Received August 8, 2020, accepted September 2, 2020, date of publication September 14, 2020, date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023750

Secure Data De-Duplication Based on Threshold Blind Signature and Bloom Filter in Internet of Things

BO MI[1], YANG LI[1], HUANG DARONG[1], (Member, IEEE), TIANCHENG WEI[1], AND QIANQIAN ZOU[2]

[1]College of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China
[2]School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China

Corresponding authors: Yang Li (ly815836@163.com) and Qianqian Zou (201811200002@stu.zuel.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61703063, Grant 61903053, Grant 61663008, and Grant 61573076; in part by the Chongqing Natural Science Foundation of China under Grant CSTC2017jcyjAX0411 and Grant CSTC2017jcyjA1665; in part by the Science and Technology Research Project of Chongqing Municipal Education Commission of China under Grant KJ1705139, Grant KJ1605002, Grant KJ1705121, and Grant KJZD-K201800701; and in part by the Program of Chongqing Innovation and Entrepreneurship for Returned Overseas Scholars of China under Grant cx2018110.

ABSTRACT Within the cloud environment, the availability of storage, as well as bandwidth, can be effectively preserved in virtue of data de-duplication. However, refraining redundancy from additional storage or communication is not trivial due to security concerns. Though intensive researches have been addressed on a convergent cryptosystem for secure data de-duplication, the conflicts amongst functionality, confidentiality, and authority remain unbalanced. More concretely, although data are obfuscated under convergent encryption, a violent dictionary attack is still efficacious since the whole pseudorandom process relies heavily on plaintexts. As for data ownership, the download privilege, which depends on hash value, may also be infringed due to the same reason. To dispose of these problems, we presented a conspiracy-free data de-duplication protocol based on a threshold blind signature in this article. With the help of multiple key servers, the outsourced file and de-duplication label will be computationally indistinguishable from random strings. We used the Boom filter as a tool to implement a proof of ownership, ensuring that the ownership claims made by users are real. It effectively prevents the attacker from using the stolen tag to get the whole file to gain file access without authorization. The most significant innovation of this article is to use homomorphism computation to aggregate and generate partial signature tags, and to introduce a secret sharing mechanism based on The Chinese Remainder Theorem to hide signature keys, thus balancing the security concerns of cloud and client. Compared with existing schemes, both communication and computation performances are preferable in our protocol. As far as we know, our scheme is the only data de-duplication scheme that satisfies the semantic security of ciphertext and label.

INDEX TERMS Cloud, secure data de-duplication, threshold blind signature, Bloom filter.

I. INTRODUCTION

With the development of cloud computing and big data, more and more enterprises and clients choose to outsource their files to the cloud for convenient storage and management, making the occupancy of cloud disk exponentially growth. According to the "White Paper on Forecasting and Methods 2015-2020" released by Cisco Global Cloud Index [1], the number of users registering on personal cloud storage service will increase from 1.3 billion in 2015 to 2.3 billion in 2020, with global data rising to 40ZB. Encountered with such a massive amount of data, how to leverage the economy and efficiency of cloud sources has become an inevitable challenge for cloud service providers.With the popularization of the Internet of Things, the privacy data exposure problem in the Internet of Things has gradually been exposed.There are a lot of duplicate privacy data in the Internet of Things. If these duplicate data can be reasonably removed, the risk of privacy leakage can be reduced.

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

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

To improve storage availability and reduce management costs, cloud service providers prone to draw support from de-duplication technology, using randomly sampled strings or hash values as labels to avoid redundantly uploading identical data. According to the phase when de-duplication is carried out, relevant techniques can be divided into two categories. (1) server-side de-duplication: Users trivially upload their data to the cloud, and the server will be responsible for detecting and striking out reduplicated data on its own. Though such an idea preserved the functionality of de-duplication, massive bandwidth will be consumed to transmit unnecessary data. (2) Client-aided de-duplication: Before uploading, the client calculates the hash function of data for cloud retrieval. Once received the hash value, cloud server checks if the same label exists within its local storage system. If so, the server will instruct to cancel the actual upload process and mark the client as an owner of the corresponding data. Since hash values are always abbreviated, this method could significantly cut down both storage space and transmission overhead.

Concerning security issues of client-aided data de-duplication, cloud servers are supposed to be honest, but also curious, which may interest in the privacy outsourced by users [2], [3]. Therefore, clients are apt to upload their data in the form of ciphertexts. Nevertheless, since distinct users spontaneously choose the keys, the same files will be encrypted as different ciphertexts, which is not feasible for redundancy detection.

Aiming at the two problems of privacy disclosure and unauthorized access in cloud data de-duplication scheme based on convergence encryption, this article proposes a ciphertext data de-duplication scheme combining blind signature and door threshold secret sharing. By introducing a series of secondary key servers, not only can the encrypted file be protected from dictionary attacks, but it also can tolerate partial destruction of cloud nodes. Since the basic primitive we use is the combination of the Chinese remainder theorem and discrete logarithm problem, unconditional security is realized in the proof to solve the problem of privacy disclosure. To ensure the authenticity of the ownership claimed by customers, we use The Bronc filter to realize the document ownership certificate, which effectively prevents the attacker from using the stolen label to obtain the complete document, thus obtaining the file access without authorization. The simulation results show that the scheme has a high efficiency in computing and communication overhead. The most significant innovation of this article is to use homomorphism computation to aggregate and generate partial signature tags, and to introduce a secret sharing mechanism based on The Chinese remainder theorem to hide signature keys, thus balancing the security concerns of cloud and client. After formal security proof, our scheme will be compared with some recent methods [4], [5], showing that fewer costs regarding uploading and de-duplication are required.

II. RELATED WORKS

The concept of convergent encryption (CE) was firstly proposed by Douceur *et al.* [6] in 2002, in which the key generation algorithm is set as deterministic to ensure that the same data produces the same key, taking their hash outputs for example. Then, over the next decade or so, many researchers studied CE.This is shown in Table 1. Among them, Li's scheme [7] has achieved remarkable results. Directing at clear security objectives with a formal description model, Li *et al.* [7] combined the CE algorithm with convergence diffusion mechanism, proposed a scheme of cd store to achieve data security and de-duplications. And their experiments show that the proposed scheme saves nearly 70 % of the storage cost.

TABLE 1. Related Work about CE.

researchers	contribution
Bellare et al. [8]	Message Lock Encryption algorithm
Keelvendhiet al. [5]	The DupLESS scheme
Miao et al. [4]	Improving the DupLESS scheme
Li et al. [9]	A updating the file key scheme
Qin et al. [10]	A scheme combining with MLE and aont-rs secret sharing mechanism

But it is worth mentioning that both the ciphertext and de-duplication label of the schemes discussed above is not semantically secure.

As for the authority of file downloading, PoW is commonly used to ensure the user's data ownership, which is mainly established on Merkle hash tree (MHT), random sampling, or generalized hash function. In 2011, Halevi *et al.* [11] proposed the concept of PoW for the first time. The core idea behind it is to compress the file on the client-side and build a Markov tree according to it. Thus, when authentication is needed, the server may launch a challenge, and the client is responsible for returning a series of paths from the root to designated leaves. Only if all of the ways are valid, the server will confirm that the user owns the file and direct him to its storage.

