1-2018

Consortium blockchain-based SIFT: Outsourcing encrypted feature extraction in the D2D network

Xiaoqin FENG

Jianfeng MA

Tao FENG

Yinbin MIAO

Ximeng LIU
Singapore Management University, xmliu@smu.edu.sg

Received July 31, 2018, accepted September 5, 2018, date of publication September 13, 2018, date of current version October 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869856

Consortium Blockchain-Based SIFT: Outsourcing Encrypted Feature Extraction in the D2D Network

XIAOQIN FENG[1], JIANFENG MA[1,2], (Member, IEEE), TAO FENG[3], YINBIN MIAO[1,4], AND XIMENG LIU[5,6], (Member, IEEE)

[1]Shaanxi Key Laboratory of Network and System Security and the School of Cyber Engineering, Xidian University, Xi'an 710071, China
[2]School of Computer, Xidian University, Xi'an 710071, China
[3]School of Computer and Communications, Lanzhou University of Technology, Lanzhou 730050, China
[4]Key Laboratory of Optical Communication and Networks, Chongqing 4000565, China
[5]School of Information Systems, Singapore Management University, Singapore 188065
[6]College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

Corresponding author: Jianfeng Ma (jfma@mail.xidian.edu.cn)

This work was supported in part by the Key Program of NSFC under Grant U1405255, in part by the Shaanxi Science & Technology Coordination & Innovation Project under Grant 2016TZC-G-6-3, in part by the National High Technology Research and Development Program (863 Program) under Grant 2015AA016007, in part by the Fundamental Research Funds for the Central Universities under Grant SA-ZD161504, in part by the Major Nature Science Foundation of China under Grant 61370078 and Grant 61309016, in part by the National Natural Science Foundation of China under Grant 61702404, in part by the China Postdoctoral Science Foundation Funded Project under Grant 2017M613080, in part by the Fundamental Research Funds for the Central Universities under Grant JB171504, and in part by the Research on Secure Storage and Privacy Diagnosis Mechanism of Cloud Outsourcing Personal Health File under Grant 61702105.

ABSTRACT Privacy-preserving outsourcing algorithms for feature extraction not only reduce users' storage and computation overhead but also preserve the image privacy. However, the existing schemes still suffer from deficiencies induced by security, applications, efficiency and storage. To solve the problems, we implement a consortium chain-based outsourcing feature extraction scheme over encrypted images by using the smart contract, distributed autonomous corporation (DAC), sharding technique, and device to device (D2D) communication, which is secure, widely applied, highly efficient, and has less storage overhead. First, the effectiveness, security, and performance of our scheme are analyzed. Then, the efficiency and storage overhead of our scheme are presented by conducting experimental results.

INDEX TERMS Outsourcing feature extraction, consortium chain, smart contract, DAC, sharding technique, D2D communication.

I. INTRODUCTION

D2D (Device to Device) communication can meet the needs for a variety of emerging business [1], such as advertising push, information sharing and large event data sharing. The advent of big data era has led to a surge in the number of images, documents and videos [2], and data owners prefer to outsource tasks [3], [4] for large-scale images feature extraction to the clouds because of limited local resources. However, simple feature extraction on image plaintext [5]–[8] will no longer meet people's needs due to the image information leakage [9]. The conventional schemes [10]–[13] on plaintext domain cannot assure the image privacy in the clouds. Hence, feature extraction methods [14]–[17] on ciphertext domain are proposed. A new approach in [18] designed a multi-servers cloud structure with a tailored practical security [19]–[21] to protect the image contents. Although various image feature detection tasks including the global and local features can be performed, the approach still has poor effectiveness (feature effects in terms of feature accuracy) compared with SIFT [22]. To gain the same feature effects as SIFT, a privacy-preserving implementation of homomorphic encryption (SHE) based SIFT method [23] was proposed, but this scheme incurs poor efficiency in terms of image feature extraction and only resists pure ciphertext and known plaintext attacks. To avoid the efficiency limitation caused by SHE, SecSIFT (Securing SIFT) [24] allocated the computing process of SIFT to a group of independent and cooperative cloud servers, which achieves high efficiency. However, these methods are either scarce of effectiveness, or worse of security and efficiency. To preserve the image privacy and feature accuracy in SIFT, and comply better security and efficiency, Hu *et al.* [25] first designed two protocols for secure multiplication (BSCP) and comparison (BSMP), then developed a practical approach using the SHE integrated with

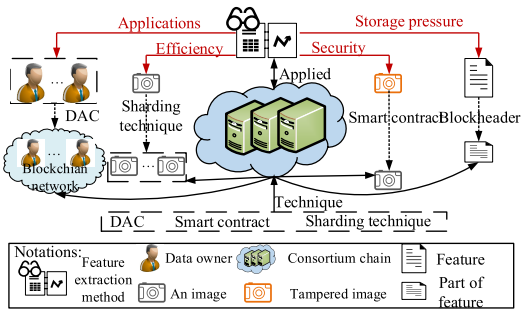

FIGURE 1. Deficiencies in the existing methods, and our solutions.

the latest batching technique Single-Instruction Multi-Data (SIMD). Unfortunately, the existing schemes still have some drawbacks, such as unsafety, limited application ranges (e.g. the clouds can only trade with a few of users), low efficiency (e.g. only one image is handled by servers during the same time) and large storage pressure as showed in FIGURE 1.

To the best of our knowledge, the blockchain can be improved with the smart contract [26], DAC (Distributed Autonomous Corporation), sharding technique and D2D communication, and flexibly applied into the feature extraction method in [25] to solve these problems [27]. To be able to execute calculations and deal with the above problems, we propose a consortium blockchain-based outsourcing feature extraction method over encrypted images (called CB-SIFT).

The main contributions of the CB-SIFT scheme are demonstrated as follows:

- **Safety.** Consensus among communities ensures the reliability of image features with keeping feature effects in SIFT, and the features and smart contract are stored in blockchain, which solve the privacy leaking problem of image features (e.g. tempering, forging by the semi-trusted clouds).
- **Wide application ranges.** Feature extraction information of clouds is published on the blockchain network in the form of DACs' (servers) pre-set rules. The users find the matching DACs by filtering on the Internet, which enables wider application ranges and less communication costs.
- **High efficiency.** With the sharding technique, different communities of servers can concurrently perform the feature extraction for different images to improve the approach efficiency. Besides, the D2D communication are applied into the consortium chain to enhance the communication efficiency among the servers and reduce the communication costs.
- **Less storage pressure.** We specialize the bolckheader contents so that each server only needs to store the block-header chain [28], and a few of blocks to alleviate the storage pressure as well as keep the blockchain integrity.

II. PRELIMINARIES

Before describing our design, we first introduce the blockchain and other related techniques (e.g. smart contract, DAC, sharding technique and D2D communication).

A. BLOCKCHAIN

The blockchain [29]–[31] is a distributed ledger, which is maintained by the network nodes through competitive computing based on the consensus mechanism [32]. There are no centralized management organizations, and the data exchange is verified by the digital signature technology. Besides, each tamper-resistant transaction in the blockchain is cryptographically linked to two adjacent blocks.

The blockchain is divided into the public chain, consortium chain and private chain by the network decentralization degree. It is worth noticing that the consortium chain targets on a certain group, and a plurality of preselected nodes are designated as consensus nodes. To comply better performance [33], it has certain requirements for the configuration and network environment of the consensus nodes.

B. SMART CONTRACT

The smart contract is a computer protocol designed to disseminate, validate, or execute contracts in an information-based manner, and it allows trusted trading without the third parties. The smart contract automatically operates once the pre-set conditions are triggered. For example, whether the users have paid for enough money, does the extracted image feature meet the criteria, is the feature extraction fulfilled within the allocated time. The purpose of smart contract is to provide a better security approach than the traditional contract and to reduce other trading costs associated with it.

C. DAC

The DAC has enough CPU (Central Processing Unit), memory and an IP (Internet Protocol) address. It automatically operates without human intervention, and manages itself through a series of open and fair rules (e.g. the money costs, feature effects, time costs, and reward mechanism for the DAC shareholders and its running) which appear in the form of open source software. The DAC shareholders acquire shares [34] by purchasing shares or providing services, and they are able to share the organization's revenue and participate in the system operation. Bitcoin and other virtual currencies are typical DACs. Each bitcoin participant can share the benefits of bitcoin growth, and the global bitcoin community is open to anyone. The self-regulation rules of the DAC are explained as follows:

- Participants of the DAC system are its self-automation units.
- The units are trustless, and they only consider their own interests in the DAC system.
- The self-government units will not influence each other if they do not trust each other.

