IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 18, 2022, accepted April 4, 2022, date of publication April 11, 2022, date of current version April 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3166536

Data Integrity Audit Based on Data Blinding for Cloud and Fog Environment

GENQING BIAN[1], YANRU FU[1], BILIN SHAO[2], AND FAN ZHANG[1], (Member, IEEE)
[1]School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an, Shaanxi 710055, China
[2]School of Management, Xi'an University of Architecture and Technology, Xi'an, Shaanxi 710055, China

Corresponding author: Yanru Fu (fyr@xauat.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872284, in part by the Shaanxi Provincial Natural Science Basic Research Project under Grant 2021JLM-16, and in part by the Scientific Research Staring Foundation for the Returned Overseas Chinese Scholars under Grant 1685.

ABSTRACT Cloud-fog computing is a novel computing model that expands the functionality of cloud computing, which provides various services through fog nodes. The issue of traditional data integrity auditing are low data security, slow data processing speed and low communication efficiency. To solve these problems, this paper proposes a data integrity audit scheme based on data blinding. This scheme uses the edge devices in the transmission node to establish a fog computing layer between the cloud service provider and the data owner to reduce transmission delay. The subordinate distribution relationship and weight between fog nodes dynamically allocate the optimal path and transmit the data to reduce transmission delay. At the same time, a blind factor is added to the integrity audit in the evidence generation process to avoid data leakage. This paper gives a security model and security proof based on computational Diffie-Hellman (CDH) assumptions. The experimental results show that the fog computing layer and blind factor are introduced into the data integrity audit process, which can reduce the data communication delay effectively and improve the security of data audit.

INDEX TERMS Cloud and fog computing, data blinding, integrity audit, cloud storage.

I. INTRODUCTION

In recent years, as the abundance of information has grown, the storage and computing requirements on mobile phones, computers, and other terminal devices have increased. To reduce the storage pressure on terminal devices, some users store their data in the cloud [1]. However, some cloud service providers could delete some infrequently used data to reduce server overhead. Deleted data may not be retrieved, resulting in cloud data loss. As users upload data, the data is stored on the cloud server instead of the local device [2]. Remotely checking the integrity of the data uploaded by users has become an urgent problem.

In response to the above problems, the concept of Remote Data Possession Checking (RDPC) is proposed, which includes proof of retrievability (POR) and provable data procession (PDP) [3]–[5]. However, from the perspective of data audit, it can be divided into private and public audits. The auditor of the private audit is the data owner, while the auditor of the public audit can be any authorized third-party audit. Due to the higher flexibility of public auditing methods, most of them will choose public auditing [6].

As the internet has found its way into people's lives, cloud computing enjoys rising popularity among individuals of all stripes. More and more users store their data in the cloud for easy use anytime, anywhere. However, in the traditional cloud storage model, the cloud service provider needs to establish a connection with each user, which invisibly increases the load pressure on the cloud service provider [7]. Therefore, how to reduce the computing and load pressure of cloud service providers has become an urgent problem to be solved.

In the context of data integrity audits, cloud servers are usually far away from the user end [8]. Long-distance data transmission would occupy network bandwidth and increase transmission delay [9]. To solve this problem, the concept of fog computing is proposed [10]. Fog computing expands the concept of cloud computing. Compared with cloud computing, it is closer to the data owner. In data transmission, the fog node layer is added to reduce the delay and bandwidth [11], [12]. Hu *et al.* [13] proposed a security and privacy protection scheme based on the fog computing framework, which did not consider the data transmission

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

model in the fog computing framework. For the remote data ownership audit scheme proposed by Yan *et al.* [14], the document label aggregation scheme in the evidence generation stage of the scheme refers to the information and coefficients of the included files. Malicious attackers can use the disclosed coefficients to calculate information by requesting file labels multiple times, resulting in information leakage. Therefore, this paper introduces a blinding factor, adds random coefficients in the evidence generation process, and discloses the public key of the random coefficients to ensure the security of the evidence generation stage. At the same time, in order to reduce the transmission delay, the data transmission model in the cloud-mist network is given and a data integrity audit scheme based on the cloud-mist architecture is proposed.

A. RELATED WORKS

In 2007, Ateniese *et al.* [15] proposed a PDP model, which allows a client storing data on an untrusted server to verify whether the server with the original data. Subsequently, Juels *et al.* [16] defined a POR model, which can generate concise proof that the user can retrieve the target file by archiving or backing up large files and allows the user to restore the entire file data. In 2008, Ateniese [17] constructed a provably secure PDP technology based entirely on symmetric key encryption and effectively supports block modification, deletion, and append operations. Shacham and Waters [18] proposed the first retrievability proof scheme, which allows anyone to act as a verifier, not just the file owner, and proposed a scheme that only allows private verification. Both schemes rely on the same state attribute aggregates the proof into a validator value. Wang [19] studied proxy provable data possession (PPDP) when the client cannot perform remote data possession checks in the public cloud. Ren *et al.* [20] proposed the designated verifier provable data possession (DV-PDP) when the client cannot perform remote data possession inspection. Yan *et al.* [14] propose a new RDPC scheme with a designated validator, in which the data owner designates a unique validator to check data integrity.

Cisco proposed the concept of fog computing in 2014. In this model, data and its processing are concentrated in devices at the edge of the network. Subsequently, Mohammed *et al.* [21] proposed an authentication protocol in the fog computing environment to ensure data integrity. Alzubi *et al.* [22] proposed a novel chaotic map image secret writing formula, which applied the security of enhancing the metric of cryptosystems to pixel-level and bit level permutations. Tian and Wang [23] proposed a data audit scheme based on the Internet of Things (IoT) and cloud-fog computing, in which the private key is separated into the fog center and held by the user. Then proposed a two-time signature method, which divides the signature process into two stages: original signature and final signature. Gu [24] introduced a secure data query framework for cloud and fog computing. When the fog network provides query data to users, cloud services are used to check the query data from the fog network. At the same time, Xu S *et al.* [25] introduced a cloud-fog-device data sharing system with data confidentiality and data source identification based on matching attribute encryption primitives (MABE) through extended matching encryption. Alzubi *et al.* [26] designed a robust cryptosystem that is based on Hermite curves and is more suitable for IoT devices with limited processing and storage power. In the same year, Alzubi *et al.* [27] proposed Hashed Needham Schroeder Cost Optimized Deep Machine Learning (HNS-CODML) method, which improves the security of data sent from the cloud. Noura *et al.* [28] proposed a new encryption solution to protect data in fog computing, which provides data confidentiality, integrity and availability, and source authentication.