TABLE 2. Related Work about PoW.

researchers	contribution
Wan et al. [12]–[16]	A fast multi-class data retrieval method
Wan et al. [17]–[20]	A data query method based on Internet of Things architecture
Zhu et al. [21]–[25]	An Influence Model based on Heterogeneous Online Social network for Influence Maximization
Xiong et al. [26]	A new method for sharing files
Tang et al. [27]	A query method based on multiple attributes
Zhao et al. [28]–[34]	A privacy protection scheme for multi-source data in the cloud.
Qi et al. [35], [36]	A data-driven service recommendation with privacy-preservation.
Lv et al. [15], [37]–[40]	QoS prediction for service recommendation with deep feature learning in edge computing environment

Then, many researchers subsequently conducted extensive studies on PoW. This is shown in Table 2.

It's worth noting that Blasco *et al.* [41] then proposed a flexible, scalable, and provably secure data de-duplication scheme based on Bloom filter. Briefly speaking, the server divides the received file into blocks of the same size, calculates their corresponding tokens through a pseudo-random function (PRF), and inserts the symbols into a Bloom filter, which will be stored as a triple data structure. During the challenge phase, the server barely asks the client to upload a certain number of tokens to validate his ownership.

In 2016, Xiong *et al.* [42] made a general survey on the curriculum development of cloud security data de-duplication, analyzed and compared different schemes, and pointed out the problems of various schemes.

III. PRELIMINARIES

In this section, we present some building blocks that underpin our final scheme.

A. CONVERGENT ENCRYPTION

For the sake of secure de-duplication in the cloud environment, a convergent key, together with a label, is generally originated from the file to be uploaded. To preserve privacy, the user exploits the convergent key for data encryption. When a file is supposed to be outsourced, the user submits its corresponding label in advance, which can be used to check the presence of duplication on the server-side.

It was evident that, once a duplication is found, real uploading would be unnecessary, and only a link should be sent back to the user. During this process, two tough but significant problems must be addressed. One is that the concurrent key and label must be statistically independent since the server may be interested in deducing the key from the label. The other is how to make the same key apparent to all data owners for correct decryption.

B. DISCRETE LOGARITHM PROBLEM

Based on a cyclic group G of order q, the computational discrete logarithm problem (CDH) can be described as, for any $a \in Z_q^*$, it is difficult to recover a form $g^a \in G$, even if the generator g is given. Another version of this problem is named decisional discrete logarithm problem (DDH), which aims to distinguish $g^{ab} \in G$ from a randomly sampled group element $g^c \in G$ in case that both $g^a \in G$ and $g^b \in G$ are present. DDH assumption is more robust than the CDH hypothesis and can thus be deduced from the latter.

C. BLIND SIGNATURE

Blind Signature was firstly proposed by Blasco *et al.* [41] in 1982, requiring that the signer must be ignorant of any information about what he signed. The reason that we draw support from a blind signature for duplication inspection is to conceal the relevance between the file and its hash value. Thus violent dictionary attacks can be avoided.

D. CTR BASED THRESHOLD SECRET SHARING

Considering that part of key servers may collude with the cloud to violate user's privacy, we resort to CTR (Chinese Remainder Theorem) threshold secret sharing scheme to introduce an auxiliary property that signing can be executed only if more than $t-1$ key servers are available.

For n key servers represented as $KS = \{KS_1, \cdots, KS_n\}$, the (t, n)-threshold secret sharing scheme is realized as below.

1) SECRET SHARING

The distribution center (DC) selects a prime q which is larger than any secret SK, an integer A and a sequence $d = \{d_1, d_2, \cdots, d_n\}$ satisfying

(1) $0 \leq A \leq \lfloor N/q \rfloor - 1$;

(2) $d_1 < d_2 < \cdots < d_n$ are strictly monotone increasing;

(3) For all $i = 1, 2, \cdots, n$,$\gcd(d_i, q) = 1$ and $\gcd(d_i, d_j) = 1$ if $i \neq j$;

(4) For $2 \leq t \leq n$, $N = \prod_{i=1}^{t} d_i > q \prod_{i=1}^{t-1} d_{n-i+1}$.

Then it calculates $L = SK + Aq$ and secretly transmit (L_i, d_i), where $L_i = L \bmod d_i$, as a share of secret SK to corresponding key server KS_i.

2) SECRET RESTORATION

Denoting $KS' = \{KS'_1, KS'_2, \cdots, KS'_{t'}\} \subseteq KS$ as a subset of key servers who are willing to reveal the secret SK.For clarity, we denote $|KS'|$ as t' in the rest of this article. Since all shares (L_j, d_j) conserved by these participants comply with a congruence equation that

$$L_j \equiv L \bmod d_j, j = 1, 2, \cdots, t'. \tag{1}$$

Once at least t secret pieces are available, it is trivial to restore the shared value by computing

$$SK = (\sum_{j=1}^{t'} \frac{D}{d_j} e_j L_j \bmod D) \bmod q \tag{2}$$

where $D = \prod_{j=1}^{t'} d_j$ and $\frac{D}{d_j} e_j \equiv 1 \bmod d_j$

It is worth noting that the secret SK should not be exposed to any participant in our scheme; thus, each server is supposed to sign the message privately via its secret share L_j as a preconditioning step.

E. BLOOM FILTER

The Bloom filter is employed as PoW to authenticate user's ownership about a specific file. Under random oracle model, the filter is a 2^l-dimensional vector $BF \in \{0, 1\}^{2^l}$ relating to a hash function assemble $\overline{H} = \{H_k : \{0, 1\}^* \rightarrow \{0, 1\}^l\}_{k=1,2,\cdots,K}$..Given a binary string m, the Bloom filter can be constructed from a zero vector by setting all its $H_k(m)$th elements to one. When checking whether a client possesses an alleged file, the cloud can request his response for those hash values. According to the structure of BF, if at least one position-specific to the user's reply is not correct, the authentication will be taken as invalid.

