_2) : \sigma_{CU} = Sig_x(z||k_1||k_2)$. It then sends a challenge request $C = (z, k_1, k_2, \sigma_{CU})$ to the CSP.
5) **Response**: the CSP first verifies the signature σ_{CU}. Then it generates a challenge set $Q = \{(i, v_i)\}$ by calculating $i = \pi_{k_1}(l)$ and $v_i = f_{k_2}(l)$ for $l \in [1, z]$, where i is the index of each challenged data block and v_i is the corresponding coefficient. The CSP calculates the aggregated HVT $\sigma = \prod_{(i,v_i \in Q)} \sigma_i^{v_i}$ and the combination of the challenged blocks $\mu' = \sum_{(i,v_i) \in Q} v_i \cdot b_i$. It computes $\bar{u} = u^r \in G_1$ with a random value $r \in Z_q^*$. It then computes $\alpha = h'(\bar{u})$ and $\mu = \alpha\mu' + r$ to blind μ'. The CSP signs $(\sigma, \mu, \bar{u})$ with its secret key $x' : \sigma_{CSP} = Sig_{x'}(\sigma||\mu||\bar{u})$. Finally, the CSP broadcasts an auditing proof $P = \{\sigma, \mu, \bar{u}, \sigma_{CSP}, C\}$ to all other nodes.
6) **Consensus**: upon receiving an auditing proof P, a non-representative node broadcasts it to all other nodes. A representative node will first verify the signature σ_{CSP}. If the verification holds, it collects the auditing proof P. The master node waits for time Δt to launch a consensus proposal $CP_m = (Proposal, h_b, blk, \sigma_m)$, where blk is the new block prepared for publishing, h_b is the hash value of blk, and σ_m is the signature of $(Proposal, h_b, blk)$. It broadcasts CP_m to all slave nodes. Upon receiving CP_m, a slave node will verify the signature. If the signature is

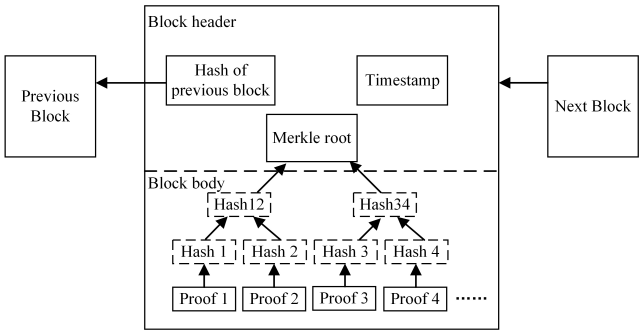

FIGURE 3. Data auditing blockchain.

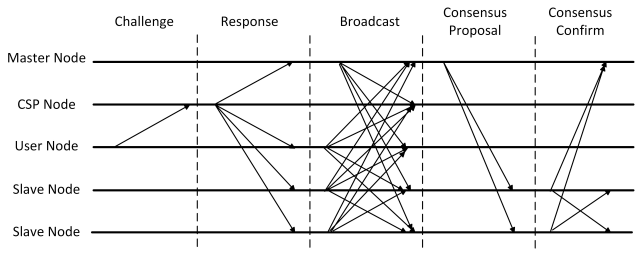

FIGURE 4. The flow from Challenge algorithm to Consensus algorithm.

valid, it broadcasts a consensus confirmation $CP_j = (Confirm, j, h_b, blk, \sigma_j)$, where j is the index of the slave node, σ_j is the signature of $(Confirm, j, h_b, blk)$. When a slave node receives more than $2t$ consensus confirmations from other slave nodes, it will accept the new data block and add it to the end of the blockchain.

7) **Verification**: The data owner or data user accesses an auditing proof from the DAB. It computes $\alpha = h'(\bar{u})$ and verifies the auditing proof by calculating the following equation:

$$e(\bar{u}, Y) \cdot e(\sigma^{\alpha}, g) = e((\prod_{(i,v_i)\in Q} h_i^{v_i})^{\alpha} \cdot u^{\mu}, Y) \quad (1)$$

where $h_i = h(ID_F||i)$ and Q is generated using the same way in Challenge algorithm. If the verification equation holds, the data owner's data is intact. Otherwise its data is corrupted.