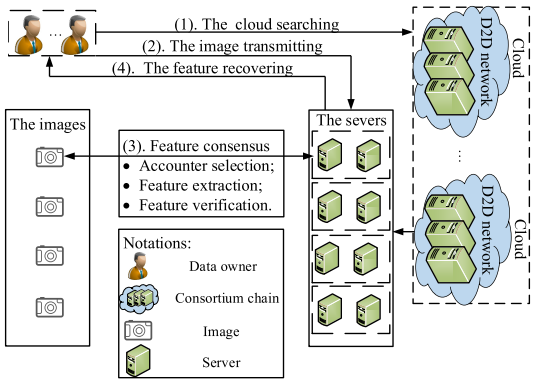

FIGURE 2. The system model.

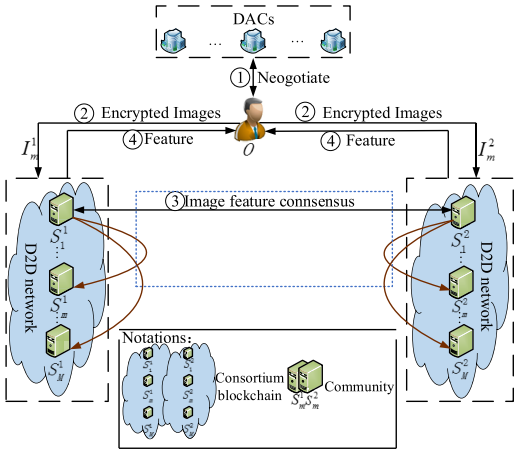

FIGURE 3. The operation process of the CB-SIFT scheme.

D. SHARDING TECHNIQUE

The sharding technique can be used to enhance the blockchain's scalability [35] and efficiency. The network nodes are demanded to save the discriminated blocks state and manage diverse transactions in the blockchain system. The nodes are usually assigned to different communities, and the handling process of the transactions are carried out in a parallel way among the communities, which greatly improves the system efficiency.

E. D2D COMMUNICATION

D2D communication technology refers to a communication method in which two peer nodes communicate directly. Firstly, user's data is directly transmitted between the terminals, which avoids its transmission in the cellular communication through the network transfers, thereby generating the link gain. Secondly, resources between the D2D users and the D2D cells can be restored. Therefore, the resource multiplexing gain can be generated. Finally, the link gain and the resource multiplexing gain can improve the efficiency of the wireless spectrum resources, thereby improving the network throughput.

After the techniques related to the CB-SIFT scheme are explained, we elaborate the problem formulation of CB-SIFT scheme in the next section.

III. PROBLEM FORMULATION

In this section, we demonstrate the problem formulation of our design including the system model, system overview and designing goals, and the detailed constructions will be provided in the next section.

A. SYSTEM MODEL

As the system model presented in the FIGURE 2, the user trades with the cloud for feature extraction of encrypted images, and the complete process includes four stages:

(1) The user looks for the clouds meeting his requirements;
(2) the user uploads the encrypted images to the cloud after both of the user and cloud catch an agreement;

TABLE 1. Notations in the CB-SIFT scheme.

Notations	Descriptions	Notations	Descriptions
B_j	The j-th block	O	Data owner
I_m^i	Image	S_m^i	Sever
V_m^i	Feature vector	A_c	Accounter
V_{vm}^i	Valid V_m^i	V^m	Final image feature
$\bar{V}_{vm}^i$	Main feature vector	$\bar{B}$	Blocks number of a sever
L^i	Laplace operator	$*$	Convolution operation
G	Gaussian function	D^i	DoG operator
H_m^i	Hessian matrix	Det_m^i	Value of H_m^i
m_m^i	Gradient magnitude	Θ_m^i	Gradient direction
$\mathcal{H}_m^i$	Gradient histogram	C_m	Community
HS^j	Hash value of block	P_{cm}^i	Candidate keypoints
P_{sm}^i	Stable keypoints	Nb_m^i	Number of bins

(3) the cloud servers carry out the feature extraction method to obtain the encrypted image features;
(4) and the cloud returns the feature results to the user and the user decrypts them to gain the final image features.

We improve the stages (1), (2) and (3) by applying the DAC, sharding technique, D2D communication and smart contract into the blockchian to construct a widely applied, highly efficient and more secure CB-SIFT scheme. In addition, to alleviate storage pressure of the servers and abandon the useless information, we specially designate the blcok-header contents such that the integrity of feature data can be guaranteed by storing the blockheader chain and some number of blocks.

B. SYSTEM OVERVIEW

We outline the CB-SIFT method which is shown in the FIGURE 3 in this section. TABLE 1 describes the notation definitions in the CB-SIFT scheme, and note that the servers serve as nodes in the consortium chain of our design. Taking the blocks B_j and B_{j+1} as an example, the CB-SIFT scheme is demonstrated as follows:

- **Consortium chain filtering.** The servers act as DACs to issue their feature-related information on the blockchain

network, and users search for the appropriate DACs online.

- **Transmitting the encrypted images.** After the two sides agree on the deal, they create a smart contract and publish it into the blockchain. The consensus servers $(S_m^i(i = 1, 2)(m = 1, \ldots, M))$ in the consortium chain are randomly divided into different communities with two servers in each community. O divides each encrypted image into two sections I_m^1 and I_m^2, and they are transmitted to the consortium chain at once. Each server among the communities is responsible for the feature extraction of certain encrypted image section with the relationship $I_m^i \longleftrightarrow S_m^i$.
- **Feature consensus.** S_m^i executes the feature extraction for I_m^i using the method in [25]. The V_m^1 and V_m^2 are separately recorded by A_c chosen through the PoW (Proof-of-Work) [36] consensus mechanism among all communities, and together with the calculation values are sent to the certain servers in other communities for verification with the correspondence $S_m^i \leftrightarrow S_m^i(m = 1, \ldots, m-1, m+1, \ldots, M)$. With the related values, $S_m^1(m = 1, \ldots, m-1, m+1, \ldots, M)$ and $S_m^2(m = 1, \ldots, m-1, m+1, \ldots, M)$ cooperatively perform the feature extraction process as $S_m^i(i = 1, 2)$ to verify V_m^i, and notice that $V_{vm}^i(i = 1, 2)$ is recorded by A_c.
- **Feature conforming.** A_c sends the recorded $V_{vm}^i(i = 1, 2)$ to O, and O decrypts V_{vm}^1 and V_{vm}^2 to get V^m. If the feature effects and spent time related to V^m meet the conditions in the smart contract, the smart contract is triggered. Then $\bar{V}_{vm}^i(i = 1, 2)$ which contains the descriptors of main direction in $V_{vm}^i(i = 1, 2)$, the related parameters and smart contract are written into the blockheaders of B_j and B_{j+1}. The V_{vm}^1 and V_{vm}^2 are separately written into the blockbodies of B_j and B_{j+1} by A_c. Each sever saves the blockheader chain and some blocks. Besides, the servers will be rewarded with the pre-set rules in the DACs.

C. DESIGNING GOAL

The ultimate goal of this work is to design a secure and tamper-resistant, widely applied, efficient and trustless privacy-preserving outsourcing image feature extraction system. To this end, we have the following specific designing goals.

- **Security and tamper-resistance constraint:** Apart from the image contents preserving, the processed and stored image feature data should be prevented from tampering by the cloud servers.
- **Application constraint:** The users should easily and quickly find the outsourcing clouds and the servers can serve for more users without incurring much communication overhead.
- **Efficiency constraint:** The feature extraction and contract signing should have less time overhead. Moreover, the communication efficiency among the servers should be as high as possible, and less communication costs should be spent.
- **Trust constraint:** There are no need for trust between the users and servers, and among the servers.

IV. THE CB-SIFT SCHEME: OUR SYSTEM CONSTRUCTION

Based on the feature extraction method in [25], we construct the CB-SIFT scheme by using the smart contract, DAC and sharding technique, which has the same notation definitions. Before presenting the concrete construction of CB-SIFT scheme, we first review the outsourcing feature extraction method in [25].

A. OUTSOURCING FEATURE EXTRACTION METHOD IN [25]

The SIFT is a computer vision algorithm used to detect and describe the local features in images. It searches for extreme points in different spatial scales, and extracts their positions, scales and rotation invariants. Based on the SIFT which is integrated with the SHE and SIMD, Hu *et al.* proposed an approach to realize the feature extraction by two severs over an encrypted outsourcing image. The two servers handle the different parts of an image, and co-work with the related calculations and numerical comparisons during feature extraction through the BSMP and BSCP.