B. MOTIVATION AND CONTRIBUTION

This paper proposes a data integrity audit scheme based on the cloud and fog architecture, meanwhile, provides a data transmission model in the cloud and fog network. In this model, the data is transmitted and calculated by fog nodes to find the lowest communication channel, thereby reducing communication overhead. At the same time, a blind factor is introduced in the evidence generation stage of the integrity audit to prevent the adversary from calculating the ciphertext in the two interrogations and improve the security of the integrity audit.

The main contributions of this paper are as follows.

1) This paper proposes a data integrity audit model in a cloud and fog environment, which can effectively reduce the communication overhead in the transmission process and reduce the computing pressure of the cloud service provider.
2) In the data integrity audit, a blind factor is introduced to avoid data leakage caused by repeated submissions of malicious auditors when challenging data.
3) Under the given security model, this article proved the security of this scheme. Experimental results show that this scheme has better performance and feasibility.

C. OUTLINE

The second section introduces the preliminary work of our proposed scheme. The third section defines the specific structure of the data blinding for cloud and fog (DBCF) system model and main steps. Section IV displays the safety analysis of DBCF. In the fifth section, the paper presents performance analysis, which includes theoretical complexity analysis and experimental performance. Section VI concludes the article.

II. PRELIMINARIES

A. NOTATIONS

Let k be a safety parameter, and q is a large prime number which's order is k. $\mathbb{G}_1$ and $\mathbb{G}_2$ are multiplicative cyclic groups, and their order is k. g is the generator of $\mathbb{G}_1$, and u is a random element of the multiplicative cyclic group. e is the bilinear mapping $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, H is a secure

TABLE 1. Frequently used notations.

Notations	Description
k	a security parameter
q	a large prime
$\mathbb{G}_1, \mathbb{G}_2$	the multiplicative cyclic groups of order q
g	a generator of multiplicative cyclic group $\mathbb{G}_1$
u	a random group element of $\mathbb{G}_1$
H	a secure hash function $\{0,1\}^* \rightarrow \mathbb{G}_1$
x, r	$x, r \in Z_q^*$
e	$\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
ϕ	$Z_q^* \times \{1, \cdots, n\} \rightarrow \{1, \cdots, n\}$
φ	$Z_q^* \times Z_q^* \rightarrow Z_q^*$
fn_m	a fog node device
m	the number of fog node devices
$D = (V, E)$	the undirected graph with vertex set V and edge set E
w_{fn_i,fn_j}	the delay between nodes $\{fn_i, fn_j\}$
com_{fn_i,fn_j}	the communication delay between nodes $\{fn_i, fn_j\}$
pro_{fn_i,fn_j}	the processing delay between nodes $\{fn_i, fn_j\}$
que_{fn_i,fn_j}	the queuing delay between nodes $\{fn_i, fn_j\}$
$tran_{fn_i}$	the transmission speed of each fog node device fn_i
z_i	the divided sub-transmission data
$dist_{fn_i,fn_j}$	the relationship between $\{fn_i, fn_j\}$

hash functions, and ϕ, φ are pseudo-random permutation and pseudo-random function. Besides, some frequently used notations are given in Table 1.

B. BILINEAR MAPS

Specify that the multiplicative cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ have the same prime order q, g is a generator of $\mathbb{G}_1$. e is the mapping of $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, which has the following properties:

1) Bilinearity: for $\forall u, v \in \mathbb{G}_1$, $\forall a, b \in Z_q^*$, there is an equation $e(u^a, v^b) = e(u, v)^{ab}$.
2) Non-Degeneracy: $\exists u, v \in \mathbb{G}_1$ such that $e(u, v) \neq 1_{\mathbb{G}_2}$, here $1_{\mathbb{G}_2}$ represents the identity element of the $\mathbb{G}_2$ group.
3) Computability: for $\forall u, v \in \mathbb{G}_1$, there is an algorithm that calculates the mapping $e(u, v)$.

C. CDH ASSUMPTION

The CDH assumption is a standard cryptographic hypothesis, and many cryptographic schemes are constructed on this CDH assumption, such as public-key encryption, digital signature, and authentication key exchange [29]. Moreover, complex agreements, such as cloud storage, refusing authentication agreements are also built on this assumption. Specifically, the CDH assumption on a cyclic group $\mathbb{G}$ with generator g refers to that it is hard to compute g^{ab} for any polynomial-time adversary $\mathcal{A}$ when given the items g, g^a, and g^b, which can be defined as:

$$ADV^{CDH}_{\mathbb{G}_1,\mathcal{A}} = Pr[\mathcal{A}(g, g^a, g^b) = g^{ab} : a, b \overset{R}{\leftarrow} Z_q^*] \leq \varepsilon \quad (1)$$

D. SYSTEM MODEL

The data integrity audit model based on data blinding in the cloud and fog environment includes four entities: data owners, fog computing nodes, cloud service providers, and third-party auditors. Figure 1 presents the system model of our proposed DCBF model.

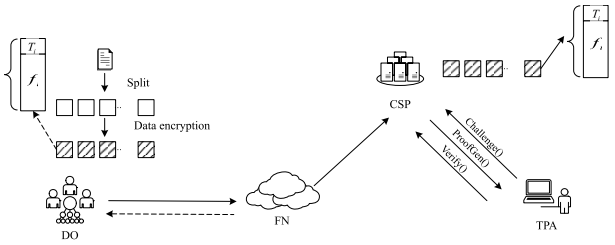

FIGURE 1. DCBF system model.