IV. PROBLEM DESCRIPTION

The main objective of our scheme is to refrain from the client from uploading reduplicative files. However, considering that

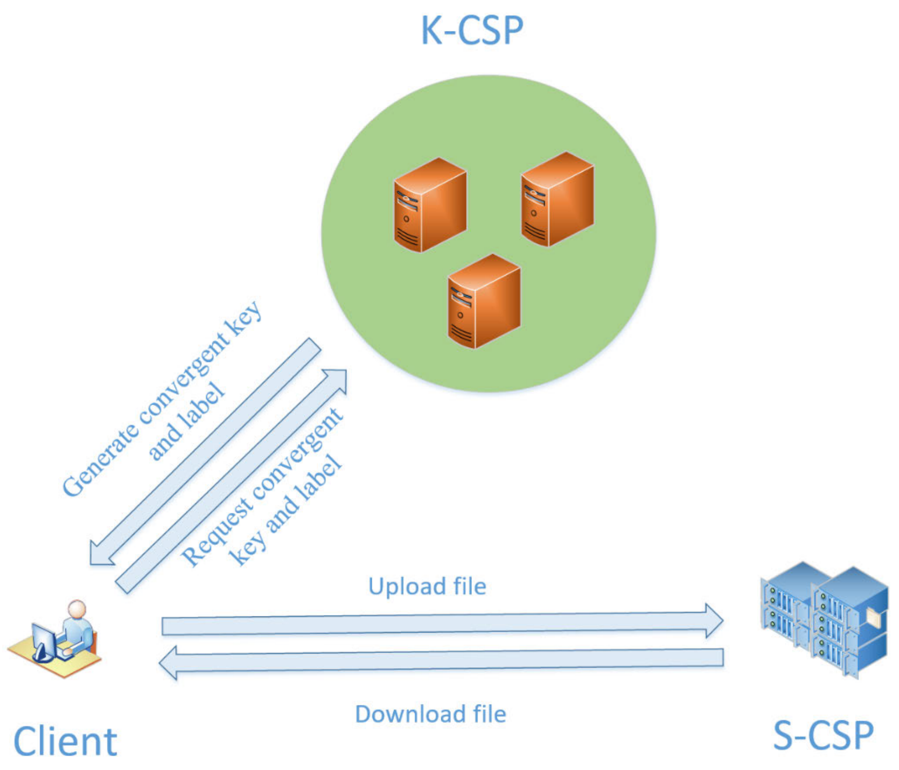
K-CSP
Generate convergent key and label
Request convergent key and label
Upload file
Download file
Client
S-CSP

FIGURE 1. System model for secure De-duplication.

cloud and key servers may conspire to usurp the user's privacy and a malicious attacker may deceive to gain illegal access rights, such a task is not trivial. To alleviate the conflicts between functionality and security, we first define the system and security models as below.

A. SYSTEM MODEL

As shown in Figure 1, a group of key servers is introduced besides a cloud server and several clients. The reason behind this is to make sure that once the number of corrupted servers are less than a threshold, nothing about the user's plaintext can be learned from uploaded information.

The roles of three kinds of participants as defined as the following.

1) CLIENTS

The roles of three kinds of participants as defined as the following. They were uploading their encrypted files to the cloud. However, if replication is detected, they save the corresponding link and PoW at local without actual uploading.

2) KEY SERVER (K-CSP)

It provided with part of the master key to cooperatively generate convergent keys for clients. It is noteworthy that some of them may be semi-malicious and league with a cloud server for interested user privacy.

3) CLOUD SERVER (S-CSP)

Act as an agency to conserve user's data. Moreover, it is also responsible for saving file labels for retrieval and access control. The cloud is assumed to be semi-malicious.

B. THREAT MODEL

For any probabilistic polynomial time (PPT) adversary, they may statically compromise a series of nodes to launch an associated attack. To prevent from being perceived and subsequently published by law, the attacker is prone to be semi-malicious, which falls into one of the two categories.

1) COLLUDED SERVERS

It is reasonable to take the cloud and part of key servers as honest but curious, meaning that they follow the protocol except for random coins but try to reveal relevant information from uploaded data beyond their privileges.

2) UNAUTHORIZED USERS

To access some files without corresponding PoWs, an illegal or revoked client may deceitfully claim that he possesses file ownerships by intercepting the communication channel or exploiting known hash values. We also consider some attackers that are out of the system, so this kind of adversary does not have to comply with the established protocol and are deemed as entirely malicious.

C. SECURITY OBJECTIVES

Directly followed by the threat model mentioned above, two secure objectives are defined concerning confidentiality and authenticity.

1) CONFIDENTIALITY

Suppose that the ciphertext of uploaded file m is c, for at most t-1 corrupted key servers colluding with the cloud server,

- On input message $\overline{m}$

- Output ciphertext $\overline{c}$, satisfying $Pr[\overline{c} = c|\overline{m} = m] < neg(\lambda)$,where neg($\cdot$) is a negligible function related to security parameter λ.

This implies that the adversary cannot learn any information about uploaded files if the threshold property is guaranteed.

2) AUTHENTICITY

For any malicious client who is not provided with the file m at the very beginning,

- On input message h, where $h = H(m)$

- Output $\overline{m}$ for $Pr[\overline{m} = m|h = H(m)] < neg(\lambda)$.

That is to say; the client is incapable of acquiring the file even if he obtained its corresponding hash value.

V. THE PROPOSED SCHEME

The main idea behind our scheme is exploiting a series of key servers to cooperatively sign the hash value of the file to generate a convergence key. Thanks to the preliminary of CRT-based threshold secret sharing, such the key cannot be reproduced by a minority of key servers who conspire with the cloud. Meanwhile, since the signature is blindly generated via the semantic secure crypto algorithm, no information about the file would be exposed to any signer. Considering that the convergence key and duplication label are statistically independent of the hash value, dictionary attacks can thus be averted. Moreover, we also take the threaten of ownership deception into account. The solution is to build a PoW according to Bloom filter, which will be capable of excluding unauthorized users with overwhelming probability.

For clarity, our scheme will be presented as three parts, namely System Initialization, File Uploading, and File download.

A. SYSTEM INITIALIZATION

As shown in Figure. 1, the system is configured as one cloud and n key servers, serving numerous clients who expect to outsource their files without privacy infringement.

To distribute the shares of the master signing key, an auxiliary DC can be trivially employed to achieve the task. However, the distribution center may also be a pothole where the key may be divulged.

For each key server KS_i, it stochastically chooses a key share SK_i together with a private integer A_i satisfying.

(1) $0 < SK_i < \lfloor q/n \rfloor$;

(2) $0 \le A_i \le (\lfloor N/q \rfloor - 1)/n$.

Then it calculates $a_{ij} = SK_i + A_i q \bmod d_j$ for $j = 1, 2, \cdots, n$ and delivers them to corresponding servers.

After receiving all a_{ij} from other nodes, the share L_j for server KS_j can be directly computed as

$$\begin{aligned} L_j &= \sum_{i=1}^{n} a_{ij} \bmod d_j \\ &\equiv \sum_{i=1}^{n} (SK_i + A_i q). \end{aligned} \tag{3}$$

Denoting $SK = \sum_{i=1}^{n} SK_i$ and $A = \sum_{i=1}^{n} A_i$,since $SK + Aq \equiv L_j \bmod d_j$, it is obvious that the secret SK is securely shared without exposed to any key server. To this point, each member takes L_j as its signing key and release $PK_j = g^{\frac{D}{d_j} e_j L_j - A_j \bmod D} \bmod q$ to jointly publish system public key as $PK = \prod_{j=1}^{n} PK_j \bmod q$.

B. FILE UPLOADING

When a file m is supposed to be updated, a convergent key k_c is computed by signing its hash value blindly with the help of key servers. Then the client transmits the label TF to cloud for duplication inspection. If such a file does already exists, the cloud hands a link back to the user without real uploading. Specifically, the protocol is described as follows.