The execution steps of the approach in [25] are described as follows:

- **Image encryption.** O generates a random matrix $I_m^2(x, y)$ and obtains $I_m^1(x, y)$ by encrypting $I_m(x, y)$ through the formula $I_m^1(x, y) = I_m(x, y) + I_m^2(x, y)$. $I_m^1(x, y)$ and $I_m^2(x, y)$ are separately sent to S_m^1 and S_m^2.
- **Keypoint localization.** Let the symbol σ be the scale in the scale space. By the formulas $L^i(x, y, \sigma) = G(x, y, \sigma) * I_m^i(x, y)$ and $D^i(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I_m^i(x, y)$, Gaussian scale space (LoG) and Difference-of-Gaussian scale space (DoG) of the image are described. Each sample point in DoG is compared with its 26 neighbors by the BSCP to find the local extreme points, and their edge responses are eliminated by computing $H_m^i(x, y, \sigma)$, and comparing $Det_m^1(x, y, \sigma)$ and $Det_m^2(x, y, \sigma)$ with the BSMP and BSCP. Finally, the stable extreme points P_{sm}^i are gained.
- **Orientation assignment.** $L^i(x, y)$ is the scale space value of the keypoint. Through the formulas (1) and (2), the servers collect $m_m^i(x, y)$ and $\Theta_m^i(x, y)$ of pixels which are distribute in 3σ neighborhood window around the keypoint (1) and (2), as shown at the bottom of the next page. The histogram $\mathcal{H}_m^i$ is used to count $m_m^i(x, y)$ and $\Theta_m^i(x, y)$ of the Gaussian-smoothed pixels. $\mathcal{H}_m^i$ divides the $0° - 360°$ direction range into 36 bins, and each bin is $10°$. Implementing the BSCP on the accumulated gradient magnitudes among all bins in $\mathcal{H}_m^1$ and $\mathcal{H}_m^2$, the highest peak is found, which acts as the main direction of the keypoint. The directions within the 80% of the highest peak are the auxiliary directions of the keypoint, and can also be found through the BSCP.

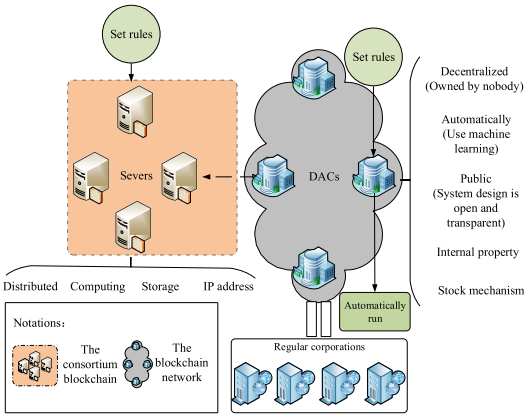

FIGURE 4. The operation principles of DACs.

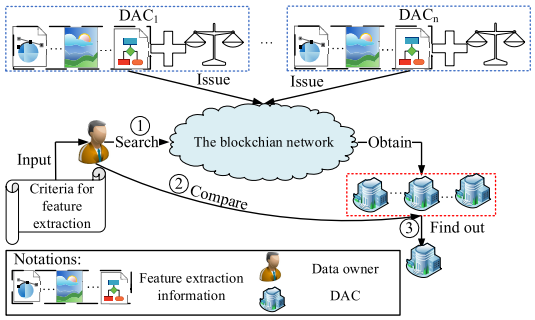

FIGURE 5. The user's filtering process for the DAC.

At this point, the information(x, y, Θ) about the keypoint position, scale, and direction is owned.

- **Descriptor generation.** S_m^i generates the descriptor for each keypoint. The neighboring area in LoG of the keypoint is divided into 4×4 sub-areas, and each one serves as a seed point. The sub-area size is the same condition as the keypoint orientation assignment, that is, 3σ sub-pixels. S_m^i generates a gradient histogram in each sub-area which has 8 directions and is weighted by the same Gaussian window, and calculates the gradient magnitude eventually accumulated in each direction. Finally, the $4 \times 4 \times 8 = 128$ gradients information is gained to be compressed into V_m^i of the keypoint.
- **Image feature obtaining.** V_m^i is transmitted to O who computes the final image feature by the formula $V^m = V_m^1 - V_m^2$.

B. CB-SIFT SCHEME

This section elaborates the details of the CB-SIFT scheme, and it contains four main processes: consortium chain filtering, transmitting the encrypted images, feature consensus, and feature conforming.

1) CONSORTIUM CHAIN FILTERING

FIGURE 4 describes the operation principles of DACs. The consensus servers within the consortium chain collectively share and invest in the DACs whose rules are same and distributed on these stakeholders in the form of open source software. The DACs are owned by nobody and no one can modify their rules once the rules are set by their owners, only if the majority of stakeholders reached a consensus. The entire blockchain network comprised of the consortium chains and users is distributed, and the pre-outsourcing stage in the system only involves the preparation works of both the users and consortium chains. FIGURE 5 describes the user's filtering process for the DAC, and the working mode of the DAC and filtering mode of the user include the following two aspects:

- For one thing, the DACs issue their feature extraction information on the blockchain network. More specifically, they public the information about price, feature extraction effects and time for an image feature extraction.
- For another thing, the users can search for the proper DACs on the Internet, and compare them to select the most satisfying one. Alternatively, the user can get the specific DAC through inputting the filtering criteria for feature extraction. Eventually, it outsources its encrypted images to the required DAC.

2) TRANSMITTING THE ENCRYPTED IMAGES

FIGURE 6 depicts the working status of smart contract. After O selects the appropriate DAC and both sides reach an agreement on the deal, they will create a smart contract, and the consortium chain contained the DAC starts working. Besides, the details of severs (i.e. capability, reputation, etc.) and legal provisions are listed on the smart contact. The conditions of the smart contract include: whether the smart contract receives enough money from O, whether the effects of extracted feature meet the user's standards, and whether the servers have completed the feature extraction within the allocated time T_{cb}. In addition, the smart contract will be published into the blockchain.

FIGURE 7 shows the sharding technique in our design. According to the sharding technique, S_m^i in the consortium chain is randomly divided into different communities containing two servers by the function $F(S_m^i) = C_m(i = 1, 2)(m = 1, \ldots, M)$ before the image transmission, and the images are managed and concurrently executed for feature

$$m_m^i(x, y) = \sqrt{(L^i(x+1, y) - L^i(x-1, y))^2 + (L^i(x, y+1) - L^i(x, y-1))^2}. \quad (1)$$

$$\Theta_m^i(x, y) = tan^{-1}((L^i(x+1, y) - L^i(x-1, y))/(L^i(x, y+1) - L^i(x, y-1))). \quad (2)$$

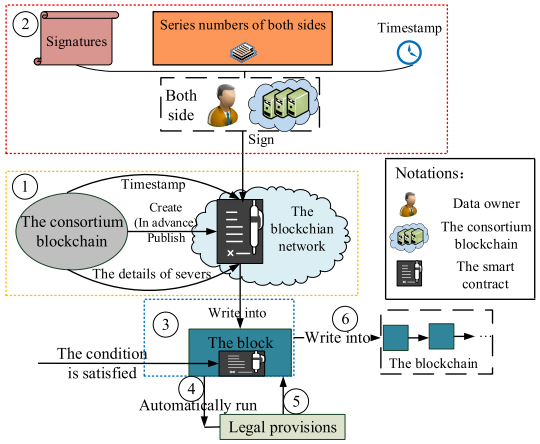

FIGURE 6. The working status of smart contract.

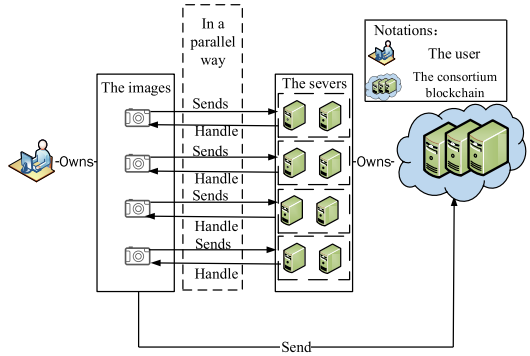

FIGURE 7. The sharding technique in the CB-SIFT approach.

extraction by different communities. For $\forall S_m^i$, $F(S_m^i)$ satisfies the following properties:

- Given $S_1^1, S_1^2, \ldots, S_m^1, S_m^2, \ldots, S_M^1, S_M^2$, and $C_1, \ldots, C_m, \ldots, C_M$ denotes range.
- $F(S_m^i) = C_m$ is a bijection from two servers to one community.
- Inputting $\forall S_m^i$, $F(S_m^i)$ randomly outputs a value C_m that has nothing to do with the S_m^i.

The encrypted images are sent to the consortium chain only if the money paid for the cloud fulfils the conditions in the smart contract and it is triggered. O produces a random $I_m^2(x, y)$ for each image, and it encrypts $I_m(x, y)$ through the function $I_m^1(x, y) = I_m(x, y) + I_m^2(x, y)$. The encrypted and divided image sections $I_m^i(x, y)$ are transmitted to the consortium chain at once, where $S_m^i(m = 1, \ldots, M)$ within the same community cooperates with each other to extract feature for its according section with the matching $S_m^1 \leftrightarrow I_m^1(x, y)$ and $S_m^2 \leftrightarrow I_m^2(x, y)(m = 1, \ldots, M)$.