1) The data owner (DO) rents cloud storage services and uploads a large amount of data to the cloud storage server to achieve the purpose of storing data remotely and accessing it at any time. The data owner can be an individual consumer or an organization's consumer.
2) Fog computing nodes (FN) are interconnected edge devices with precise computing capabilities, such as gateways, switches and routers. In this model, the data is preprocessed and transmitted through the fog computing node, thereby reducing the computing and communication pressure of the cloud service provider.
3) Cloud service providers (CSP) have massive storage capacity and robust computing power. Cloud service providers receive data uploaded by users through fog nodes, provide cloud storage and computing services to data owners, and return data integrity certificates to third-party auditors after receiving data challenges. In particular, the cloud service provider divides users into blocks and stores tagged data. When proofs are needed, they only need to aggregate and generate proofs through tags [30].
4) The third-party auditor (TPA) will review the integrity of the outsourced data for the data owner. And TPA is trusted by the data owner and the cloud storage server. The third-party auditor will send the audit results to the data owner in the subsequent data integrity audit process.

The DBCF model includes the five polynomial time algorithms.

1) $Setup(1^k) \rightarrow (sk, pk)$: This algorithm is used to initialize the system and generates the user's public and private key pair. Enters the security parameter k, and output the corresponding public key and private key.
2) $TagGen(F, x) \rightarrow T$: The data owner executes this algorithm to generate the tag set of the uploaded file, and the data owner uploads the tag set and data block to the cloud accordingly.
3) $Challenge(cb) \rightarrow chal$: This algorithm is executed by a third-party auditor, inputs the number of blocks to be challenged, and outputs challenge information to the cloud service provider.
4) $ProofGen(F, T, chal) \rightarrow P$: This algorithm is executed by the cloud service provider and generates evidence. According to the challenge information, read

the files stored in the cloud and the corresponding tag information to calculate the evidence and return it to the third-party auditor.

5) $Verify(X, chal, P) \rightarrow \{0, 1\}$: The third-party auditor executes this algorithm and judges whether the data is entirely based on the evidence returned by the cloud service provider. If it is completed, outputs 1 to indicate.

E. SECURITY MODEL

In this subsection, the security model of DBCF is defined. This scheme is characterized by indistinguishability under chosen-plaintext attack (IND-CPA) game plaintext attack in the random oracle model [31]. The specific steps are as follows.

1) Initialization. Challenger $\mathcal{B}$ generates the system environment and initializes public parameters, and the adversary (denoted as $\mathcal{A}$) obtains these parameters.
2) Query. The adversary $\mathcal{A}$ can make the following query in the bounded order of the polynomial.
   a) H-Query: Challenger $\mathcal{B}$ establishes a hash query table to record and answer the adversary's hash query.
   b) Tag-Query: Adversary $\mathcal{A}$ submits file information to challenger $\mathcal{B}$, and the challenger runs the following formula and returns the result to adversary $\mathcal{A}$.

$$TagGen(F, x) \rightarrow T$$

   c) Verify-Query: The audit query is based on the tag query in the previous step. Challenger $\mathcal{B}$ runs $Challenge(cb) \rightarrow chal$ and sends the challenge block information $chal$ to adversary. The adversary $\mathcal{A}$ calculates the evidence P by running $ProofGen(F, T, chal) \rightarrow P$. Then, the adversary $\mathcal{A}$ returns the result. Challenger $\mathcal{A}$ calculates $Verify(X, chal, P) \rightarrow \{0, 1\}$ after receiving evidence P, and the final result will be returned to adversary $\mathcal{A}$.
3) Final phase. At this stage, challenger $\mathcal{B}$ submits challenge information $chal^*$ to adversary $\mathcal{A}$, then adversary $\mathcal{A}$ returns evidence P^*.

If $Verify(X, chal, P) \rightarrow 1$, the following conditions hold.

1) If the challenge information $chal/chal^*$ is submitted, the challenge file block has previously calculated the tag T.
2) The returned evidence P^* is not equal to P, and P^* will be calculated by $ProofGen(F, T, chal^*) \rightarrow P*$.

III. OUR PROPOSED DBCF MODEL

A. CLOUD AND FOG COMPUTING MODEL

The cloud and fog computing model in the DBCF model can be composed of a cloud service layer and a fog computing layer. The fog computing layer contains m fog node devices $(fn_1, fn_2, \cdots, fn_m)$, and its network structure is shown in Figure 2.

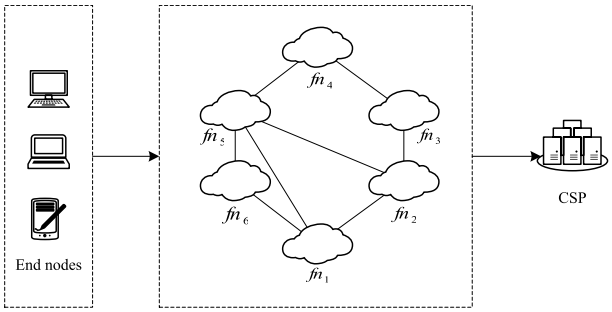

FIGURE 2. Network structure.

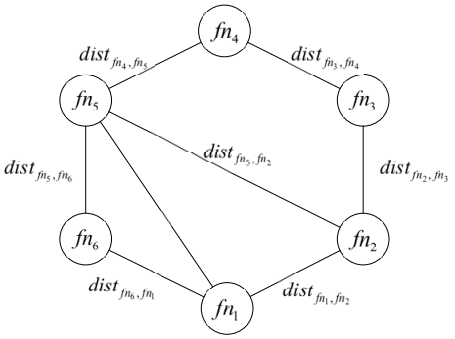

FIGURE 3. Weighted undirected graph.

According to the above figure, it can be abstracted as a weighted undirected graph $D = (V, E)$, V is a set of vertices in the graph D, representing the fog node device, and E is a set of edges represents the communication link between nodes. w_{fn_i,fn_j} represents the delay between nodes $\{fn_i, fn_j\}$, including communication delay, processing delay and queuing delay. The weighted undirected graph is shown in Figure 3.