Figure 4 shows the flow of the communication from the Challenge algorithm to the Consensus algorithm.

D. SUPPORT FOR BATCH AUDITING

All historical auditing proofs are stored in the DAB and hence they cannot be tampered with. Therefore, in our scheme, a data owner or user can verify multiple previous auditing proofs in the DAB simultaneously. This relieves them of online burdens and computational overheads associated with data integrity checking.

Consider that s challenge requests $\{C_j\}_{1\le j\le s}$ have been sent by a data owner to the CSP. Anyone can retrieve auditing proofs $\{P_j\}_{1\le j\le s}$ from the DAB. It then performs batch auditing by calculating the following equation:

$$e(\prod_{j=1}^{s} \bar{u}_j, Y) \cdot e(\prod_{j=1}^{s} \sigma_j^{\alpha_j}, g) = e(\prod_{j=1}^{s} (\prod_{(i,v_i)\in Q_j} h_i^{v_i})^{\alpha_j} \cdot u^{\sum_{j=1}^{s} \mu_j}, Y) \quad (2)$$

where $h_i = h(ID_F||i)$.

E. SUPPORT FOR DYNAMIC AUDITING

Data owners or users may access cloud data as well as updating this data. It is essential for a big data auditing scheme in the smart city context to support dynamic data. We extend the proposed scheme to support dynamic data based on a modified Merkle Hash Tree (mMHT) [30], which is stored and managed by the CSP.

In order to support dynamic operations, there are several steps that should be added to current auditing algorithms. In the Key Generation algorithm, a data owner or user constructs a mMHT and sends the root to the CSP. In the Tag Generation algorithm, the HVT should be calculated as: $\sigma_i = (h(ID_F) \cdot u^{b_i})^x$. The index verification of the data block is guaranteed by the mMHT. In the Response algorithm, the CSP adds the auxiliary authentication information (AAI) of the mMHT to the auditing proof P. In the Verification algorithm, the data owner or user verifies the AAI generated by the CSP and then performs the auditing through Equation 1.

V. SECURITY ANALYSIS

Theorem 1: If all entities are honest and the proposed scheme is executed correctly, the CSP can pass the verification.

Proof: We prove the correctness of our scheme as follows:

$$\begin{aligned} e(\bar{u}, Y) \cdot e(\sigma^{\alpha}, g) &= e(u^r, Y) \cdot e((\prod_{(i,v_i)\in Q} \sigma_i^{v_i})^{\alpha}, g) \\ &= e(u^r, Y) \cdot e((\prod_{(i,v_i)\in Q} (h_i \cdot u^{b_i})^{v_i})^{\alpha}, g) \\ &= e((\prod_{(i,v_i)\in Q} (h_i \cdot u^{b_i})^{v_i})^{\alpha} \cdot u^r, Y) \\ &= e(\prod_{(i,v_i)\in Q} h_i^{v_i\alpha} \prod_{(i,v_i)\in Q} u^{v_i b_i \alpha} \cdot u^r, Y) \\ &= e(\prod_{(i,v_i)\in Q} h_i^{v_i\alpha} \cdot u^{\alpha \sum_{(i,v_i)\in Q} v_i b_i + r}, Y) \\ &= e((\prod_{(i,v_i)\in Q} h_i^{v_i})^{\alpha} \cdot u^{\mu}, Y) \end{aligned} \quad (3)$$

where $h_i = h(ID_F||i)$.

Theorem 2: If the CSP successfully passes the Verification algorithm, it must indeed possess the intact data blocks.

Proof: We first prove that in the random oracle model, there exists an extractor of μ'. We then show that with a valid response $\{\sigma, \mu'\}$, the correctness of the theorem follows from [15].