3) FEATURE CONSENSUS

The section involves three stages for feature consensus: the **accounter selection**, **feature extraction** and **feature verification**. We exemplify the executing process of the community C_m.

At the **first stage**, the servers in the consortium chain compete for the accounter selection through the PoW mechanism.

- *Confirming the difficulty and target.* The difficulty determines the required times of hash operation performed by a node to generate a legal nonce, and much lower difficulty than the bitcoin blockchain is adopted in the CB-SIFT method because there are only a few consensus servers in the consortium chain and the computing is limited. In addition, a target is needed for the PoW mechanism, and the PoW target of bitcoin blockchain is calculated as follows:

$$Target = \frac{max\{Target\}}{Difficulty}. \tag{3}$$

- *Nonce calculation.* The core idea of the PoW consensus mechanism is to ensure the data consistency and consensus security by introducing the computing competition among distributed nodes. Each server competes with each other based on their respective computing to solve a complex and easy-to-verify *SHA*256 mathematics, which is presented in the formula below:

$$SHA256(SHA256(version + HS^{j-1} + HS^j + timestamp + Difficulty + nonce)) \leq Target. \tag{4}$$

- *Accounter determination.* The fastest server calculating the legal nonce get the block accounting rights and is the A_c, and it accounts for $V_m^i(m = 1, \ldots, M)$.

The **feature extraction** process is initiated in this stage. S_m^i in C_m cooperatively extracts feature of I_m^i, and the course is concurrently performed among different communities. S_m^i executes the following steps to fulfill the feature extraction on I_m^i for obtaining V_m^i:

- *Keypoint location.* The preliminary exploration of the keypoints is accomplished by comparing the two adjacent layers in DoG within the same octave. To find the extreme points in DoG, each pixel is compared with its 26 neighbors. S_m^1 and S_m^2 cooperatively compare two values with the BSCP. For P_{cm}^i, S_m^i calculates the main curvature H_m^i of P_{cm}^i by the BSCP and BSMP to eliminate the edge response and select out P_{sm}^i.
- *Orientation assignment.* In order to use the image's local feature to assign a reference direction for each keypoint in P_{sm}^i, S_m^i counts the contributions made by the pixels around the keypoint to its directions. The scale value σ of the keypoint is obtained after its precise positioning in the previous step. Then the Gaussian image closest to σ can be calculated by the formula below:

$$L^i(P_{sm}^i(x, y)) = G(P_{sm}^i(x, y, \sigma)) * I_m^i(P_{sm}^i(x, y). \tag{5}$$

$L^i(P_{sm}^i(x, y))$ is the keypoint value in LoG. The image gradient method is used to find the stable directions of the local structure. For $P_{sm}^i(x, y)$ detected in DoG, the gradient magnitude and direction distribution

$m^i_m(x,y)$ and $\Theta^i_m(x,y)$ of the pixels in the 3σ field around it are calculated with the formulas (1) and (2). Then with $m^i_m(x,y)$ and $\Theta^i_m(x,y)$ weighted by the same Gaussian window, S^i_m builds $\mathcal{H}^i_m$ where $0° - 360°$ is divided into 36 bins. The horizontal axis of $\mathcal{H}^i_m$ is the direction angle of gradient, and the vertical axis is the cumulative value of the gradient magnitude during the orientation angle. The highest peak HP^i_m and local peak within 80% of the highest peak LHP^i_m in $\mathcal{H}^i_m$ can be determined by continuously using the BSCP between two bins.

- *Descriptor generation.* There are three pieces of information for each keypoint so far: location, scale, and direction. This step is to create a descriptor for each keypoint and use a vector to describe it. The descriptor is a representation of the statistical results for the Gaussian image gradient in the neighborhood of each keypoint, and S^i_m performs three steps to obtain it. S^i_m first determines the image area required to calculate the descriptor. The area is divided into 4×4 subareas and each one is used as a seed point with 8 directions, ie, its histogram of the gradient direction divides $0° - 360°$ into 8 direction intervals. S^i_m then rotates the coordinate axis to the direction of the keypoint to ensure the rotation invariance. Finally, S^i_m splits the rotated area into 4×4 subregions and calculates their gradient histograms. The V^i_m with $4 \times 4 \times 8 = 128$ dimensions is acquired to be the keypoint descriptor.

In addition, $S^i_m(m = 1, \ldots, m-1, m+1, \ldots, M)$ performs the same steps as S^i_m to get $V^i_m(m = 1, \ldots, m-1, m+1, \ldots, M)$, and $V^i_m(i = 1, 2)(m = 1, \ldots, M)$ are separately recorded by A_c.

Next, A_c sends the recorded V^i_m respectively to $S^i_m(m = 1, \ldots, m-1, m+1, \ldots, M)$ for **verification**, and the same verification operations are applied for $V^i_m(m = 1, \ldots, m-1, m+1, \ldots, M)$.

- *Transmitting the feature data.* S^1_m sends its image data and computation related data $VD^i_m = (I^1_m, H^1_m, Det^1_m, P^1_{cm}, P^1_{sm}, m^1_m, \Theta^1_m, Nb^1_m, HP^1_m, LHP^1_m)$, and A_c sends the recorded V^1_m to $S^1_1, \ldots, S^1_{m-1}, S^1_{m+1}, \ldots, S^1_M$, which are the same as S^2_m.
- *Verifying the feature vector.* Once receiving the feature data from S^i_m and A_c, $S^i_m(m = 1, \ldots, m-1, m+1, \ldots, M)$ performs the same steps in the feature extraction stage to check V^i_m.
- *Accounting for the valid feature vector.* $S^i_m(m = 1, \ldots, m-1, m+1, \ldots, M)$ sends the signed V^i_m which is valid to the entire network. If V^1_m and V^2_m receive all the signatures of $S^i_m(m = 1, \ldots, m-1, m+1, \ldots, M)$, they are indicated valid and finally recorded by A_c.

$S^i_m(m = 1, \ldots, m-1, m+1, \ldots, M)$ performs the same steps as S^i_m to get $V^i_m(m = 1, \ldots, m-1, m+1, \ldots, M)$ verified, only the $V^i_{vm}(i = 1, 2)(m = 1, \ldots, m-1, m+1, \ldots, M)$ can be recorded by A_c.

4) FEATURE CONFORMING

This section includes the feature recovering and blocks building. We use V^m and B_j as an example, and it is the same steps for $V^m(m = 1, \ldots, m-1, m+1, \ldots, M)$ and block B_{j+1}.

- *Feature recovering.* The recorded $V^i_{vm}(i = 1, 2)$ is sent to O by A_c. O decrypts $V^i_{vm}(i = 1, 2)$ to recover the final image feature V^m according to the formula (6):

$$V^m = V^1_{vm} - V^2_{vm}. \tag{6}$$

  Besides, $V^m(m = 1, \ldots, m-1, m+1, \ldots, M)$ is recovered through the same steps.
- *Blocks building.* The conditions whether the effects of V^m meet the user's standards and whether the consortium chain have completed the image feature extraction within the time T_{cb} are checked once the V^m is decrypted. The smart contract is triggered as long as the two conditions fulfil the requirements in the DACs, and the checking results are also returned to O. Once the smart contract run automatically, the hashed V^1_{vm} is written into the blockbody of B_j by A_c. In addition, we extract the elements corresponding to the main direction's descriptors from V^1_{vm} to form $\bar{V}^1_{vm}$, which is stored in the blockheader of B_j together with the parameters HS^{j-1}, HS^j, version number, difficulty, timestamp and the smart contract. Unlike the traditional blockchain, since it is not necessary for each server to store the complete V^1_{vm}, the blockheader is specialized so that each server only needs to store the block headerchain and $\bar{B}$ blocks, which eases each server of its storage pressure. It is the same operation to complete the storages of $V^1_{vm}(m = 1, \ldots, m-1, m+1, \ldots, M)$ and $\bar{V}^1_{vm}(m = 1, \ldots, M)$. Furthermore, $V^1_{vm}(m = 1, \ldots, M)$ and $\bar{V}^1_{vm}(m = 1, \ldots, M)$ are separately stored in the blockbodies and blockheaders of B_j in the order of $m = 1, \ldots, M$.

  Besides, the block B_{j+1} is established through the similar steps. At this point, B_j and B_{j+1} are generated.

S^i_m will be rewarded according to the pre-determined rules in the DACs after the above stages are completed. FIGURE 8 describes a case study to demonstrate the entire process of the CB-SIFT scheme.

V. SYSTEM PRACTICALITY AND PERFORMANCE

To be able to construct a practical and well performed feature extraction approach for outsourcing encrypted images, we have to carefully analyze several characteristics of our design including the method practicality in terms of effectiveness and security, and the method performance in terms of efficiency and storage pressure.