Assuming that the transmission speed of each fog node device fn_i is $tran_{fn_i}$, during the data transmission process, the data owner divides the transmission data Z into $z_i = \lambda_i Z$, and z_i represents the divided sub-transmission data. The transmission time of the entire transmission data Z at the fog computing layer can be expressed as:

$$t(\lambda_i) = \max\left\{\frac{\lambda_i Z}{tran_{fn_i}} + w_{fn_i,fn_j} dist_{fn_i,fn_j}\right\} \tag{2}$$

$$w_{fn_i,fn_j} = com_{fn_i,fn_j} + pro_{fn_i,fn_j} + que_{fn_i,fn_j} \tag{3}$$

Among them, $\lambda_i Z/tran_{fn_i}$ indicates the time for the fog node to process the subtask z_i, $w_{fn_i,fn_j} dist_{fn_i,fn_j}$ indicates the delay between $\{fn_i, fn_j\}$, $dist_{fn_i,fn_j}$ indicates whether there is a subordinate allocation relationship between $\{fn_i, fn_j\}$, and $dist_{fn_i,fn_j} = 1$ indicates an allocation relationship existing, and vice versa.

Since the total transmission time in the fog calculation is equal to the most extensive transmission delay among all transmission times, in order to achieve the minimum delay, a set of optimal λ_i is required to minimize the objective function. The fog node calculation optimization model can be established as follows:

$$\min\left\{\max\left[\frac{\lambda_i Z}{tran_{fn_i}} + w_{fn_i,fn_j} dist_{fn_i,fn_j}\right]\right\}, i, j \in [1, m]$$

$$s.t.\ dist_{fn_i,fn_j} = \begin{cases} 1, & \lambda_i \neq 0 \\ 0, & \lambda_i = 0 \end{cases}, \sum_{i=1}^{m} \lambda_i = 1 \tag{4}$$

The task processed on each fog node is $z_i = \lambda_i Z$, then the task to be processed on each node can be constructed into a m dimensional vector $\mathbf{z} = [z_1, z_2, \cdots, z_m]^T$. Then the total time from node fn_r to transmit data Z' at the fog computing layer can be expressed as:

$$t(\mathbf{z}) = \max \left\{ \begin{array}{l} \dfrac{z_1}{tran_{fn_1}} + w_{fn_r,fn_1} dist_{fn_r,fn_1}, \\ \cdots, \\ \dfrac{z_m}{tran_{fn_m}} + w_{fn_r,fn_m} dist_{fn_r,fn_m} \end{array} \right\} \tag{5}$$

Therefore, in the search space $\Gamma = \prod_{i=1}^{m}[Z_{min}, Z_{max}] = \prod_{i=1}^{m}[0, Z]$, Z_{min} and Z_{max} represent the maximum and minimum values that the subtask z_i can take. Solving the corresponding transmission task z_i of each node on the fog node can be transformed into the following optimization problem:

$$\begin{aligned} \mathbf{z} &= \arg\min_{\mathbf{z}\in\Gamma}\{t(\mathbf{z})\} \\ s.t.\ z_i &\geq 0, \sum_{i=1}^{m} z_i = Z \end{aligned} \tag{6}$$

B. MAIN STEPS OF INTEGRITY AUDIT

This section gives a data integrity audit model based on data blinding. This model prevents anyone other than the data owner from knowing the original data. First, given the security parameter k, randomly select a large prime number q, where the order of q is k. $\mathbb{G}_1$ and $\mathbb{G}_2$ are two multiplicative cyclic groups. The length of the groups are q, and g is the generator, u is the random group element of $\mathbb{G}_1$. e is a bilinear mapping $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, H is a safe hash function. ϕ and φ are respectively a pseudo-random function and a pseudo-random permutation. Public parameters are $(q, g, u, \mathbb{G}_1, \mathbb{G}_2, e, H, \phi, \varphi)$.

Setup$(1^k) \rightarrow (sk, pk)$: The data owner randomly selects a number x as the private key, where $x \in Z_q^*$. Calculate $X = g^x$, and the data owner publishes X as the public key.

TagGen$(F, x) \rightarrow T$: First, before uploading the file F, the data owner divides the file F into n small pieces, denoted as $F = (f_1, f_2, \cdots, f_n)$. The data owner calculates the label T_i for each small file, and the calculation label equation is:

$$T_i = (H(F_{id}\|i) \cdot u^{f_i})^x \tag{7}$$

Among the equation, F_{id} represents a specific file identifier. Finally, the data owner calculates the tag set T of the file F, in which $T = (T_1, T_2, \ldots, T_n)$. Then, uploads the pairs $\{(T_i, f_i | i \in [i, n]\})$ to the cloud service provider (CSP).

Challenge$(cb) \rightarrow chal$: The third-party auditor randomly selects two numbers (k_1, k_2), where k_1, k_2 are the seeds of pseudo-random permutation and pseudo-random function. The third-party auditor sends the total challenge block count $cb \in [1, n]$ together with the pseudo-random seeds as a challenge to the CSP, where challenge denotes $chal = (k_1, k_2, cb)$.

ProofGen$(F, T, chal) \rightarrow P$: After receiving the challenge information, the cloud service provider calculates the indexs of challenge blocks according to k_1, the challenge blocks index $v_i = \phi(k_1, i)$. Then uses k_2 to calculate the random parameter $a_i = \varphi(k_2, i)$, where $1 \leq i \leq cb$. At the same time, the cloud service provider randomly selects a number r, calculates $R = u^r$, publishes R and saves r as a blinding factor. Then, the cloud service provider calculates T and F as follows:

$$\overline{T} = \prod_{i=1}^{cb} T_{v_i}^{a_i} \tag{8}$$

$$\overline{F} = \sum_{i=1}^{cb} a_i f_{v_i} + r \tag{9}$$

Finally, the CSP returns the proof $P = (\overline{F}, \overline{T})$ to the third-party auditor as a response to the challenge.

Verify$(X, chal, P) \rightarrow \{0, 1\}$: After receiving the evidence named P, the third-party auditor checks the equation $e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i} \cdot u^{\overline{F}}, X) = e(\overline{T}, g) \cdot e(R, X)$. If it holds, it outputs 1 to indicate that the challenged data block information is complete, otherwise, it outputs 0.