The CSP is regarded as the adversary. The extractor responds the random oracle $h'(\cdot)$ queried by the CSP. Suppose that a challenge is requested by the extractor and the CSP outputs a valid response $\{\sigma, \mu, \bar{u}\}$. Thus, we can obtain:

$$e(\bar{u}, Y) \cdot e(\sigma^{\alpha}, g) = e((\prod_{(i,v_i)\in Q} h_i^{v_i})^{\alpha} \cdot u^{\mu}, Y) \quad (4)$$

We assume that the extractor can rewind the CSP to the point just before the random oracle $h'(\cdot)$ is given. Now the extractor sets the $h'(\bar{u})$ to be $\alpha^* \neq \alpha$. The CSP then outputs a new valid response $\{\sigma, \mu^*, \bar{u}\}$. Then we can obtain:

$$e(\bar{u}, Y) \cdot e(\sigma^{\alpha^*}, g) = e((\prod_{(i,v_i)\in Q} h_i^{v_i})^{\alpha^*} \cdot u^{\mu^*}, Y) \quad (5)$$

TABLE 1. Property comparison with existing schemes.

Scheme	Blockchain based	Decentralized auditing	Public auditing	Traceability
[2]	N	N	N	N
[7]	N	N	N	N
[18]	N	N	Y	N
[23]	N	N	Y	N
[4]	N	N	Y	N
[26]	N	N	Y	N
[8]	N	N	Y	N
[29]	N	N	N	N
Our scheme	Y	Y	Y	Y

Dividing Equation 4 by Equation 5, we have

$$
\begin{aligned}
e(\sigma^{\alpha-\alpha^*}, g) &= e((\prod_{(i,v_i)\in Q} h_i^{v_i})^{\alpha-\alpha^*} \cdot u^{\mu-\mu^*}, Y) \\
e(\sigma^{\alpha-\alpha^*}, g) &= e((\prod_{(i,v_i)\in Q} h_i^{v_i})^{\alpha-\alpha^*} \cdot u^{\mu-\mu^*}, g^x) \\
e(\sigma^{\alpha-\alpha^*}, g) &= e((\prod_{(i,v_i)\in Q} h_i^{v_i})^{x(\alpha-\alpha^*)} \cdot u^{x(\mu-\mu^*)}, g) \\
\sigma^{\alpha-\alpha^*} &= (\prod_{(i,v_i)\in Q} h_i^{v_i})^{x(\alpha-\alpha^*)} \cdot u^{x(\mu-\mu^*)} \\
(\prod_{(i,v_i)\in Q} \sigma_i^{v_i})^{x(\alpha-\alpha^*)} &= (\prod_{(i,v_i)\in Q} h_i^{v_i})^{x(\alpha-\alpha^*)} \cdot u^{x(\mu-\mu^*)} \\
(\prod_{(i,v_i)\in Q} \sigma_i^{v_i})^{\alpha-\alpha^*} &= (\prod_{(i,v_i)\in Q} h_i^{v_i})^{\alpha-\alpha^*} \cdot u^{\mu-\mu^*} \\
u^{\mu-\mu^*} &= (\prod_{(i,v_i)\in Q} \frac{\sigma_i}{h_i})^{v_i(\alpha-\alpha^*)} \\
u^{\mu-\mu^*} &= (\prod_{(i,v_i)\in Q} \frac{h_i \cdot u^{bi}}{h_i})^{v_i(\alpha-\alpha^*)} \\
u^{\mu-\mu^*} &= (\prod_{(i,v_i)\in Q} u^{b_i})^{v_i(\alpha-\alpha^*)} \\
u^{\mu-\mu^*} &= u^{(\sum_{(i,v_i)\in Q} v_i b_i)\cdot(\alpha-\alpha^*)} \\
\mu-\mu^* &= (\sum_{(i,v_i)\in Q} v_i b_i)\cdot(\alpha-\alpha^*) \\
(\sum_{(i,v_i)\in Q} v_i b_i) &= \frac{\mu-\mu^*}{\alpha-\alpha^*} \qquad (6)
\end{aligned}
$$

Therefore, the extractor obtains $\{\sigma, \mu' = \frac{\mu-\mu^*}{\alpha-\alpha^*}\}$ as a valid response of the auditing proof. Note that the random oracle model and the extraction argument are also used in the proof of [15] and [18].