A. PRACTICALITY

In this section, we prove that the CB-SIFT scheme is practical through the proofs of effectiveness and security.

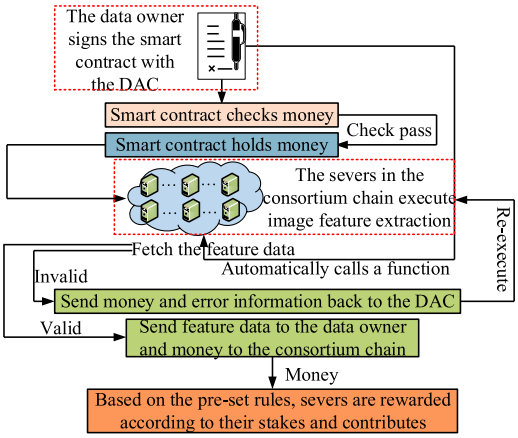

FIGURE 8. A case study of the CB-SIFT scheme.

1) EFFECTIVENESS

The effectiveness is deemed reached as long as the feature effects in our design are closest to the outputs in the original SIFT. In the CB-SIFT scheme, the effectiveness of feature effects is determined by the effectiveness of feature consensus which includes the **accounter selection**, **feature extraction** and **feature verification**.

Proposition 1 (The Effectiveness of Accounter Selection): The accounter selection in the CB-SIFT scheme is effective.

Proof: The accounter selection is realized by the computing competition based on the PoW mechnism which is a mature and widely applied consensus mechanism. Thus, the accounter selection process is effective.

Proposition 2 (The Effectiveness of Feature Extraction): In the CB-SIFT scheme, the feature extraction realized by the method in [25] is effective.

Proof: The feature extraction process in the CB-SIFT scheme is complied based on the feature extraction method in [25], which keeps the same feature effects as SIFT and has been examined effective.

Proposition 3 (The Effectiveness of Feature Verification): For $\forall S_m^i (i \in \{1,2\}, m \in \{1,\ldots,M\})$, if V_m^i is not tampered by S_m^i, it will be checked valid by all $S_m^i (m = 1,\ldots,m-1,m+1,\ldots,M)$ and agreed to be the V_{vm}^i. If not, $\exists S_m^i (i \in \{1,2\}, m \in \{1,\ldots,M\})$ who tampers with V_m^i, then it will not be agreed by at least one $S_m^i (i \in \{1,2\}, m \in \{1,\ldots,m-1,m+1,\ldots,M\})$, and there is no V_{vm}^i generated for it.

Proof: The effectiveness of feature verification is mainly decided by the stage of verifying the feature vector. Assume that $C_m (m = 1,\ldots,m-1,m+1,\ldots,M)$ concurrently verifies V_m^i, that is, $S_m^i (m = 1,\ldots,m-1,m+1,\ldots,M)$ performs the same steps in feature extraction to verify the validity of V_m^i withholding the same VD_m^i as the other $M-2$ servers ($S_1^i \leftrightarrow \cdots \leftrightarrow S_{m-1}^i \leftrightarrow S_{m+1}^i \leftrightarrow \cdots \leftrightarrow S_M^i$). We use C_1 as an example to prove the verification effectiveness of V_m^i from three related processes.

- *Verification of located keypoint.* S_1^i examines P_{sm}^i, P_{cm}^i, $\{P_{cm}^i\} - \{P_{sm}^i\}$ and the sample pixels orderly. S_1^i first compares the size of P_{sm}^i with its 26 neighbors by the BSCP, and it continues to execute the same operations to P_{cm}^i if P_{sm}^i is indeed larger or smaller. If the calibration of P_{cm}^i passes, $\{P_{cm}^i\} - \{P_{sm}^i\}$ is singled out to be testified for its edge response. Withholding the values of $H_m^i(x,y)$ and $Det_m^i(x,y)$ (actually the correlation values for calculating $H_m^i(x,y)$ and $Det_m^i(x,y)$ by the BSCP and BSMP), S_1^i determines whether $\{P_{cm}^i\} - \{P_{sm}^i\}$ is possessed with the edge response according to the ratio γ in the formula (7). In the formulas (7) and (8), $Tr(H_m^i)$ is the trace of H_m^i, α and β respectively denote the maximum and minimum eigenvalues of H_m^i. $\{P_{cm}^i\} - \{P_{sm}^i\}$ is unstable when γ is greater than the threshold (usually $\gamma = 10$).

$$\frac{Tr(H_m^i(x,y))^2}{Det_m^i(x,y)} = \frac{(\alpha+\beta)^2}{\alpha\cdot\beta} = \frac{(\gamma+1)^2}{\gamma}. \tag{7}$$

Where

$$\begin{aligned} H_m^i(x,y) &= \begin{pmatrix} D_{xx}(x,y) & D_{xy}(x,y) \\ D_{yx}(x,y) & D_{yy}(x,y) \end{pmatrix}; \\ Tr(H_m^i) &= D_{xx}(x,y) + D_{yy}(x,y) = \alpha + \beta; \\ Det(H_m^i) &= D_{xx}(x,y)D_{yy}(x,y) - (D_{yy}(x,y))^2 \\ &= \alpha\cdot\beta; \\ \gamma &= \frac{\alpha}{\beta}. \end{aligned} \tag{8}$$

The sample pixels are examined through the same operations, and P_{sm}^i is valid if all examinations pass.

- *Verification of assigning orientation.* For the valid P_{sm}^i, S_1^i further validates its HP_m^i and LHP_m^i. S_1^i uses the data $m_m^i(x,y)$, $\Theta_m^i(x,y)$, Nb_m^i in VD_m^i to draw $\mathcal{H}_m^i$, and testifies the authenticity of HP_m^i and LHP_m^i in $\mathcal{H}_m^i$ by the BSCP. The P_{sm}^i whose HP_m^i and LHP_m^i are valid will be further validated.
- *Verification of generated descriptor.* Each element in V_m^i of the finally valid P_{sm}^i is calibrated, and V_m^i is valid by C_1 when all its elements are verified pass.

At the same time, $C_2,\ldots,C_{m-1},C_{m+1},\ldots,C_M$ verifies V_m^i by the three processes, and V_m^i is finally effective only if all the verifications by $C_m (m = 1,\ldots,M)$ are passed. Therefore, for $\forall V_m^i$, if it has been tampered, the verification of V_m^i will not be passed in any part. In summary, the feature verification is effective.

Therefore, the same feature effects as SIFT are reached in the CB-SIFT scheme, and the CB-SIFT scheme is certified effective.

2) SECURITY

The security of outsourcing images contains the privacy of image pixels and features. Although the confidentiality of image pixels is guaranteed in [25], there are still security risks on the image features, which mainly exists in two processes: the feature extraction and storage about the following situations.

- **Feature extraction.** The cooperated servers may jointly tamper with the feature data.

- **Feature storage.** The attackers may try to obtain or tamper with the stored image features.

We prove that the CB-SIFT scheme solves these security issues from three small aspects.

For the security of **feature extraction**:

- *Security of feature consensus.* Feature extraction exists in the stage of feature consensus of the CB-SIFT scheme. Therefore, to keep the security of feature extraction, we require to certify the security of the complete feature consensus stage. Apart from the security of accounter selection, the extracted image features need to be verified by $S_m^i (m = 1, \ldots, M)$, which prevents S_m^1 and S_m^2 from jointly tampering with V_m^i.

For the security of **feature storage**:

- *Security of block data.* Features are stored in the blocks of the blockchain. The encrypted features are hashed and written into the blocks B_j and B_{j+1}. Besides, B_{j+1} is connected with the previous block B_j through HS^j, which can keep the block data secure.
- *Security of feature co-storage.* Different from the traditional blockchain, each server in the CB-SIFT method only saves the blockheader chain and $\bar{B}$ blocks, which can keep the blockchain integrity and security.

Proposition 4 (Security of Feature Consensus): On the one hand, the CB-SIFT scheme keeps the security of accounter chosen through the PoW consensus mechanism; on the other hand, the feature verification process is secure, and only the valid V_m^i is conformed to be V_{vm}^i.

Proof: The security of feature consensus includes the security of accounter selection, feature extraction and feature verification.

For the security of accounter selection, the PoW mechanism is a mature consensus algorithm and is widely applied, thus its security about selecting the accounter will not be described again. The security of feature extraction is determined by both the feature extraction and feature verification. As proven in the proposition 3, the verification process is valid. For example, for $\forall V_m^i$, if it is tampered during the feature extraction process, it will not be finally approved by all servers. Therefore, security of both the feature extraction and feature verification can be assured.

Proposition 5 (Security of Block Data): The block data containing $\bar{V_{vm}^2}$, V_{vm}^2, HS^j, HS^{j+1}, version number, difficulty, timestamp, and smart contract are secure after they are hashed and written into block B_{j+1} of the blockchain.