If the cloud service provider complies with the rules of this agreement, it verifies the correctness of the data integrity equation as follows:

$$\begin{aligned} & e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i} \cdot u^{\overline{F}}, X) \\ &= e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i} \cdot u^{\sum_{i=1}^{cb} a_i f_{v_i} + r}, X) \\ &= e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i} \cdot (\prod_{i=1}^{cb} u^{a_i f_{v_i}}) \cdot u^r, X) \\ &= e(\prod_{i=1}^{cb}((H(F_{id}\|i) \cdot u^{f_{v_i}})^{a_i} \cdot u^r, g^x) \\ &= e(\prod_{i=1}^{cb}((H(F_{id}\|i) \cdot u^{f_{v_i}})^{xa_i} \cdot (u^r)^x, g) \\ &= e(\prod_{i=1}^{cb}(T_{v_i})^{a_i} \cdot (u^r)^x, g) \\ &= e(\overline{T} \cdot (u^r)^x, g) \\ &= e(\overline{T}, g) \cdot e(u^r, g^x) \\ &= e(\overline{T}, g) \cdot e(R, X) \end{aligned} \tag{10}$$

IV. SECURITY ANALYSIS

Theorem 1: Suppose the CDH assumption holds in the group $\mathbb{G}_1$ and the hash function is regarded as a random oracle. In that case, the advantage of all adversaries in the DBCF model breaking IND-CPA security within probabilistic polynomial time is negligible.

Proof: Suppose a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ attacks the IND-CPA security of the DBCF encryption scheme, and challenger $\mathcal{B}$ is an attacker who breaks the CDH assumption. Challenger $\mathcal{B}$ knows (g, g^a, h), use $\mathcal{A}$ as a subroutine, and the goal is to calculate h^a.

Challenger $\mathcal{B}$ regards g^a as his public key, a is the secret key, but challenger $\mathcal{B}$ does not know the key a, then h^a is the generation of a specific tag by challenger $\mathcal{B}$. Since challenger $\mathcal{B}$ wants to hide the problem instance (g, g^a, h), $\mathcal{B}$ needs to choose a random number o and sends it to $\mathcal{A}$ with $(g^a)^o$ as the public key.

1) Initialization. Challenger $\mathcal{B}$ sends the generator g of group $\mathbb{G}_1$ and public key $X = (g^a)^o \in \mathbb{G}_1$ to adversary $\mathcal{A}$.
2) Query. The adversary $\mathcal{A}$ can make the following query in the bounded order of the polynomial.
   a) H-Query: Challenger $\mathcal{B}$ builds an H^{list}, which is initially empty, and the element type is triples $(f_i, H(f_i), b_i)$. When adversary $\mathcal{A}$ initiates the i-th inquiry, and the inquiry value is f_i, challenger $\mathcal{B}$ answers as follows:
      i) If there are items corresponding to f_i in H^{list}, which is $(f_i, H(f_i), b_i)$, then respond with $H(f_i)$.
      ii) Otherwise, $\mathcal{B}$ randomly selects $b_i \leftarrow Z_q$ and calculates $H(f_i) = g^{b_i} \in \mathbb{G}_1$.
      Take $H(f_i)$ as the response to the query, and store $(f_i, H(f_i), b_i)$ in the table.
   b) Tag-Query: The adversary $\mathcal{A}$ submits the file F to the challenger $\mathcal{B}$, and the challenger divides the file F into n blocks, $F = \{f_1, f_2, \cdots, f_n\}$). When adversary $\mathcal{A}$ requests file tag T_i, challenger $\mathcal{B}$ calculates:
   $$T_i = (g^{ao})^{b_i} \tag{11}$$
   And respond to adversary $\mathcal{A}$ with T_i. Because of $T_i = ((g^a)^o)^{b_i} = g^{b_i(ao)} = H(f_i)^{ao}$, T_i uses the key ao to label the file block.
   c) Verify-Query: In this step, the challenger $\mathcal{B}$ runs $Challenge(cb) \rightarrow chal$, then the challenger sends the challenge block information $chal$ to $\mathcal{A}$. The adversary $\mathcal{A}$ calculates the proof P by running $ProofGen(F, T, chal) \rightarrow P$ and returns the result to the challenger. Challenger $\mathcal{B}$ receives proof P and computes as follow, then the result $\{0, 1\}$ will be returned to $\mathcal{A}$.
   $$Verify(X, chal, P) \rightarrow \{0, 1\} \tag{12}$$
3) Final phase. In this stage, challenger $\mathcal{B}$ submits challenge information $chal^* = (k_1, k_2, cb)$ to adversary $\mathcal{A}$, challenges part of the file block and checks the data integrity, adversary $\mathcal{A}$ returns forged proof:
   $$P^* = (\overline{F}^*, \overline{T}^*) \tag{13}$$
   Let
   $$P = (\overline{F}, \overline{T}) \tag{14}$$
   be the correct proof. The forged proof will be computed in $ProofGen(F^*, T^*, chal^*)$. Hence, $(\overline{F}, \overline{T}) \neq (\overline{F}^*, \overline{T}^*)$. According to the proof P and P^*, it holds that:
   $$e(\overline{T}, g) \cdot e(R, X) = e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i} \cdot u^{\overline{F}}, X) \tag{15}$$
   and
   $$e(\overline{T}^*, g) \cdot e(R, X) = e(\prod_{i=1}^{cb}(H(F_{id}\|i)^{a_i} \cdot u^{\overline{F}^*}, X) \tag{16}$$
   Divide (15) by (16):
   $$\begin{aligned} e(\overline{T}/\overline{T}^*, g) &= e(\prod_{i=1}^{cb} u^{f_i - f_i^*}, X) \\ &= e(\prod_{i=1}^{cb}(g^b)^{f_i - f_i^*}, g^{ao}) \\ &= e((g^b)^{\sum_{i=1}^{cb} f_i - f_i^*}, g^{ao}) \\ &= e((g^{ab})^{\sum_{i=1}^{cb} o(f_i - f_i^*)}, g) \\ &= e((h^a)^{\sum_{i=1}^{cb} o(f_i - f_i^*)}, g) \end{aligned} \tag{17}$$
   In the formula, since at least one $f_i - f_i^*$ is not equal to 0, the probability of the denominator being zero is $1/q$, which is negligible. Therefore, challenger $\mathcal{B}$ can calculate h^a by the following formula:
   $$h^a = (\overline{T}/\overline{T}^*)^{\frac{1}{\sum_{i=1}^{cb} o(f_i - f_i^*)}} \tag{18}$$
   This proves up Theroem 1.