Theorem 3: In the proposed scheme, a data owner or data user cannot retrieve any file content from auditing proofs stored in the DAB.

Proof: A data owner or data user can access another data owner's auditing proofs from the DAB. Specifically, it can collect a challenge request $C = (z, k_1, k_2, \sigma_{CU})$, an aggregated HVT σ, a combination of challenged data blocks μ, a verification parameter $\bar{u}$, and a signature of the CSP from an associated auditing proof P. The challenged data blocks are hidden in σ and μ. Leveraging the similar security analysis of [18], the HVT σ is secure based on the CDH problem. Thus, the privacy of the proposed scheme is reduced to the privacy of μ. Due to the introduction of the random value in μ, the proposed scheme can resist recovery attacks as mentioned in [18] and [21]. Therefore, our scheme can preserve the privacy of owner's data.

VI. EVALUATION AND ANALYSIS

In this section, we evaluate the performance of our scheme from two aspects, namely property comparison and the comparison of the computational cost.

Table 1 shows the property comparison of the state of the art against the proposed scheme. As seen the proposed scheme supports all features compared over existing schemes. The proposed scheme is more stable and reliable because of the decentralized architecture. Thus, if some nodes are compromised, the scheme is still available. The proposed scheme also supports public auditing, which means auditing proofs can be verified by any data owner or user at any time. Moreover, all auditing proofs in the scheme are traceable since all historic auditing proofs are stored in the DAB permanently and cannot be tampered with.

The computational cost of the data owner/user and the CSP in the proposed scheme during auditing is shown in Table 2. Let M be the multiplication operation on the group. Let E be the exponentiation operation on the group. Let P be the bilinear pairing operation. H denotes the hash function mapping a string to a point on the group. The computational overheads are mainly in the Tag Generation, Response and Verification algorithms. In the Tag Generation algorithm, a data owner generates HVTs for all data blocks. Thus the computation cost is $n(M + 2E + H)$. In the Response algorithm, the CSP performs $(z + 1)$ exponentiation and $(z - 1)$ multiplications to calculate the aggregated HVT σ and $\bar{u}$. In the Verification algorithm, the CSP performs three pairings, $(z + 3)$ exponentiation, $(z + 1)$ multiplications and z map-to-point hash functions to verify the equation.

We validate the efficiency and effectiveness of our scheme by conducting experiments using JAVA JDK 9.0.4 and all algorithms are implemented using the Java Pairing-Based Cryptography Library (JPBC) library version 2.0.0. The experiments utilize type A pairing parameters in which the group order is 160 bits and the base field order is 512 bits.

TABLE 2. Computation cost of the proposed scheme.

	Tag Generation	Response	Verification
Owner/User	$n(M+2E+H)$	–	$3P+(z+3)E+(z+1)M+zH$
CSP	–	$(z+1)E+(z-1)M$	–