Proof: The CB-SIFT method hashes $\bar{V_{vm}^2}$ and V_{vm}^2 on the basis of the encrypted images, and saves them in the block B_{j+1}. In addition, for B_{j+1} itself, it is always connected with B_j by HS^j. Therefore, the block data is secure according to the tamper-resistant property of the blockchain.

Proposition 6 (Security of Feature Co-Storage): Let whp, $|B_{sum}|$ and S_m^{hon} respectively represent the malicious servers, the total number of blocks and the honest servers, and suppose that $|whp| = \varepsilon \cdot 2M \leq \frac{1}{3} \cdot 2M$. For the given $|B_{sum}|$, $\exists |B_0| \in N^+$, when $\bar{B} \geq |B_0|$, it is safe for each server to store $\bar{B}$ blocks and the blockheader chain, and the blockchain integrity can be assured.

Proof: In the CB-SIFT scheme, each server stores the blockheader chain. Analogous to the traditional blockchain, it is secure for each server to store the blockchain. Therefore, the blockheader chin in the system is secure. To keep the system secure, the safety of $\bar{B}$ blocks must be guaranteed.

The total number of blocks saved by whp is $2M \cdot \varepsilon \cdot \bar{B} (0 < \varepsilon \leq \frac{1}{3})$, and blocks by S_m^{hon} is $[2M - 2M \cdot \varepsilon] \cdot \bar{B}$.

Consider the most terrible case where whp jointly tampers with their common block B_w. There would be $|(B_w)^{whp}| = 2M \cdot \varepsilon$ copies of B_w saved by whp. To keep the blockchain secutiry, B_w saved by S_m^{hon} must fulfill the following circumstance:

$$|(B_w)^{hon}| > 2M \cdot \varepsilon. \tag{9}$$

From the formula (9), there is

$$min\{|(B_w)^{hon}|\} = 2M \cdot \varepsilon + 1. \tag{10}$$

Then the total number of blocks stored by S_M^{hon} fulfills

$$\Sigma_{hon} = (2M \cdot \varepsilon + 1) \cdot |B_{sum}|. \tag{11}$$

To keep the blockchain integrity, we let S_M^{hon} evenly save the blocks, and each one saves $|(B)^{hon}|$ blocks:

$$|(B)^{hon}| = \frac{\Sigma_{hon}}{2M - 2M \cdot \varepsilon} = \frac{(2M \cdot \varepsilon + 1) \cdot |B_{sum}|}{2M \cdot (1 - \varepsilon)}. \tag{12}$$

Let $\bar{B} = |(B)^{hon}| = \frac{(2M \cdot \varepsilon + 1) \cdot |B_{sum}|}{2M \cdot (1 - \varepsilon)}$, and the storage security and blockchain integrity are proved.

At this point, we theoretically certify that the CB-SIFT scheme is practical in terms of effectiveness and security. The next section certifies its performance in terms of efficiency and storage pressure.

B. PERFORMANCE

In this section, we prove that the CB-SIFT scheme is well performed through the proofs of efficiency and storage pressure.

1) EFFICIENCY

The efficiency means the time costs for image feature extraction. This section analyzes the available efficiency of the CB-SIFT scheme, which is mainly determined by the **efficiency of feature consensus** including the *accounter selection*, *feature extraction* and *feature verification*.

Proposition 7 (Efficiency of the CB-SIFT Scheme): The CB-SIFT scheme is highly efficient in image feature extraction, and it is almost M times of that in method [25].

Proof: We prove the efficiency of feature consensus from three aspects:

- *Accounter selection.* During the feature consensus stage of the CB-SIFT scheme construction, quite low difficulty is adopted in the CB-SIFT method according to the determination rules for the value of difficulty, that is, the smaller the computing, the lower the difficulty. The accounter selection time is mainly decided by the time to

count the legal nonce. Assume that the time required for calculating the legal nonce is at least T_{as} in the CB-SIFT scheme, and the total time for feature consensus is T_{fc}, therefore, $T_{as} \ll T_{fc}$.

- *Feature extraction.* Assume that the time required for an image feature extraction is T in the method [25] and T_{fe} in the CB-SIFT method, and T_{fe} satisfies:

$$T_{fe} = \frac{T}{M}. \tag{13}$$

- *Feature verification.* The total time for feature verification contains the time for transmitting the feature data, verifying the feature vector and accounting for the valid feature vector, where the time used to account for the valid feature vector is mainly spent in data transmission. During the processes of transmitting the feature data and accounting for the valid feature vector, A_c and S_m^i send V_m^i and VD_m^i to $S_m^i (m = 1, \ldots, m-1, m+1, \ldots, M)$, and $S_m^1, \ldots, S_m^{m-1}, S_m^{m+1}, \ldots, S_m^M$ separately sends the verification results to S_m^i. Let the T_{ft1} and T_{ft2} be the time separately wasted during the two processes in the CB-SIFT scheme, and T_{ft} be the time that it takes to perform the two processes for one image, and it satisfies:

$$T_{ft} = \frac{T_{ft1}}{M} + T_{ft2}. \tag{14}$$

Since the data transmission occurs among the consensus servers within the same consortium chain which has certain requirements for the configuration and the network environment of the consensus nodes, and is deployed in the D2D network, therefore, $T_{ft} \ll T_{cb}$.

S_m^i performs the same steps as the feature extraction during the feature verification process. Differently, the data $H_m^i(x, y)$, $Det_m^i(x, y)$, $m_m^i(x, y)$, $\Theta_m^i(x, y)$ and Nb_m^i, and the values I_m^i, P_{cm}^i, P_{sm}^i, HP_m^i, LHP_m^i and V_{vm}^i are known by S_m^i, therefore, the execution speed in the verification process is much faster than that in the feature extraction. Assume that it requires the time T_{fv} for validating an image by all communities, then it satisfies $T_{fv} \ll T_{fe}$. Let T_{fv}^M be the time for validating features of M images, and it satisfies:

$$T_{fv}^M = (M-1) \cdot T'_{fv} \leq (M-1) \cdot T_{fv}.$$
$$T_{fv} = \frac{(M-1) \cdot T'_{fv}}{M} \leq \frac{M-1}{T_{fv}^M}.$$

$$\begin{aligned} T_{cb} \approx T_{fc} &= T_{as} + T_{fe} + T_{ft} + T_{fv} \\ &= T_{as} + \frac{T}{M} + \frac{T_{ft1}}{M} + T_{ft2} + \frac{(M-1) \cdot T'_{fv}}{M} \\ &\approx \frac{T}{M} + \frac{T_{ft1}}{M} + T_{ft2} + \frac{(M-1) \cdot T'_{fv}}{M} \\ &\leq \frac{T}{M} + \frac{T_{ft1}}{M} + T_{ft2} + \frac{(M-1) \cdot T_{fv}}{M} \\ &\approx \frac{T}{M}. \end{aligned} \tag{15}$$

At this point, the proposition has been proved.

TABLE 2. Items saved by each server of our design and the bitcoin blockchain.

Items in CB-SIFT	Space	Items in bitcoin	Space
$(HS^j)^{cb}$	V_1	$(HS^j)^b$	V_1
$(HS^{j+1})^{cb}$	V_2	$(HS^{j+1})^b$	V_2
Ver^{cb}	V_3	Ver^b	V_3
$(D)^{cb}$	V_4	$(D)^b$	V_4
T^{cb}	V_5	T^b	V_5
$(SC)^{cb}$	V_6	$(SC)^b$	V_6
$(K)^{cb}$	V_7	$(K)^b$	V_7
$\bar{V}_{vm}^2$	V_8	V_{vm}^2	V_9

2) STORAGE PRESSURE

Storage pressure is determined by the space needed for each server to store the blockchain. Different from the traditional blockchain (e.g. bitcoin blockchain) where the whole blockchian is stored by each node, we re-design the blockheader to make each server only save the blockheader chain and $\bar{B}$ blocks. This section analyzes and compares the storage pressure of each server in the conditions of our design and the bitcoin blockchain.

Proposition 8 (Storage Pressure of Each Server): The servers in the blockchain system of the CB-SIFT scheme have less storage pressure than those in the bitcoin blockchain.

Proof: We specify all storage items of blocks saved by each server in the conditions of our design and the bitcoin blockchain.

- In the CB-SIFT method, items in the blockheader $(B_{head})^{cb}$ include $\bar{V}_{vm}^2$, $(HS^j)^{cb}$, $(HS^{j+1})^{cb}$, the version number Ver^{cb}, difficulty $(D)^{cb}$, timestamp T^{cb}, and smart contract $(SC)^{cb}$. Items in the blockbody $(B_{body})^{cb}$ include the number of images $(K)^{cb}$ and V_{vm}^2.
- In the bitcoin blockchain, items in the blockheader $(B_{head})^b$ include $(HS^j)^b$, $(HS^{j+1})^b$, the version number Ver^b, difficulty $(D)^b$, timestamp T^b, and smart contract $(SC)^b$. Items in the blockbody $(B_{body})^{cb}$ include the number of images $(K)^b$ and V_{vm}^2.