*Step*1) Client permutes and divides the file into n blocks as $\widetilde{m} = f_p(m) = \widetilde{m_1}|\widetilde{m_2}| \cdots |\widetilde{m_n}$.according to a fixed permutation function $f_p(\cdot)$.Then it calculates $H(\widetilde{m_i})$ for $i = 1, 2, \cdots, n$ in terms of a pre-defined hash function $H(\cdot) : \{0, 1\}^* \rightarrow Z_q^*$ and secretly deliver them to their corresponding key servers.

*Step*2) Once $H(\widetilde{m_i})$ is received, key server KS_i figures out $u_i = H(H(\widetilde{m_i})||k_{KS_i})$ according to its private but fixed random string k_{KS_i} and broadcasts $r_i = g^{u_i} \bmod q$ to the client as well as a subgroup KS' containing at least t servers. The servers who will not participate in the following procedure should also send their u_i secretly to the client.

*Step*3) Client randomly samples φ from Z_q^* and broadcasts

$$h' = \varphi r^{1-\varphi} h \bmod q. \tag{4}$$

to all servers in KS', where $h = H(m)$ and

$$\begin{aligned} r &= \prod_{i=1}^{n} r_i \bmod q \\ &\equiv g^{\sum_{i=1}^{n} u_i}. \end{aligned} \tag{5}$$

*Step*4) After all aforementioned information is collected, each participant KS'_j signs h' with its private key and returns the partial signature back to the client, that is

1)KS'_jcalculates r the same as formula(5).

2)Then it figures out

$$v_j = \frac{D}{d_j} e_j L_j \tag{6}$$

for $D = \prod_{j=1}^{t'} d_j$ and $\frac{D}{d_j} e_j \equiv 1 \bmod d_j$

3) Finally, it delivers a triad (r, D, s_j),where

$$s_j = u_j h' + r v_j. \tag{7}$$

as a reply to the user.

*Step*5) The client halts if the inputs regarding r or D are not coincident. Otherwise, it computes

$$s = (\sum_{j=1}^{t'} s_j \bmod D) \bmod q. \tag{8}$$

and de-blind it via

$$\begin{aligned} s' &= r^{\varphi-1}(s + \sum_{KS_i \notin KS'} u_i h') \bmod q \\ &\equiv \varphi U h + SK r' \end{aligned} \tag{9}$$

for $U = \sum_{i=1}^{n} u_i$ and $r' = r^{\varphi} \bmod q$.

*Step*6) Concerning about the requirement of de-duplication, we devise the replication inspection label as $TF = (tf_{maj}, tf_{sub})$,where

$$\begin{aligned} tf_{maj} &= g^{s'/\varphi} \bmod q \\ &\equiv g^{Uh+SK(g^{U\varphi}/\varphi)} \end{aligned} \tag{10}$$

and

$$tf_{sub} = g^{U\varphi}/\varphi \bmod q \tag{11}$$

Based on TF, a secure de-duplication sub-protocol can be carried out as follows.

1) Without loss of generality, assume $\overline{TF} = (\overline{tf}_{maj}, \overline{tf}_{sub})$ stands for the label submitted to check duplication. Before real uploading, the client transmits $\overline{TF}$ to the cloud who will inspect whether $tf_{maj}/\overline{tf}_{maj}$ is congruential with $PK^{tf_{sub}/\overline{tf}_{sub}}$ over group G for any exist TF. If such equivalence is found, the file should be not be uploaded again since the cloud has already held a copy of it.

2) Depending on whether or not replication is detected, our protocol will proceed in two different ways.

A. As for a fresh file, the client takes actions as below.

(1)Divides the file into K blocks, marked as $B_1, B_2, \cdots, B_K$ respectively, and computes all $p_k = f_{PRF}(H(B_k)||k)$ in terms of a pseudo-random function$f_{PRF}(\cdot) : \{0, 1\}^{\lfloor log_2 q \rfloor + \lfloor log_2 K \rfloor} \rightarrow \{0, 1\}^l$. Then converts all of the p_ith bits in Bloom filter BF to 1.

(2)Exploits the pre-defined symmetric cryptosystem to compute $c = E_{k_c = H(s'')}(m)$, where $s'' = s'/\varphi \bmod q$.

(3) Conserves all p_k corresponding to K blocks and uploads the index $A = (TF, BF)$ as well as the ciphertext c to the cloud.

B. Howbeit, if duplication is found in the cloud, the client has to prove its ownership of the file and acquire the symmetric key k_c.

(1) The cloud randomly selects K' blocks and sends their indexes k' to the client. Once the correspondent hash values $h_{k'} = H(B_{k'})$ are figured out, the client calculates $p_{k'} = f_{PRF}(h_{k'}||k')$ and sends them back to the cloud. For all K' received values, the cloud checks if all of the $p_{k'}$th bits in BF are 1 and terminates once an inconsistency comes to light. They were noting that the client should calculate and preserve all p_k for K blocks for further retrieval

(2)If the ownership is validated, the cloud simply transmits $\widetilde{tf}_{sub} = tf_{sub} - \widetilde{tf}_{sub} \bmod q$ to t' key servers. On received $\widetilde{tf}_{sub}$, each key server KS_j computes $\xi_j = \widetilde{tf}_{sub} v_j$ and privately sends it to the client. By computing

$$k_c = H((\overline{s}'' + \sum_{i=1}^{t} \xi_j \bmod D) \bmod q) \tag{12}$$

the client can correctly obtain the convergent key k_c.

(3) The cloud sends file link to the client.

C. FILE DOWNLOAD

Under the circumstances of file retrieval, the cloud must authenticate the PoWs for clients. That is to say, once a user launched a retrieval request, the cloud should verify his ownership about the Bloom filter of such a file. The process is similar to *Step*6) B (1), and the client will be allowed to download the ciphertext, which can be decrypted by k_c.

D. CORRECTNESS

We now prove that any replication can be correctly detected, and the convergent key achieved by the client is accurate.

Lemma 1: Based on de-duplication labels $TF = (tf_{maj}, tf_{sub})$, the cloud is able to determine the equality of their corresponding data.

Proof: After system initialization, each key server $KS'_j \in KS'$ is provided with a share L_j related to the secret key SK.Respectively, they sign the deformed hash of a message as $s_j = u_j h' + r v_j$, in accordance with equitation (4),(5) and (6). By collecting t' shares of the signature, the client combines them together with $\sum_{KS_i \notin KS'} u_i h'$ and achieves $s = Uh' + rSK \bmod q$ in terms of Lemma 1 and $nq^2 + nqd_n < N$. Since the client is aware of $r = g^U \bmod q$ and φ, he can then compute the label as $tf_{maj} = g^{Uh+SK(g^{U\varphi}/\varphi)}$ and $tf_{sub} = g^{U\varphi}/\varphi$.