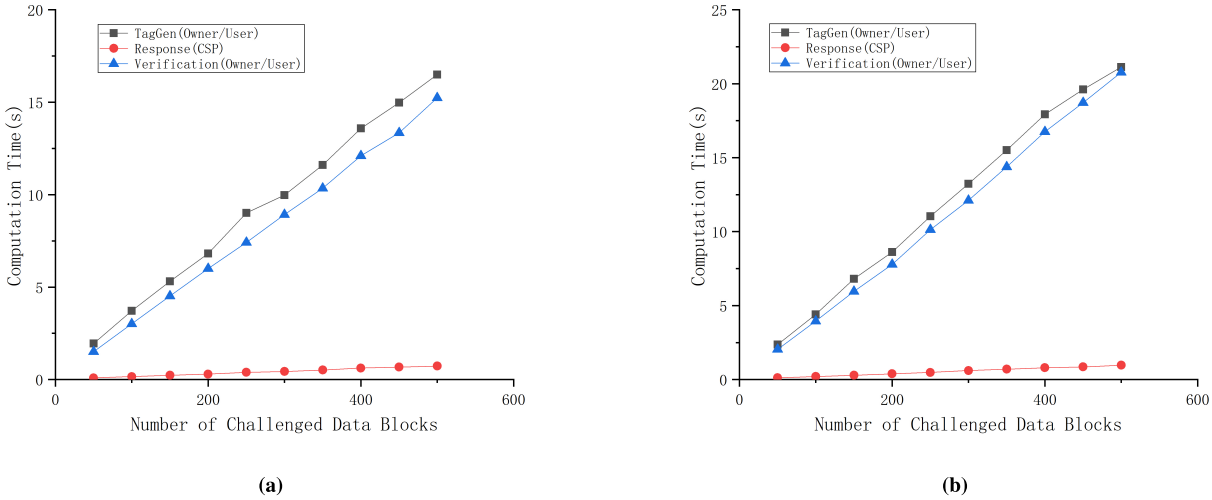

FIGURE 5. Performance of the proposed scheme on different platforms. (a) Server platform. (b) PC laptop platform.

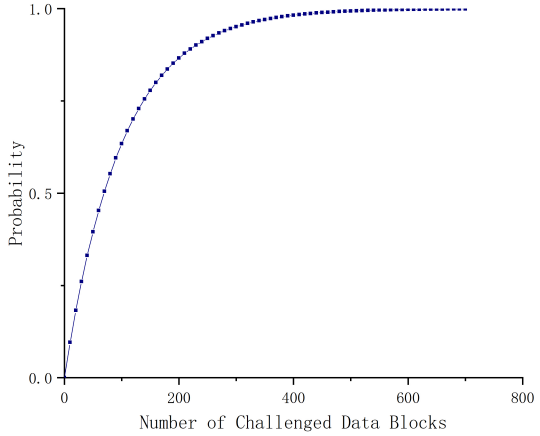

FIGURE 6. Detection probability of the proposed scheme.

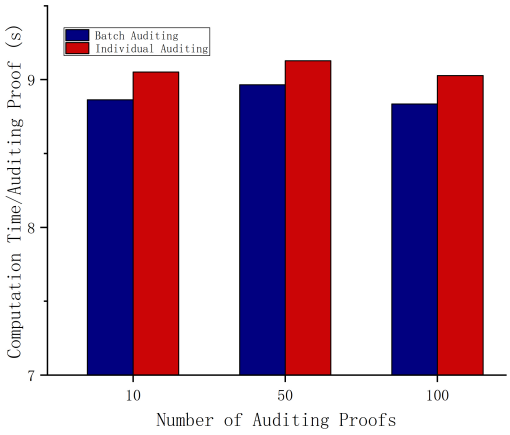

FIGURE 7. Comparison of the computation cost between individual auditing and batch auditing.

Figure 5 shows the computation time of the data owner/user and the CSP in our scheme on both the server and PC laptop platform. The server runs Windows 10 on an Intel Xeon E5 CPU at 3.7GHz and 32GB RAM. The PC laptop runs Windows 10 on an Intel Core i5 CPU at 2.30 GHz and 4GB DDR3 RAM. The performance of two platforms have a similar pattern, although the overall computation time of the proposed scheme on the PC laptop platform is slightly larger than on the server platform.