In the bitcoin blockchain, each server saves the blockchain which includes all $(B_{head})^b$ and $(B_{body})^b$. However, the blockchain system of CB-SIFT scheme requires each server to save $\bar{B}$ different blocks and the blockheader chain including all $(B_{head})^{sr}$. From the formula (12), the $\bar{B}$ blocks occupies pretty small storage in the blockchain, then the $\bar{B}$ blocks can be ignored compared with the blockheader chain and the blockchain. TABLE 2 shows the items saved by each server of our design and the bitcoin blockchain.

Let $V_1, \ldots, V_9$ be the volumes of the items above. From the TABLE 2, the items $\bar{V}_{vm}^2$ and V_{vm}^2 are the only different items between our design and the bitcoin blockchain. However, $\bar{V}_{vm}^2$ only contains just a few elements in V_{vm}^2, therefore, $\bar{V}_{vm}^2 \cdot |B_{sum}| < V_{vm}^2 \cdot |B_{sum}|$. Thus, the proposition is proved.

In summary, based on the theoretical analysis, the CB-SIFT scheme is conformed to be effective, secure, high efficient and small of storage pressure. We experimental evaluate the characteristics in the next section.

VI. EXPERIMENTAL EVALUATION

The experimental evaluation incudes the evaluations in terms of the effectiveness of feature consensus, efficiency and storage pressure.

- **Effectiveness.** The effectiveness of feature consensus including the accounter selection, feature extraction and feature verification. A detailed experimental implementation of the effectiveness about feature extraction has been carried out in [25]. The accounter selection and feature verification are actually the execution process of the PoW consensus mechanism. According to the property of the PoW consensus mechanism to assure the data consistency and consensus security, the consistency of the feature extracted through the feature extraction methodin [25] can be self-insured by the PoW consensus mechanism, that is, the effectiveness of feature effects can be kept. Thus, we would not evaluate the effectiveness experimentally.
- **Efficiency.** Efficiency refers to the time costs used for image feature extraction and it is mainly determined by the time spent during the feature consensus process of the CB-SIFT scheme. We empirically evaluate the efficiency of the CB-SIFT method by the total time costs spent during the three stages of feature consensus: the accounter selection, feature extraction and feature verification.
- **Storage pressure.** Besides, we experimentally measure the storage situation of each server within the consortium chain, and compare it with the bitcoin blockchian.

Our procedure is designed with JAVA, and we implement all components of the CB-SIFT scheme on the procedure and the consortium chain on the popular hyperledger fabric [37] with the computer attribute of Intel(R)core(TM) i5-4590 CPU @ 3.30GHZ and RAM:8GB. The goals of our evaluation are twofold. We first measure the time costs for feature consensus in the CB-SIFT scheme when the image size increases, and compare it with the method [25]. The second goal is to measure and compare the storage size of each server in the system of CB-SIFT scheme and bitcoin blockchain. We aim to establish that the efficiency and storage overhead of the CB-SIFT scheme matches its theoretical analysis in Section of PERFORMANCE.

A. EFFICIENCY EVALUATION

Experimental setup. We run several experiments with different number of the communities in the consortium chain and different size of the images to measure the time costs for feature consensus in the CB-SIFT scheme, and compare it with the scheme in [25]. We vary the number of communities from 2 to 4, and the size of images from $40*40$ to $200*200$.

Time costs for feature consensus. The results show that the time costs for feature consensus in the CB-SIFT scheme is much less than those in [25], and it almost decreases in multiples with the growing number of communities, which agrees with the theoretical analysis. The experimental results are plotted in FIGURE 9.

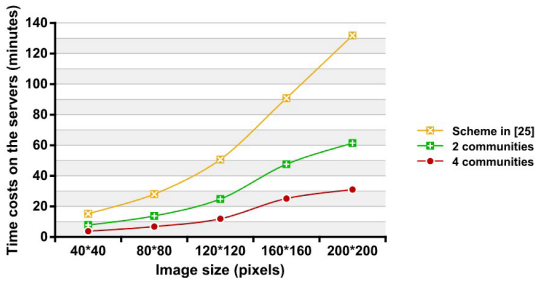

FIGURE 9. The time costs for feature consensus in the CB-SIFT scheme.

FIGURE 9 describes the time costs for an image feature consensus under the conditions of the scheme [25], and 2 and 4 communities in the CB-SIFT method respectively with the image pixels increasing by $40*40$, $80*80$, $120*120$, $160*160$ and $200*200$. With the same size of images, the time costs in [25] is rather longer than those in the CB-SIFT method with 2 and 4 communities, and the time costs is almost M times of those in the CB-SIFT approach, which satisfies the theoretical analysis. According to the FIGURE 9, for the image size of $200*200$, the scheme [25] spends respectively 70.4 and 100.768 minutes more than the time costs of the CB-SIFT method with 2 and 4 communities. Besides, with the image size growing, the time costs in the CB-SIFT method with 2 and 4 communities slightly change, and the more number of communities, the less time for feature consensus.

Besides, with the experimental results, the Eq. 15 could be further validated. Take the condition with the image size of $200*200$ and the number of communities of 4 as an example, and there are:

$$\begin{aligned} T_{as} &= 10.64s \\ T_{fe} &= 1761.6s \\ T_{ft} &= 63.035s \\ T_{fv} &= 25.655s \end{aligned} \tag{16}$$

With the value in Eq. 17, there is:

$$\begin{aligned} T_{cb} \approx T_{fc} &= T_{as} + T_{fe} + T_{ft} + T_{fv} \\ &= 10.64s + 1761.6s + 63.035s + 25.655s \\ &= 1860.96s \end{aligned} \tag{17}$$

B. STORAGE PRESSURE EVALUATION

Experimental setup. Different from the tradition blockchain (e.g. bitcoin blockchain), each server saves the blockheader chain and some blocks in the CB-SIFT approach. Suppose that the image size is $400*400$, we show how the CB-SIFT approach outperforms the bitcoin blockchain in ameliorating the storage pressure of each server with the length of blockchain changes from 1000 to 1000000.

Storage size. The experimental results in FIGURE 10 show that the storage size of each server in the CB-SIFT scheme is small and much less needed than that in the bitcoin blockchain, which agrees with the theoretical analysis.

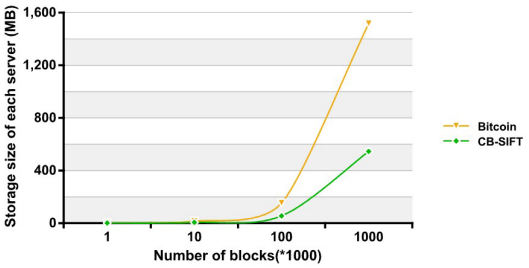

FIGURE 10. The stotrage size of each server in the CB-SIFT algorithm.

FIGURE10 describes the storage size needed by each server to store the block data under the conditions of the bitcoin and the CB-SIFT method when the blockchain length is growing from 1000, 10000, 100000 to 1000000. The results show that the required storage size in the bitcoin blockchain grows rather faster and is much larger than that in the system of CB-SIFT method. With the same length of the blockchain, the occupied space of each server in the bitcoin blockchain is about 2 to 5 times of that in the system of CB-SIFT method. The experiments show that when the length of blockchain is 100 0000, the occupied space of each server is 545.0563*MB* in our system, however, 1516.6411*MB* in the bitcoin blockchain, which is about 1000*MB* larger. The experimental results meet the theoretical analysis.

In summary, our experiments confirm the theoretically expected efficiency and storage pressure in the CB-SIFT scheme.

VII. CONCLUSION

The consortium chain-based outsourcing feature extraction over encrypted image provides further security, application ranges, efficiency and less storage pressure compared with the existing schemes. The CB-SIFT algorithm is based on the blockchain with the PoW consensus mechanism, and is implemented with the existing smart contract, DAC, sharding technique and D2D as well as a specially self-designed blockheader. The image feature is reached for consensus through the blockchain classical PoW mechanism; and the smart contract is also introduced into the blockchain, which strengthens security of the CB-SIFT scheme with preserving the advantages in the existing methods. The DAC plays a big role in widening the application ranges between the users and clouds. Most importantly, the sharding technique greatly enhances efficiency of the CB-SIFT scheme. Furthermore, the re-designated blockheader is in favor of alleviating the servers' storage burden. We also implement the detailed theoretical analysis and experimental evaluation to conform the practicality in terms of effectiveness and security, and performance in terms of efficiency and storage pressure of our design, which verifies the advantages realized in the CB-SIFT approach.