□

V. PERFORMANCE ANALYSIS

In this section, the communication overhead and computational overhead of the proposed DBCF scheme and experimental results are evaluated. In order to calculate the efficiency of this scheme, two schemes PPDP and RDPC are evaluated, which were proposed by documents [14] and [19].

A. COMMUNICATION COST

The communication cost of this protocol includes three parts: the data owner through the fog node uploads the data and the block tag T to the cloud server, the third-party auditor sends the challenge information chal, and the cloud server returns the challenge evidence. These three parts are respectively represented as DOtoCSP, TPAtoCSP, and CSPtoTPA.

Since the cloud node is used for communication, when the transmitted data is Z, the saved communication overhead can be expressed as:
$$\tau(Z) = \arg\min_{\mathbf{z} \in \Gamma} \left\{ \max \left[\frac{\lambda_i Z}{tran_{fn_i}} + w_{fn_i, fn_j} dist_{fn_i, fn_j} \right] \right\}$$
$$s.t.\ z_i \geq 0, \sum_{i=1}^{m} z_i = Z \tag{19}$$

TABLE 2. Comparison of communication cost.

Protocol	DOtoCSP	TPAtoCSP	CSPtoTPA
PPDP	$n\|\mathbb{G}_1\| + \|F\| + \|\psi\|$	$3\|Z_q^*\| + \|\psi\| + \|Sign(\psi)\|$	$\|\mathbb{G}_1\| + \|Z_q^*\|$
RDPC	$n\|\mathbb{G}_1\| + \|F\|$	$3\|Z_q^*\|$	$\|\mathbb{G}_1\| + \|Z_q^*\|$
Ours	$n\|\mathbb{G}_1\| + \|F\| - \tau(n\|\mathbb{G}_1\| + \|F\|)$	$3\|Z_q^*\| - \tau(3\|Z_q^*\|)$	$\|\mathbb{G}_1\| + \|Z_q^*\| - \tau(\|\mathbb{G}_1\| + \|Z_q^*\|)$

* $|\psi|$ and $|Sign(\psi)|$ represent the warrant size and its signature size.

TABLE 3. Comparison of computation cost.

Protocol	TagGen	ProofGen	Verify
PPDP	$T_p + (2n+1)T_{exp} + nT_{mul} + Sign$	$cbT_{exp} + (cb-1)T_{mul} + Ver$	$3T_p + (cb+2)T_{exp} + cbT_{mul}$
RDPC	$(2n+1)T_{exp} + nT_{mul}$	$cbT_{exp} + (cb-1)T_{mul}$	$2T_p + (cb+2)T_{exp} + cbT_{mul}$
Ours	$2nT_{exp} + nT_{mul}$	$cbT_{exp} + (cb-1)T_{mul}$	$3T_p + (cb+1)T_{exp} + cbT_{mul}$

* $Sign$ and Ver represent the computational cost of the signature and verification method in PPDP.

Assume that the scheme has n data blocks and cb challenge blocks. In DOtoCSP, the data upload stage, the data owner uploads the data block and the data block tag to the cloud service provider, and the communication overhead is n elements of $\mathbb{G}_1$ and file F. Since the cloud-fog node will be used to upload the data, the communication overhead is $n|\mathbb{G}_1|+|F|$. For transmission at the cloud-fog node, the actual communication overhead of this solution can be expressed as $n|\mathbb{G}_1| + |F| - \tau(n|\mathbb{G}_1| + |F|)$.

In the TPAtoCSP stage, the verifier submits challenge information to the cloud service provider, including the number of challenge blocks cb and two random numbers k_1, k_2, so the communication overhead is $3|Z_q^*|$. Considering the actual communication overhead under the cloud-fog node is $3|Z_q^*| - \tau(3|Z_q^*|)$.

In the CSPtoTPA stage, the cloud service provider uses the number of challenge blocks and two random numbers to generate evidence. The communication cost is $|\mathbb{G}_1| + |Z_q^*|$, and the actual communication cost in the cloud and fog environment is $|\mathbb{G}_1| + |Z_q^*| - \tau(|\mathbb{G}_1| + |Z_q^*|)$.

Table 2 compared the communication overhead of this scheme, PPDP and RDPC scheme. In the DOtoCSP stage, this scheme reduces the overhead of the warrant size compared with the PPDP scheme. At the same time, the overhead of the cloud-fog node ($\tau(n|\mathbb{G}_1| + |F|)$) is less than that of the other two schemes. In the TPAtoCSP stage, compared with the other two schemes, the communication overhead of a challenge block and two random number seeds transmitted in the cloud-fog node is reduced, which is $\tau(3|Z_q^*|)$. Using this scheme only needs to pass the challenge block number and the random number seeds in the challenge stage, which reduces the communication overhead of the warrant size and signature size. In the CSPtoTPA stage, CSP returns the proof block and the proof label, therefore the communication overhead is $|\mathbb{G}_1| + |Z_q^*|$. Compared with the previous two schemes, this scheme reduces the transmission overhead in the cloud-fog node. This scheme considers the cloud-fog node and simplifies the amount of data required for communication, and the communication overhead in the three stages is smaller than that of the PPDP and RDPC schemes.

B. COMPUTATION COST

Let T_p, T_{exp} and T_{mul} represent the bilinear mapping, multiplication and exponential operations on the multiplication cyclic group $\mathbb{G}_1$. Since the calculation cost of operations such as hashing and pseudo-random number generation is meager, they are ignored in calculating the overhead.

In the tag generation stage, the data owner runs the TagGen algorithm, and its computational cost is $2nT_{exp} + nT_{mul}$. For the ProofGen algorithm, the computational cost of $cbT_{exp} + (cb-1)T_{mul}$ is required. However, in the verification stage, the verifier runs the Verify algorithm, and the computational cost is $3T_p + (cb+1)T_{exp} + cbT_{mul}$. Table 3 compares the computation overhead of our scheme with PPDP and RDPC scheme.