Denoting TF and $\overline{TF}$ as two labels whose corresponding files are m and $\widetilde{m}$, then

$$\frac{tf_{maj}}{\overline{tf}_{maj}} = g^{(Uh - \overline{U}\overline{h}) + SK(g^{U\varphi}/\varphi - g^{\overline{U}\overline{\varphi}}/\overline{\varphi})} \bmod q \tag{13}$$

If the two files are duplicate, we have $U = \overline{U}$ and $h = \overline{h}$ since $h = H(m)$ and $U = \sum_{i=1}^{n} H(H(\widetilde{m}_i)||k_{KS_i})$ where $\widetilde{m}_i$ is deterministically extracted from m and k_{KS_i} is fixed for each key server. It means the above equation can be rewritten as

$$\begin{aligned} \frac{tf_{maj}}{\overline{tf}_{maj}} &= g^{SK(g^{U\varphi}/\varphi - g^{\overline{U}\overline{\varphi}}/\overline{\varphi})} \bmod q \\ &\equiv PK^{g^{U\varphi}/\varphi - g^{\overline{U}\overline{\varphi}}/\overline{\varphi}} \end{aligned} \tag{14}$$

which is equal to $PK^{tf_{sub} - \overline{tf}_{sub}} \bmod q$ because $tf_{sub} = g^{U\varphi}/\varphi \bmod q$

Once a duplication is found, the client does not need to upload his file any longer. However, if he retrieved the ciphertext of his data, it should be guaranteed that he can decode it. Therefore, we claim the correctness of retrieval in Theorem 2.

Theorem 2: For any valid user, he is capable of downloading and deciphering his file.

Proof: As for the PoW of a file, any legal client must be aware of all $p_k = f_{PRF}(H(B_k)||k)$ corresponding to file blocks $B_1, B_2, \cdots, B_k$.Therefore, he is able to respond every challenge launched for ownership authentication, thus downloading the ciphertext of his file.

In order to decrypt the ciphertext, valid clients must be provided with the convergent key k_c It is obvious that the client, who uploaded the file for the first time, is conscious of $k_c = H(Uh + SKr'/\varphi \bmod q)$, As for other clients, they received $\xi_j = ((tf_{sub} - \overline{(tf)}_{sub}) \bmod q)v_j$ from t' key servers and know $\overline{s}' = \overline{\varphi}Uh + SKg^{U\overline{\varphi}} \bmod q$. Thus, they can compute

$$\begin{aligned} k_c &= H(\overline{s}'' + \sum_{i=1}^{t} \xi_j \bmod D) \bmod q \\ &= H(Uh + SKg^{U\varphi/\overline{\varphi}} + SK(g^{U\varphi}/\varphi - g^{U\overline{\varphi}}/\overline{\varphi}) \bmod q \\ &= H(Uh + SKg^{U\varphi}/\varphi \bmod q) \qquad (15) \end{aligned}$$

to obtain the key in light with Lemma 1 and $q + nqd_n < nq^2 + nqd_n < N$.

VI. SECURITY ANALYSIS

In light of the aforementioned security objectives, we will prove that semi-malicious servers are unable to violate user's privacy, and fraudulent clients cannot deceive to access unauthorized files. To certify the security of our scheme, we give an impossibility in advance.

Impossibility: Disguised as a valid user who is provided with a file $\overline{m}$, the cloud can tell if it coincides with any outsourced file m, implying that the ciphertexts would no longer be indistinguishable.

The proof of such impossibility is straightforward due to the functionality of de-duplication. That is to say; any valid user is entitled to carry out a file label $\overline{TF}$, which can be used to detect if any uploaded file is corresponding to it by the cloud. Owing to the conflict between de-duplication and user privacy, we propose a weaker definition of the indistinguishability of ciphertexts as below.

Definition 1: For any probabilistic polynomial time (PPT) adversary Λ who is provided with all hashes $h \in H$ of the plaintexts $m \in M$, if the probability that he reveals the corresponding hash h of any ciphertext c is

$$\begin{aligned} Pr[\overline{h} = h, \overline{h} = \Lambda(H, c) | c = E_{k_c}(m), h = H(m)] \\ = 1/|M| + \xi(\lambda). \qquad (16) \end{aligned}$$

then the cryptosystem is indistinguishable under chosen hash attack. Herein,
M, H stand for the ciphertext and hash space, while $\xi(\lambda)$ is a negligible function in the security parameter λ and k_c is the symmetric key with regard to m.

Furthermore, our scheme also achieved a security level that if less than t key servers collude with the cloud, the ciphertext can even be indistinguishable under the chosen-plaintext attack, where the security of CRT-based secret sharing should be exploited.

Lemma 2: The private key SK is unconditionally secure if less than t key servers collude.

Proof: Suppose only $t' < t$ key servers participate in the process of key recovery, they compute $D = \prod_{j=1}^{t'} d_j$. Because $N = \prod_{i=1}^{t} d_i > q\prod_{i=1}^{t-1} d_{n-i+1}$,we have $D \leq \prod_{i=1}^{t-1} d_{n-i+1} < N/q$.Denoting $\sum_{i=1}^{n} SK_i + \sum_{i=1}^{n} A_i q = Z = z + \alpha D$, it is obvious that $0 \leq \alpha \leq \lfloor Z/D \rfloor - 1 \leq N/D$ in line with $\sum_{i=1}^{n} SK_i + \sum_{i=1}^{n} A_i q < N$. Since $\gcd(D, d) = 1$ and $q < N/D$, α can be any integer ranging from 0 to q at least, meaning that the possibility of revealing the exact $Z = \sum_{i=1}^{n} SK_i + \sum_{i=1}^{n} A_i q$ is not better than $1/q$ after modulo D. That is to say, they are incapable of acquiring $SK = \sum_{i=1}^{n}$except for coin guess.

Based on the definition of indistinguishability under chosen hash attack, we consider the situation when only the hash value of plaintext is available at first.

As for user's privacy, we claim that the cloud is unable to relate a hash value $\overline{h}$ to any outsourced ciphertext c even if he impersonates as a valid user.

Lemma 3: On hash value $\overline{h}$, the probabilistic polynomial-time cloud cannot determine whether it corresponds to an uploaded ciphertext c under chosen hash attack.

Proof: In order to decide if a copy c of file $\overline{m}$, where $H(\overline{m} = \overline{h})$, dose already exist on the disc, the cloud has to compute $\overline{TF} = (\overline{tf}_{maj}, \overline{tf}_{sub})$ for duplication inspection. However, since he knows nothing about $\overline{m}$ but its hash value h,the key servers cannot help to compute the correct value of $\overline{U} = \sum_{i=1}^{n} H(H(\widetilde{\overline{m}}_i)||k_{KS_i})$ for him. Denoting $\overline{U}'$ as the specious value he derived by coin tossing, the de-duplication label should be rewritten as $\overline{tf}'_{maj} = g^{\overline{U}'\overline{h}+SK(g^{\overline{U}'\overline{\varphi}}/\overline{\varphi})}$ and $\overline{tf}'_{sub} = g^{\overline{U}'\overline{\varphi}}/\overline{\varphi}$ Even if a ciphertext is really encrypted by $c = E_{k_c}(\overline{m})$ for $\overline{m} = m$ thus $\overline{h} = h = H(m)$, the value $tf_{maj}/\overline{tf}_{maj} = g^{(h(U-\overline{U}')+SK(g^{U\varphi}/\varphi - g^{\overline{U}\overline{\varphi}}/\overline{\varphi})}$ mod q takes only $1/q$ chance to be equal to $PK^{tf_{sub}-\overline{tf}_{sub}} = g^{SK(g^{U\varphi}/\varphi - g^{\overline{U}\overline{\varphi}}/\overline{\varphi}) \bmod q}$. In other words, the cloud cannot carry out de-duplication if the file itself is unknown. Noting that, since the convergent key is computed as $k_c = H(\overline{U}'h + SKg^{U\varphi/\varphi} \bmod q)$ according to the label, it is also impossible for him to decrypt the file even if such duplication is detected.