It is easy to see that the computation time of the CSP is very little. This is because there exists no calculation of expensive map-to-point hash functions in this algorithm. Although the computation time of the Tag Generation algorithm grows linearly, the efficiency of generating HVTs is acceptable since HVTs are only generated once in the whole life time of data storage and auditing, which may last for many years. The computation time of the Verification algorithm is limited. Although the computation cost of the Verification algorithm grows linearly with the number of challenged data blocks, only a limited number of data blocks need to be challenged for achieving very high detection probability. This is due to the adoption of the sampling strategy [2]. The relationship between the number of challenged data blocks and detection probability is shown in Figure 6. We can see that the proposed scheme can guarantee more than 95% detection probability by only checking 300 data blocks. Meanwhile, in our scheme, checking 300 data blocks only takes 8.9 seconds and 12.1 seconds on the server and PC laptop platform respectively. Thus,

the computation cost of the Verification algorithm is acceptable. Besides, the performance of the Verification algorithm can be further improved by pre-computing the expensive map-to-point hash function.

Figure 7 illustrates the comparison of computation time between batch auditing and individual auditing on the server platform. As seen, batch auditing can effectively reduce the computation time in the Verification algorithm. This is because by aggregating multiple verification equations into one equation, the computation cost of expensive pairing operations can be saved.

VII. CONCLUSIONS

Nowadays, smart cities demand big data collection and the ability to analyze big data using intelligent algorithms such as machine learning. These depend greatly on the quality and the reliability of data. In this paper, we investigate big data integrity auditing in cloud environments to meet the challenges of secure smart city infrastructures. A blockchain-based remote data integrity auditing scheme for the cloud has been proposed. We design a decentralized auditing architecture and a novel blockchain instantiation named the Data Auditing Blockchain (DAB) to improve the stability and reliability of the whole scheme. By introducing the DAB, the proposed scheme can trace all of the auditing history and allow all data files to be verified by any data owner or user at any time. The proposed scheme is further extended to support batch verification of multiple auditing proofs and dynamic auditing. The security analysis demonstrates that the proposed scheme is both secure and privacy-preserving. The performance evaluation shows that our scheme is also efficient compared with the state of the art.

REFERENCES

[1] *The Lambda Network. Report*, SkyLab, Inc., Atlanta, Georgia, 2018.

[2] G. Ateniese *et al.*, "Provable data possession at untrusted stores," in *Proc. 14th Conf. Comput. Commun. Secur.*, 2007, pp. 598–609.

[3] G. Ateniese, P. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw.*, New York, NY, USA, 2008, Art. no. 9.

[4] A. F. Barsoum and M. A. Hasan, "Provable multicopy dynamic data possession in cloud computing systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 3, pp. 485–497, Mar. 2015.

[5] W. L. Chang, "NIST big data interoperability framework: Reference architecture," NIST Big Data Public Work. Group, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. (NIST SP)-1500-6r1, 2015.

[6] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-replica provable data possession," in *Proc. 28th Int. Conf. Distrib. Comput. Syst.*, Jun. 2008, pp. 411–420.

[7] C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proc. Conf. Comput. Commun. Secur.*, 2009, pp. 213–222.

[8] H. Jin, K. Zhou, H. Jiang, D. Lei, R. Wei, and C. Li, "Full integrity and freshness for cloud data," *Future Gener. Comput. Syst.*, vol. 80, pp. 640–652, Mar. 2018.

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

[10] Z. Ma, Y. Lai, W. B. Kleijn, Y.-Z. Song, L. Wang, and J. Guo, "Variational Bayesian learning for Dirichlet process mixture of inverted Dirichlet distributions in non-Gaussian image feature modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2018.2844399.

[11] Z. Ma and A. Leijon, "Bayesian estimation of beta mixture models with variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2160–2173, Nov. 2011.

[12] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, "Variational Bayesian matrix factorization for bounded support data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 4, pp. 876–889, Apr. 2015.

[13] Z. Ma, H. Yu, W. Chen, and J. Guo, "Short utterance based speech language identification in intelligent vehicles with time-scale modifications and deep bottleneck features," *IEEE Trans. Veh. Technol.*, to be published, doi: 10.1109/TVT.2018.2879361.

[14] N. Satoshi, "Bitcoin a peer-to-peer electronic cash system," Bitcoin.org, White Paper, 2008.