REFERENCES

[1] R. Wang, J. Yan, D. Wu, H. Wang, and Q. Yang, "Knowledge-centric edge computing based on virtualized D2D communication systems," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 32–38, May 2018.

[2] D. Wu, Q. Liu, H. Wang, D. Wu, and R. Wang, "Socially aware energy-efficient mobile edge collaboration for video distribution," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2197–2209, Oct. 2017.

[3] H. R. Motahari-Nezhad, B. Stephenson, and S. Singhal, "Outsourcing business to cloud computing services: Opportunities and challenges," *IEEE Internet Comput.*, vol. 11, no. 2, pp. 1–17, Sep. 2009.

[4] M. Rekik, K. Boukadi, and H. Ben-Abdallah, "Business process outsourcing to the cloud: What activity to outsource?" in *Proc. IEEE/ACS 12th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2015, pp. 1–7.

[5] J. Liu, Y. Y. Tang, and Y. C. Cao, "An evolutionary autonomous agents approach to image feature extraction," *IEEE Trans. Evol. Comput.*, vol. 1, no. 2, pp. 141–158, Jul. 1997.

[6] A. Hyvarinen, E. Oja, P. Hoyer, and J. Hurri, "Image feature extraction by sparse coding and independent component analysis," in *Proc. 14th Int. Conf. Pattern Recognit.*, vol. 2, Aug. 1998, pp. 1268–1273.

[7] T. Kobayashi and N. Otsu, "Image feature extraction using gradient local auto-correlations," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 346–358.

[8] L. Juan and O. Gwun, "A comparison of SIFT, PCA-SIFT and SURF," *Int. J. Image Process.*, vol. 3, no. 4, pp. 143–152, 2009.

[9] S. Yang, J. Hu, and W. Huang, "Recognition of electro-magnetic information leakage of computer based on multi-image blind deconvolution," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl. (WASA)*, 2017, pp. 883–889.

[10] M. Ye, Z. Cao, Z. Yu, and X. Bai, "Crop feature extraction from images with probabilistic superpixel Markov random field," *Comput. Electron. Agricult.*, vol. 114, pp. 247–260, Jun. 2015.

[11] A. R. Yadav, R. S. Anand, M. L. Dewal, and S. Gupta, "Multiresolution local binary pattern variants based texture feature extraction techniques for efficient classification of microscopic images of hardwood species," *Appl. Soft Comput.*, vol. 32, pp. 101–112, Jul. 2015.

[12] B. Wang, Y. Liu, F. Liu, and R. Zhang, "A new cloud detection method based on multi-scale feature extraction," in *Proc. Int. Conf. Natural Comput. (ICNC)*, Jul. 2013, pp. 863–867.

[13] E. Altantsetseg, Y. Muraki, K. Matsuyama, and K. Konno, "Feature line extraction from unorganized noisy point clouds using truncated fourier series," *Vis. Comput.*, vol. 29, nos. 6–8, pp. 617–626, 2013.

[14] Z. Qin, J. Yan, K. Ren, C. W. Chen, and C. Wang, "Towards efficient privacy-preserving image feature extraction in cloud computing," in *Proc. 22nd ACM Int. Conf. Multimedia (ACM MM)*, 2014, pp. 497–506.

[15] J. Serafin, E. Olson, and G. Grisetti, "Fast and robust 3D feature extraction from sparse point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4105–4112.

[16] J. Wang, S. Hu, Q. Wang, and Y. Ma, "Privacy-preserving outsourced feature extractions in the cloud: A survey," *IEEE Netw.*, vol. 31, no. 5, pp. 36–41, Oct. 2017.

[17] Z. Xia, X. Ma, Z. Shen, X. Sun, N. N. Xiong, and B. Jeon, "Secure image LBP feature extraction in cloud-based smart campus," *IEEE Access*, vol. 6, pp. 30392–30401, 2018.

[18] K. Ren, "Secure outsourcing image feature extraction: Challenges and solutions," in *Proc. Int. Workshop*, 2015, p. 1.

[19] D. Wu, F. Zhang, H. Wang, and R. Wang, "Security-oriented opportunistic data forwarding in mobile social networks," *Future Gener. Comput. Syst.*, vol. 87, pp. 803–815, Oct. 2018.

[20] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing," in *Proc. Comput. Sci. Electron. Eng. (ICCSEE)*, vol. 1, Mar. 2012, pp. 647–651.

[21] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic trust relationships aware data privacy protection in mobile crowd-sensing," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2958–2970, Aug. 2018.

[22] T. Lindeberg, "Scale invariant feature transform," *Scholarpedia*, vol. 7, no. 5, p. 10491, May 2012.

[23] C. Y. Hsu, C. S. Lu, and S. C. Pei, "Image feature extraction in encrypted domain with privacy-preserving SIFT," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4593–4607, Nov. 2012.

[24] Z. Qin, J. Yan, K. Ren, C. W. Chen, and C. Wang, "SecSIFT: Secure image SIFT feature extraction in cloud computing," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 4s, p. 65, 2016.

[25] S. Hu, Q. Wang, J. Wang, Z. Qin, and K. Ren, "Securing SIFT: Privacy-preserving outsourcing computation of feature extractions over encrypted image data," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3411–3425, Jul. 2016.

[26] B. Guo, D. Zhang, and D. Yang, "Read more from business cards: Toward a smart social contact management system," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, Aug. 2011, pp. 384–387.
[27] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Blockchain based efficient and robust fair payment for outsourcing services in cloud computing," *Inf. Sci.*, vol. 462, pp. 262–277, Jun. 2018.
[28] A. Chepurnoy, M. Larangeira, and A. Ojiganov. (2016). "Rollerchain, a blockchain with safely pruneable full blocks." [Online]. Available: https://arxiv.org/abs/1603.07926
[29] A. Judmayer, N. Stifter, K. Krombholz, and E. Weippl, "Blocks and chains: Introduction to bitcoin, cryptocurrencies, and their consensus mechanisms," *Synthesis Lect. Inf. Secur., Privacy, Trust*, vol. 9, no. 1, pp. 1–123, 2017.
[30] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 776, pp. 11676–11686, 2018.
[31] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A blockchain based truthful incentive mechanism for distributed P2P applications," *IEEE Access*, vol. 6, pp. 27324–27335, 2018.
[32] J. Bruggeman, "Consensus, cohesion and connectivity," *Social Netw.*, vol. 52, pp. 115–119, Jan. 2018.
[33] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2016, pp. 3–16.
[34] D. Wu, J. Yan, H. Wang, D. Wu, and R. Wang, "Social attribute aware incentive mechanism for device-to-device video distribution," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1908–1920, Aug. 2017.
[35] M. Ruta, F. Scioscia, S. Ieva, G. Capurso, and E. Di Sciascio, "Semantic blockchain to improve scalability in the Internet of Things," *Open J. Internet Things*, vol. 3, no. 1, pp. 46–61, 2017.
[36] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 34–37, 2014.
[37] *Hyperledger*. (2017). [Online]. Available: https://www.hyperledger.org/projects/fabric/

XIAOQIN FENG received the B.S. degree in information and computing science from Xidian University, Xi'an, China, in 2016, where she is currently pursuing the Ph.D. degree in cyberspace security. Her research interests include blockchain, consensus mechanism, and blockchain applications.

JIANFENG MA (M'16) received the M.S. degree in mathematics from Shaanxi Normal University in 1988 and the Ph.D. degree in computer software and communications engineering from Xidian University in 1995. He is currently a Full Professor and a Ph.D. Supervisor with Xidian University. His main research interests include information security, coding theory, and cryptography. He is a member of the China Computer Federation.

TAO FENG received the M.E. degree in control theory and control engineering from the Lanzhou University of Technology in 1999 and the Ph.D. degree in computer architecture from Xidian University in 2008. He is currently a Full Professor and a Ph.D. Supervisor with the Lanzhou University of Technology. His main research interests include information security, provable theory of security protocols, wireless network security, and sensor network security. He is a member of the China Computer Federation and China Cryptography Federation.

YINBIN MIAO received the B.E. degree in telecommunication engineering from Jilin University, Changchun, China, in 2011, and the Ph.D. degree in telecommunication engineering from Xidian university, Xi'an, China, in 2016. He is currently a Lecturer with the School of Cyber Engineering, Xidian University. His research interests include information security and applied cryptography.

XIMENG LIU received the B.S. and Ph.D. degrees in electronic engineering and cryptography from Xidian University in 2010 and 2015, respectively. He is currently a Full Professor with Fuzhou University, Fuzhou, China. He is also a Research Fellow at the School of Information System, Singapore Management University, Singapore. His research interests include cloud security, applied cryptography, and big data security.