Without the file beforehand, the following conclusion can also be achieved concerning a malicious client.

Lemma 4: The user is incapable of using a hash value h to deceive for unauthorized download permission.

Proof: Assume that the user is aware of the link corresponding to h. Before downloading, he has to show the PoW of the file. However, since he is ignorant of the values $p_k = f_{PRF}(H(B_k)||k)$ corresponding to the file blocks, the probability that he passes any k' verifications of the Bloom filter checking is only $(1-(1-1/(l-1))^{KT})^{T}$, which is negligible,T represents the number of independent hash functions in BF.

Lemma 5: On hash value h, a malicious user is unable to decrypt its corresponding ciphertext c even if he conspires with the cloud.

Proof: Due to the fact that the ciphertext c in encrypted by $k_c = H(Uh + SKg^{\varphi})$ which is computed by the original uploader. Without being aware of the file m, they can only guess $\overline{U}'$ the same as that of Theorem 3. After carrying out $\bar{s}'' = \overline{U}'h + SKg^{\overline{U}'\overline{\varphi}} \bmod q$ in terms of *step*6) B (2), they obtain $\bar{k}'_c = H(\overline{U}'h + SKg^{\varphi}/\varphi \bmod q)$ which is equal to k_c with a negligible probability $1/q$

Far from the authenticity objective, an attractive security property can also be achieved as below.

Theorem 6: Taking advantage of a known hash value h, a malicious user is not able to refrain the real file m from uploading.

Proof: Without the knowledge of m, the adversary is only capable of forging a label $\overline{TF}' = (\overline{tf}'_{maj}, \overline{tf}'_{sub})$ the same as that of Theorem 4. Therefore, even if he outsourced a counterfeit of the ciphertext, the valid label $TF = (tf_{maj}, tf_{sub})$ uploaded by a subsequent user will not be coincide with $\overline{TF}'$ thus a real copy can be outsourced independently.

Then we consider the circumstance that the file is explicit to the cloud and testify the user's privacy according to Lemma 2.

Theorem 7: Colluding with less than t, the cloud is unable to determine which ciphertext corresponds to a given file $\overline{m}$

Proof: Assume that $\overline{m} = m$ and $c = E_{k_c}(m)$, the cloud is capable of launching a request in terms of the correct file. However, since only $t' < t$ key servers will cooperate with him, the de-duplication label he obtained would be $\overline{TF} = (\overline{tf}_{maj}, \overline{tf}_{sub})$, where $\overline{tf}_{maj} = g^{Uh+\overline{SK}'(g^{U\overline{\varphi}}/\overline{\varphi})}$ and $\overline{tf}_{sub} = g^{U\overline{\varphi}}/\overline{\varphi}$ According to Lemma 2,it is obvious that there is at most $1/p$ possibility for $\overline{SK}' = \sum_i^n SK_i$ even if U is accurately generated. Therefore, after he computes $tf_{maj}/\bar{t}f_{maj} = g^{SKg^{U\varphi/\varphi}-\overline{S}K'g^{U\overline{\varphi}/\overline{\varphi}}} \bmod q$, $PK^{(}tf_{sub} - \overline{tf}_{sub}) = g^{SK(g^{U\varphi/\varphi}-g^{U\overline{\varphi}}/\overline{\varphi})} \bmod q$ will negligibly be congruential with it. Since the concurrent key is derived from the label, it also means that the cloud is not able to decrypt the corresponding ciphertext.

To sum up, the user's privacy can be protected because, if the file is not available or less than t key servers collude with him even when the file is known, the cloud is unable to inspect any duplication, not to mention decrypting the ciphertexts. As for the authenticity, a malicious user is incapable of downloading or decrypting any unauthorized file or even refrain the file from uploading if he is not provided with it beforehand.

VII. PERFORMANCE ANALYSIS

In this section,to demonstrate the efficiency of the scheme,we choose two schemes [4], [5] that based on the blind signature for comparison. And We also compare this scheme with many other schemes that are not based on blind signatures.In our performance evaluation, for the convenience of discussion, some notations are introduced, we denote by *Exp* the modular exponentiation, by *Hash* the convergent encryption, by n the total number of K-CSPs, and by t the number of K-CSPs participated in the convergent key creation. Please note that we omitted the general file transfer and file encryption/decryption modules for simplification. Besides, the file downloading process only has symmetrical encryption that is very efficient. It will not influence the efficiency of the system, so we omit it in our discussion.

Our experimental environment was a computer with Intel(R) Core(TM) i5-3470 CPU processor running at 3.2GHz and 8G memory running Linux. Here we set the reliability level $n - t = 2$, the quantity of the file in the data upload from 10 to 70.

A. SET SYSTEM COST

In the establishment phase of the system, the primary computation cost is dominated by modular exponentiation operations, which are used to generate public/private pairs for each K-CSP and the system public/private pair. In Miao [4] scheme, each K-CSP needs $n + 2$ times exp operations to publish its public and private keys by interacting with other K-CSPs; In Dupless [5] scheme, because a single server is used to build a system, we ignore this comparison; In our scheme, each K-CSP only requires three times exp operations for publishing its public and private keys by interacting with other K-CSPs. In the experiment, we set the number of K-CSPs from 5 to 10, as shown in Figure.2.

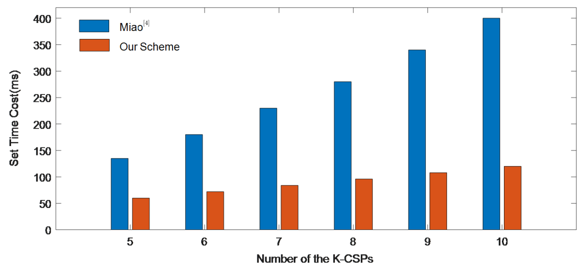

FIGURE 2. Set system cost.

B. FILE UPLOAD COST

In the file upload phase, the user generates the convergent key by interaction with K-CSPs, the time cost includes all signatures and labels, we denote m by the numbers of the uploaded file, it ranges from 10 to 70. Please note that we only consider the situation of the first upload.When the number of key servers n = 10,in the Miao [4] scheme,the total computation consists of $2t + 1$ times *Exp* and $2m$ times *Hash*;In the Dupless [5] scheme,the total computation consists of 2 times *Exp* and $3m$ times *Hash*;In our scheme,the total computation consists of $t + 6$ times *Exp* and m times *Hash*,as is shown Figure.3.