[15] H. Shacham and B. Waters, "Compact proofs of retrievability," *J. Cryptol.*, vol. 26, no. 3, pp. 442–483, 2013.

[16] W. T. Shen, G. Yang, J. Yu, H. L. Zhang, F. Y. Kong, and R. Hao, "Remote data possession checking with privacy-preserving authenticators for cloud storage," *Future Gener. Comput. Syst.*, vol. 76, pp. 136–145, Nov. 2017.

[17] D. Vorick and L. Champine, "Sia: Simple decentralized storage," Nebulous Labs, Nebulous Inc., Boston, MA, USA, White Paper, 2018.

[18] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.

[19] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Proc. 14th Eur. Conf. Res. Comput. Secur.*, 2009, pp. 355–370.

[20] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.

[21] Z. Xu, L. Wu, D. He, and M. K. Khan, "Security analysis of a publicly verifiable data possession scheme for remote storage," *J. Supercomput.*, vol. 73, no. 11, pp. 4923–4930, 2017.

[22] L. Xue, J. Ni, Y. Li, and J. Shen, "Provable data transfer from provable data possession and deletion in cloud storage," *Comput. Standards Interfaces*, vol. 54, pp. 46–54, Nov. 2017.

[23] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1717–1726, Sep. 2013.

[24] Z. Yang, J. Lei, K. Fan, and Y. Lai, "Keyword extraction by entropy difference between the intrinsic and extrinsic mode," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 19, pp. 4523–4531, Oct. 2013.

[25] Z. Yang, J. Tang, and H. Liu, "Cloud information retrieval: Model description and scheme design," *IEEE Access*, vol. 6, pp. 15420–15430, 2018.

[26] H. Y. Yu, Y. Cai, S. Kong, Z. Ning, F. Xue, and H. Zhong, "Efficient and secure identity-based public auditing for dynamic outsourced data with proxy," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 10, pp. 5019–5041, 2017.

[27] Y. Yuan and F.-Y. Wang, "Blockchain: The state of the art and future trends," *Acta Automat. Sinica*, vol. 42, no. 4, pp. 481–494, 2016.

[28] Z. Faheem *et al.*, "A survey of cloud computing data integrity schemes: Design challenges, taxonomy and future trends," *Comput. Secur.*, vol. 65, pp. 29–49, Mar. 2017.

[29] L. Zang, Y. Yu, L. Xue, Y. Li, Y. Ding, and X. Tao, "Improved dynamic remote data auditing protocol for smart city security," *Pers. Ubiquitous Comput.*, vol. 21, no. 5, pp. 911–921, 2017.

[30] Y. Zhang, J. Ni, X. Tao, Y. Wang, and Y. Yu, "Provable multiple replication data possession with full dynamics for secure cloud storage," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 4, pp. 1161–1173, 2016.

[31] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2231–2244, Dec. 2012.

HAIYANG YU was born in Lixin, Anhui, China, in 1991. He received the B.S. degree in computer science and technology from the Beijing University of Technology, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree. He is also pursuing the joint Ph.D. degree with the Faculty of Engineering, The University of Melbourne. His research interests include cloud storage auditing, data integrity checking, and data mining.

ZHEN YANG received the Ph.D. degree in signal processing from the Beijing University of Posts and Telecommunications. He is currently a Full Professor of computer science and engineering with the Beijing University of Technology. He has published more than 30 papers in highly ranked journals and top conference proceedings. His research interests include data mining, machine learning, trusted computing, and content security. He is a Senior Member of the Chinese Institute of Electronics.

RICHARD O. SINNOTT received the B.S. degree in theoretical physics from the University of East Anglia, U.K., in 1988, and the M.S. degree in software engineering and the Ph.D. degree from the University of Stirling, Stirling, Scotland, in 1993, and 1997, respectively. Since 2010, he has been the Director of eResearch with The University of Melbourne and a Professor of applied computing systems. His research interests include IT security, big data management, distributed systems, and especially cloud-based infrastructures.