According to Table 3, this scheme reduces the operation steps in the calculation and verification process without reducing the security. In the TagGen algorithm, this scheme reduces the computational cost of bilinear mapping, signature and multiplication compared with the PPDP scheme, and reduces the computational overhead of multiplication compared with the RDPC scheme. In the proof generation algorithm, this scheme reduces the computational cost of the verification part compared with the PPDP scheme. In terms of the computational cost of verification, the cost of this scheme is close to that of the PPDP and RDPC schemes.

C. EXPERIMENTAL RESULTS

In order to evaluate the performance of this scheme, experiments are carried out based on the Pairing-Based Cryptography Libarary (PBC) [32]. The data owner and auditor are simulated by HUAWEI Matebook 14, configured with Intel Core i5-10210U CPU @2.11 GHz and 16GB RAM. The cloud service provider is simulated by a server configured with Intel Core i9-9900KF CPU @3.60GH and 32GB RAM.

In this experiment, the file is divided into 100, 200, 300, 400, 500 blocks, and the file size of each block is 1MB. The time for label generation is shown in Figure 4. The experimental results show that as the number of blocks that need to generate tags increases, the tag generation time

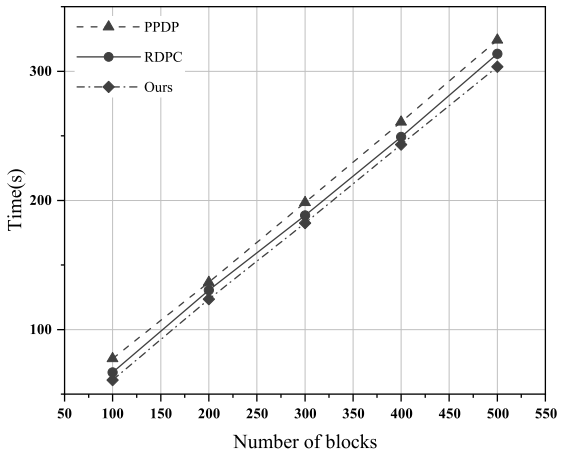

FIGURE 4. Tag generation overhead.

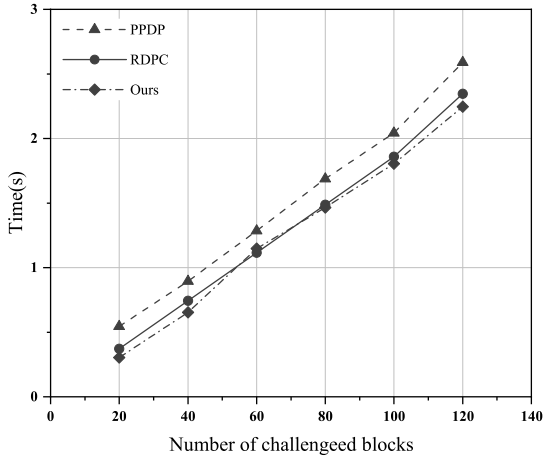

FIGURE 5. Proof generation overhead.

gradually increases, and the tag generation speed of this scheme is close to that of the PPDP and RDPC schemes. However, this scheme generates less hash information than the previous two schemes when generating tags, generating tags will be slightly faster.

In the proof generation and verification stage, the number of challenge blocks is set to 20, 40, 60, 80, 100, and 120, respectively. The results are shown in Figure 5. It can be seen that this scheme is linearly related to the proof calculation time of the PPDP and RDPC schemes and the number of challenge blocks. This scheme is better than the PPDP scheme and is equal to the proof calculation time of the RDPC scheme. In the verification phase, the auditor verifies the integrity of the data through the Verify algorithm and conducts experiments on different numbers of challenge blocks. The experimental results are shown in Figure 6. Since the calculation overhead of the PPDP proof generation stage is independent of the number of challenge blocks, the calculation overhead of the PPDP solution during the proof

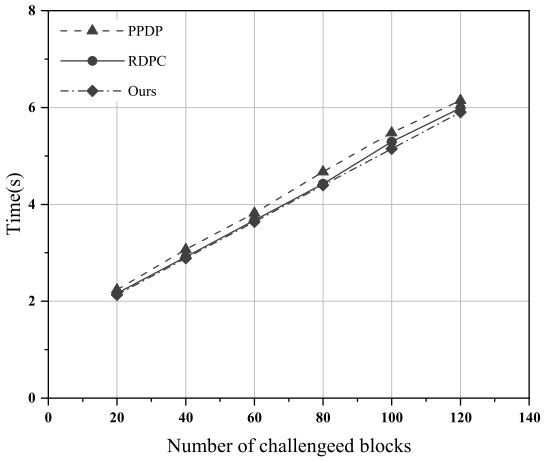

FIGURE 6. Verify overhead.

generation is constant. The calculation time of this scheme is close to that of the RDPC scheme in the verification stage.

According to the experimental results of the label generation, proof generation and verification stages, the speed of this scheme is greatly improved in the label generation and proof generation stages, while the speed in the verification stage is relatively close. In the process of data integrity auditing, the steps of reducing computational cost and operation can effectively reduce computational cost. If the audit process is similar, the overall cost can be reduced by ensuring security and eliminating redundant verification steps. However, based on the theoretical analysis of communication and computing overhead, cloud and fog computing can be used to reduce communication delay and simplify verification steps in data integrity auditing to reduce computing overhead.

Comprehensive experimental results and analysis can conclude that this scheme is more efficient and safer than PPDP and RDPC schemes.

VI. CONCLUSION

This paper proposes a DBCF protocol in the cloud and fog environment. This protocol can ensure data security in the case of data integrity auditing. This scheme introduces a blind factor in the data verification process, and adds random values to each verification, thereby avoiding the adversary's multiple requests to obtain user information. At the same time, the fog computing layer is established, and the cloud and fog structure is used to change the architecture of the transmission network, which can effectively reduce the communication overhead. In addition, the security model is given and proved to be secure under the random oracle model assumed by CDH. Finally, the performance analysis shows that this protocol will be more efficient in practical applications. In future work, the architecture model of the fog computing layer can be improved to make it more efficient.