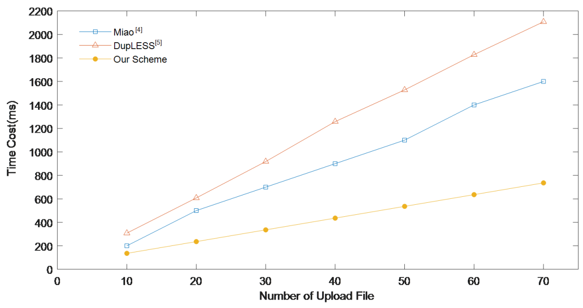

FIGURE 3. Upload file cost.

C. COMPARED TO OTHER SCHEMES

Then, we compare the performance of our solution with other major technologies. For the third party's necessity, de-duplication level, the necessity of participation, the necessity of key fusion and other functions, see Table 3 below for comparison.

TABLE 3. Computation overheads for De-duplication.

Schemes	Technology	TTP	Deduplication Level	Per-form Ob-ject
scheme [43]	BL-MLE+Pow	—-	Block	Single User
scheme [44]	Authentication Protocol+ Authorization Detection	Private Cloud Server	File	Multiple Users
scheme [45]	Attribute Encryption+ Random	Sampling At-tribute Certifi-cation Center	Block	Multiple Users
our scheme	Threshold Blind Signature+ Verifiable Secret Sharing	Multiple Key Server	File	Single User

Now, we compare the computational overhead for PoW, respectively on client, third-party and cloud side. The results are shown in Table 4.

TABLE 4. Computation overheads for PoW.

Schemes	Client	TTP	Cloud
scheme [43]	$O(b)Hash + O(b)$	—-	$O(f)Add$
scheme [44]	$O(f) + O(kL)$	—-	$O(kLf)Add$
scheme [45]	$O(f)Hash + O(N \log N)$	$O(N)CE\_K$	$O(kLf)Add$
our scheme	$O(f)Hash + O(kL)$	$O(N)CE\_K$	$O(kLf)Add$

In order to achieve security and resist violent dictionary attack, this scheme has no obvious advantage over the one without blind signature technology in terms of time cost, but its security is guaranteed.Compared with the same scheme using blind signature technology, this scheme is more efficient and more secure.

VIII. CONCLUSION

Aiming at the violent dictionary attack, we proposed a conspiracy-free data de-duplication protocol based on a threshold blind signature in this article. We argued that the scheme [4], [5] is defective in both security and efficiency aspects. So we use CRT secret sharing, blind signature, and bloom filter to construct the scheme, the experimental results show that the calculation cost of the systems establishment and files upload is relatively small. And we used homomorphism computation to aggregate and generate partial signature tags, and introduced a secret sharing mechanism based on CRT to hide signature keys, thus balancing the security concerns of cloud and client.

REFERENCES

[1] C. G. C. Index, "Forecast and methodology, 2015–2020 white paper," *Retrieved 1st June*, p. 15, Jun. 2016.

[2] X. Liu, R. Zhu, B. Jalaian, and Y. Sun, "Dynamic spectrum access algorithm based on game theory in cognitive radio networks," *Mobile Netw. Appl.*, vol. 20, no. 6, pp. 817–827, Dec. 2015.

[3] W. Li, W. Meng, and M. H. Au, "Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments," *J. Netw. Comput. Appl.*, vol. 161, Jul. 2020, Art. no. 102631.

[4] M. Miao, J. Wang, H. Li, and X. Chen, "Secure multi-server-aided data deduplication in cloud computing," *Pervas. Mobile Comput.*, vol. 24, pp. 129–137, Dec. 2015.

[5] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Server-aided encryption for deduplicated storage," in *PProc. 22nd USENIX Secur. Symp. USENIX Secur.*, 2013, pp. 179–194.

[6] J. R. Douceur, A. Adya, W. J. Bolosky, P. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in *Proc. 22nd Int. Conf. Distrib. Comput. Syst.*, Jul. 2002, pp. 617–624.

[7] M. Li, C. Qin, and P. P. Lee, "Cdstore: Toward reliable, secure, and cost-efficient cloud storage via convergent dispersal," in *Proc. USENIX Annu. Tech. Conf. USENIX ATC*, 2015, pp. 111–124.

[8] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, Springer, 2013, pp. 296–312.

[9] J. Li, C. Qin, P. P. C. Lee, and J. Li, "Rekeying for encrypted deduplication storage," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2016, pp. 618–629.

[10] C. Qin, J. Li, and P. P. C. Lee, "The design and implementation of a rekeying-aware encrypted deduplication storage system," *ACM Trans. Storage*, vol. 13, no. 1, pp. 1–30, Mar. 2017.

[11] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proc. 18th ACM Conf. Comput. Commun. Secur. CCS*, 2011, pp. 491–500.

[12] S. Wan and S. Goudos, "Faster R-CNN for multi-class fruit detection using a robotic vision system," *Comput. Netw.*, vol. 168, Feb. 2020, Art. no. 107036.

[13] C. Chen, J. Hu, T. Qiu, M. Atiquzzaman, and Z. Ren, "CVCG: Cooperative V2 V-aided transmission scheme based on coalitional game for popular content distribution in vehicular ad-hoc networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 12, pp. 2811–2828, Dec. 2019.

[14] S. Wan, Y. Xia, L. Qi, Y.-H. Yang, and M. Atiquzzaman, "Automated colorization of a grayscale image with seed points propagation," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1756–1768, Jul. 2020.

[15] S. Ding, S. Qu, Y. Xi, and S. Wan, "A long video caption generation algorithm for big video data retrieval," *Future Gener. Comput. Syst.*, vol. 93, pp. 583–595, Apr. 2019.

[16] S. Ding, S. Qu, Y. Xi, and S. Wan, "Stimulus-driven and concept-driven analysis for image caption generation," *Neurocomputing*, vol. 398, pp. 520–530, Jul. 2020.

[17] S. Wan, X. Li, Y. Xue, W. Lin, and X. Xu, "Efficient computation offloading for Internet of vehicles in edge computing-assisted 5G networks," *J. Supercomput.*, pp. 1–30, Oct. 2019.

[18] C. Chen, L. Liu, T. Qiu, K. Yang, F. Gong, and H. Song, "ASGR: An artificial spider-Web-based geographic routing in heterogeneous vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1604–1620, May 2019.

[19] X. Deng, J. Li, E. Liu, and H. Zhang, "Task allocation algorithm and optimization model on edge collaboration," *J. Syst. Archit.*, vol. 110, Nov. 2020, Art. no. 101778.

[20] S. Wan, Y. Zhao, T. Wang, Z. Gu, Q. H. Abbasi, and K.-K.-R. Choo, ''Multi-dimensional data indexing and range query processing via Voronoi diagram for Internet of Things,'' *Future Gener. Comput. Syst.*, vol. 91, pp. 382–391, Feb. 2019.
[21] R. Zhu, X. Zhang, X. Liu, W. Shu, T. Mao, and B. Jalaian, ''ERDT: Energy-efficient reliable decision transmission for intelligent cooperative spectrum sensing in industrial IoT,'' *IEEE Access*, vol. 3, pp. 2366–2378, 2015.
[22] S. Wan, Z. Gu, and Q. Ni, ''Cognitive computing and wireless communications on the edge for healthcare service robots,'' *Comput. Commun.*, vol. 149, pp. 99–106, Jan. 2020.
[23] X. Deng, F. Long, B. Li, D. Cao, and Y. Pan, ''An influence model based on heterogeneous online social network for influence maximization,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 737–749, Apr. 2020.
[24] T.-T. Goh, Z. Xin, and D. Jin, ''Habit formation in social media consumption: A case of political engagement,'' *Behav. Inf. Technol.*, vol. 38, no. 3, pp. 273–288, Mar. 2019.
[25] D. Jin, S. Shi, Y. Zhang, H. Abbas, and T.-T. Goh, ''A complex event processing framework for an adaptive language learning system,'' *Future Gener. Comput. Syst.*, vol. 92, pp. 857–867, Mar. 2019.
[26] J. Xiong, Y. Zhang, L. Lin, J. Shen, X. Li, and M. Lin, ''Ms-PoSW: A multi-erver aided proof of shared ownership scheme for secure deduplication in cloud,'' *Concurrency Comput., Pract. Exper.*, vol. 32, no. 3, Feb. 2020, Art. no. e4252.
[27] J. Tang, B. Zhang, Y. Zhou, and L. Wang, ''An energy-aware spatial index tree for multi-region attribute query aggregation processing in wireless sensor networks,'' *IEEE Access*, vol. 5, pp. 2080–2095, 2017.
[28] Y. Zhao, H. Li, S. Wan, A. Sekuboyina, X. Hu, G. Tetteh, M. Piraud, and B. Menze, ''Knowledge-aided convolutional neural network for small organ segmentation,'' *IEEE J. Biomed. Health Informat.*, vol. 23, no. 4, pp. 1363–1373, Jul. 2019.
[29] C. Chen, L. Liu, T. Qiu, D. O. Wu, and Z. Ren, ''Delay-aware grid-based geographic routing in urban VANETs: A backbone approach,'' *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2324–2337, Dec. 2019.
[30] P. Zhao, J. Li, F. Zeng, F. Xiao, C. Wang, and H. Jiang, ''ILLIA: Enabling k -anonymity-based privacy preserving against location injection attacks in continuous LBS queries,'' *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1033–1042, Apr. 2018.
[31] J. Wang, J. Luo, X. Liu, Y. Li, S. Liu, R. Zhu, and A. Anjum, ''Improved Kalman filter based differentially private streaming data release in cognitive computing,'' *Future Gener. Comput. Syst.*, vol. 98, pp. 541–549, Sep. 2019.
[32] L. Li, T.-T. Goh, and D. Jin, ''How textual quality of online reviews affect classification performance: A case of deep learning sentiment analysis,'' *Neural Comput. Appl.*, vol. 32, no. 9, pp. 4387–4415, May 2020.
[33] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, and Q. He, ''Efficient query of quality correlation for service composition,'' *IEEE Trans. Services Comput.*, early access, Apr. 27, 2018, doi: 10.1109/TSC.2018.2830773.
[34] C. Wang, G. Liu, H. Huang, W. Feng, K. Peng, and L. Wang, ''MIASec: Enabling data indistinguishability against membership inference attacks in MLaaS,'' *IEEE Trans. Sustain. Comput.*, vol. 5, no. 3, pp. 365–376, Jul. 2020.
[35] L. Qi, X. Zhang, S. Li, S. Wan, Y. Wen, and W. Gong, ''Spatial-temporal data-driven service recommendation with privacy-preservation,'' *Inf. Sci.*, vol. 515, pp. 91–102, Apr. 2020.
[36] Z. Lv, X. Li, H. Lv, and W. Xiu, ''BIM big data storage in WebVRGIS,'' *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2566–2573, Apr. 2020.
[37] Z. Lv, W. Kong, X. Zhang, D. Jiang, H. Lv, and X. Lu, ''Intelligent security planning for regional distributed energy Internet,'' *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3540–3547, May 2020.
[38] Z. Lv, B. Hu, and H. Lv, ''Infrastructure monitoring and operation for smart cities based on IoT system,'' *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1957–1962, Mar. 2020.
[39] J. Jia, Q. Ruan, Y. Jin, G. An, and S. Ge, ''View-specific subspace learning and re-ranking for semi-supervised person re-identification,'' *Pattern Recognit.*, vol. 108, Dec. 2020, Art. no. 107568.
[40] Z. Gao, Y. Li, and S. Wan, ''Exploring deep learning for view-based 3D model retrieval,'' *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 1, pp. 1–21, Apr. 2020.
[41] J. Blasco, R. Di Pietro, A. Orfila, and A. Sorniotti, ''A tunable proof of ownership scheme for deduplication using Bloom filters,'' in *Proc. IEEE Conf. Commun. Netw. Secur.*, Oct. 2014, pp. 481–489.
[42] J. Xiong, Y. Zhang, F. Li, S. Li, J. Ren, and Z. Yao, ''Research progress on secure data deduplication in cloud,'' *J. Commun.*, vol. 37, no. 11, pp. 169–180, 2016.
[43] R. Chen, Y. Mu, G. Yang, and F. Guo, ''BL-MLE: Block-level message-locked encryption for secure large file deduplication,'' *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2643–2652, Dec. 2015.
[44] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, ''A hybrid cloud approach for secure authorized deduplication,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1206–1216, May 2015.
[45] B. Dan, A. Sahai, and B. Waters, ''Functional encryption: Definitions and challenges,'' in *Proc. Theory Cryptogr. Conf.*, 2011, pp. 253–273.

BO MI was born in 1982. He received the Ph.D. degree in computer system architecture from Chongqing University, China, in 2009. Since 2011, he has been an Associate Professor with the College of Information Science and Engineering, Chongqing Jiaotong University, China. His current research interests include intelligent transportation, vehicular *ad hoc* networks, and cryptography.

YANG LI was born in 1995. He received the B.S. and M.S. degrees in computer science and technology from Chongqing Jiaotong University, China. His current research interests include vehicular *ad hoc* networks and cryptography.

HUANG DARONG (Member, IEEE) was born in 1978. He received the B.S. degree in applied mathematics from the Hubei National Institute, Hubei, China, in 2000, the M.S. degree in applied mathematics from Liaoning University, Liaoning, China, in 2003, and the Ph.D. degree in control theory and control engineering from Chongqing University, Chongqing, China, in 2006. Since 2011, he has been a Professor with the College of Information Science and Engineering, Chongqing Jiaotong University, Chongqing. His research interests include fault diagnosis and fault-tolerant control of dynamic systems, analysis and design of complex systems, big data analytic of transport systems, reliability engineering, and so on.

TIANCHENG WEI was born in 1993. He received the B.S. degree in computer science and technology from the Anhui University of Technology, Anhui, China, in 2017, and the M.S. degree in computer science and technology from Chongqing Jiaotong University, China, in 2020. His current research interests include vehicular *ad hoc* networks and cryptography.

QIANQIAN ZOU was born in 1996. She received the bachelor's degree from the School of Information and Safety Engineering, Zhongnan University of Economics and Law. She is currently pursuing the master's degree in computer science and technology with the Zhongnan University of Economics and Law, focusing on edge computing.