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[2] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Comput. Secur.*, vol. 72, pp. 1–12, Jan. 2018.
[3] Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote integrity checking," in *Proc. Working Conf. Integrity Internal Control Inf. Syst.* Boston, MA, USA: Springer, 2003, pp. 1–11.
[4] H. Wang, D. He, A. Fu, Q. Li, and Q. Wang, "Provable data possession with outsourced data transfer," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 1929–1939, Nov. 2021.
[5] C. C. Erway, A. Küpçü, and C. Papamanthou, "Dynamic provable data possession," *ACM Trans. Inf. Syst. Secur.*, vol. 17, no. 4, pp. 1–29, 2009.
[6] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–35, 2018.
[7] H. Wang, L. Feng, Y. Ji, B. Shao, and R. Xue, "Toward usable cloud storage auditing, revisited," *IEEE Syst. J.*, vol. 16, no. 1, pp. 693–700, Mar. 2022.
[8] J. Chang, B. Shao, Y. Ji, M. Xu, and R. Xue, "Secure network coding from secure proof of retrievability," *Sci. China Inf. Sci.*, vol. 64, no. 12, Dec. 2021, Art. no. 229301.
[9] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.
[10] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Cham, Switzerland: Springer, 2014, pp. 169–186.
[11] M. Ma, D. He, D. Kumar, K.-K. R. Choo, and J. Chen, "Certificateless searchable public key encryption scheme for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 759–767, May 2017.
[12] J. Zhou, T. Wang, M. Z. A. Bhuiyan, and A. Liu, "A hierarchic secure cloud storage scheme based on fog computing," in *Proc. IEEE 15th Int. Conf. Dependable, Auton. Secure Comput., 15th Int. Conf. Pervasive Intell. Comput., 3rd Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Nov. 2017, pp. 470–477.
[13] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1143–1155, Oct. 2017.
[14] H. Yan, J. Li, and Y. Zhang, "Remote data checking with a designated verifier in cloud storage," *IEEE Syst. J.*, vol. 14, no. 2, pp. 1788–1797, Jun. 2020.
[15] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 598–609.
[16] A. Juels and B. S. Kaliski, "Pors: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 584–597.
[17] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw. (SecureComm)*, 2008, pp. 1–10.
[18] H. Shacham and B. Waters, "Compact proofs of retrievability," *J. Cryptol.*, vol. 26, no. 3, pp. 442–483, Jul. 2013.
[19] H. Wang, "Proxy provable data possession in public clouds," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 551–559, Oct./Dec. 2013.
[20] Y. Ren, J. Xu, J. Wang, and J.-U. Kim, "Designated-verifier provable data possession in public cloud storage," *Int. J. Secur. Appl.*, vol. 7, no. 6, pp. 11–20, Nov. 2013.
[21] KashifMunir and L. A. Mohammed, "Secure third party auditor(TPA) for ensuring data integrity in fog computing," *Int. J. Netw. Secur. Appl.*, vol. 10, no. 6, pp. 13–24, Nov. 2018.
[22] J. A. Alzubi, O. A. Alzubi, G. Suseendran, and D. Akila, "A novel chaotic map encryption methodology for image cryptography and secret communication with steganography," *Int. J. Recent Technol. Eng.*, vol. 8, no. 1C2, pp. 1122–1128, 2019.
[23] J.-F. Tian and H.-N. Wang, "An efficient and secure data auditing scheme based on fog-to-cloud computing for Internet of Things scenarios," *Int. J. Distrib. Sensor Netw.*, vol. 16, no. 5, pp. 1–15, 2020.
[24] K. Gu, N. Wu, B. Yin, and W. Jia, "Secure data query framework for cloud and fog computing," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 1, pp. 332–345, Mar. 2020.
[25] S. Xu, J. Ning, Y. Li, Y. Zhang, G. Xu, X. Huang, and R. Deng, "Match in my way: Fine-grained bilateral access control for secure cloud-fog computing," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1064–1077, Mar./Apr. 2020.
[26] O. A. Alzubi, J. A. Alzubi, O. Dorgham, and M. Alsayyed, "Cryptosystem design based on Hermitian curves for IoT security," *J. Supercomput.*, vol. 76, no. 11, pp. 8566–8589, Nov. 2020.
[27] J. A. Alzubi, R. Manikandan, O. A. Alzubi, I. Qiqieh, R. Rahim, D. Gupta, and A. Khanna, "Hashed needham schroeder industrial IoT based cost optimized deep secured data transmission in cloud," *Measurement*, vol. 150, Jan. 2020, Art. no. 107077.
[28] H. Noura, O. Salman, A. Chehab, and R. Couturier, "Preserving data security in distributed fog computing," *Ad Hoc Netw.*, vol. 94, Nov. 2019, Art. no. 101937.
[29] N. Döttling and S. Garg, "Identity-based encryption from the Diffie–Hellman assumption," *J. ACM*, vol. 68, no. 3, pp. 1–46, Mar. 2021.
[30] J. Chang, H. Wang, F. Wang, A. Zhang, and Y. Ji, "RKA security for identity-based signature scheme," *IEEE Access*, vol. 8, pp. 17833–17841, 2020.
[31] Y. Liu, L. Y. Zhang, J. Wang, Y. Zhang, and K.-W. Wong, "Chosen-plaintext attack of an image encryption scheme based on modified permutation–diffusion structure," *Nonlinear Dyn.*, vol. 84, no. 4, pp. 2241–2250, 2016.
[32] H. A. Al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," *IEEE Access*, vol. 5, pp. 22313–22328, 2017.

GENQING BIAN received the Ph.D. degree from the School of Management, Xi'an University of Architecture and Technology (XAUAT), Shaanxi, China. He is currently a Professor with XAUAT. He is also a member of the China Computer Federation (CCF) and the Association for Computing Machinery (ACM). His research interests mainly include information security, cloud computing security, and data analysis.

YANRU FU is currently pursuing the master's degree with the School of Information and Control Engineering, Xi'an University of Architecture and Technology, Shaanxi, China. Her research interests include cloud computing security and privacy protection.

BILIN SHAO received the B.S. degree from the School of Management, XAUAT, Shaanxi, China. He is currently a Professor with XAUAT. He is also a member of the China Computer Federation (CCF). His research mainly includes information security, information management technology, cloud computing security, and VANETS security.

FAN ZHANG (Member, IEEE) received the Ph.D. degree from the Department of Computing, University of Surrey, U.K. Her current research interests include information security, cloud computing security, and data mining.