IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON NOVEL LEARNING APPLICATIONS AND SERVICES FOR SMART CAMPUS

Received May 14, 2018, accepted June 2, 2018, date of publication June 8, 2018, date of current version June 26, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2845456

Secure Image LBP Feature Extraction in Cloud-Based Smart Campus

ZHIHUA XIA[1], (Member, IEEE), XIAOHE MA[1], ZIXUAN SHEN[1], XINGMING SUN[1], NEAL N. XIONG[2], AND BYEUNGWOO JEON[3]

[1]Jiangsu Engineering Center of Network Monitoring, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2]Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA
[3]College of Information and Communication Engineering, Sungkyunkwan University, Seoul 110-745, South Korea

Corresponding author: Neal N. Xiong (xiongnaixue@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672294, Grant 61502242, Grant 61702276, Grant U1536206, Grant U1405254, Grant 61772283, Grant 61602253, Grant 61601236, and Grant 61572258, in part by the Six Peak Talent Project of Jiangsu Province under Grant R2016L13, in part by the National Key Research and Development Program of China under Grant 2018YFB1003205, in part by NRF under Grant 2016R1D1A1B03933294, in part by the Jiangsu Basic Research Programs–the Natural Science Foundation under Grant BK20150925 and Grant BK20151530, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions fund, and in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology fund, China. The work of Z. Xia was supported by the BK21+ Program from the Ministry of Education of Korea.

ABSTRACT The smart campus can monitor students in real time by analyzing students' images, but a large number of images bring an unbearable burden to the smart campus. The convenience of cloud computing has attracted smart campus to outsource their huge amount of data to cloud servers. Although the outsourcing of data can reduce the computational and storage burden on smart campus, the privacy preserving becomes the biggest concern. This issue has attracted many researchers to study the protection of outsourced multimedia data. In this paper, we propose an effective and practical privacy-preserving computation outsourcing protocol for the local binary pattern (LBP) feature over huge encrypted images. The image owner uploads the encrypted version of images to the cloud. The cloud server takes the responsibility of extracting the LBP features from encrypted images for various applications. In the encryption process, an image is divided into non-overlapping blocks at first, and the blocks are shuffled to protect the image content. Next, all the non-center pixels in each block are shuffled. Finally, the pixels are encrypted by splitting the original image data randomly. When such an encrypted image is received, the cloud servers can calculate the LBP features by secure multiparty computation. The extracted features can be applied to many applications, such as texture classification, image retrieval, face recognition, and so on.

INDEX TERMS Cloud computing, local binary pattern, privacy-preserving, smart campus.

I. INTRODUCTION

In the past few years, the explosive growth of knowledge has led to a variety of ways to spread knowledge and education. The transformation of education model has induced the smart campus, which implements education by combining information and information technology to meet the various needs of students and schools. Specifically, there are many new learning applications and services in smart campuses. A typical example is to feedback the student's position status in real time by continuously monitoring and analyzing various students' image information (such as cloud computing [1] platform). In the daily life of campus, billions of digital images are generated every day. We can continuously monitor and analyze information of students by analyzing image features. However, this leads to an unaffordable storage and calculation problem. Smart campus images can be stored in cloud servers and smart campus image features can also be extracted through cloud computing to alleviate storage and computation problems. Nevertheless, there are many challenges that need to be solved, such as the security problem of smart campus images in cloud computing.

As a new information technology, cloud computing provides the data owners with a wealth of storage and computing resources. By outsourcing large amounts of multi-media data and complex computations such as image feature extraction operations to the cloud, data owners can reduce local

data management and huge computational burden. The data owners get a huge profit, but outsourcing inevitably brings security problems, because the data owners have very little control over the data after the outsourced data. Specifically, images may involve sensitive information such as personal identity, geographic location and even social relationships. Therefore, uploading the unprotected multimedia data to the cloud server may cause personal privacy to be leaked.

Outsourced image data can reveal the privacy of image owners. Extracted image features may also reveal important privacy information. An attacker could guess the image contents by analyzing the image database in the cloud server. Therefore, it is important to protect the privacy of the image by means of encryption. In order to protect the privacy, images should be encrypted before outsourcing. The encryption is common way to protect information, but the encrypted images (i.e., ciphertext) will hinder the operations normally performed in plaintext. This leads to the fact that the extracted image features in ciphertext domain lose the effectiveness of original features.

In the present study, a number of privacy protection outsourcing solutions have been proposed. These works mainly focus on the calculation of digital data or text data which can handle a variety of mathematical problems, including modular exponents [2], sequence comparisons [3], linear equations [4] and kNN searches [5]. On the other hand, in recent years, some existing works focus on extracting image features from encrypted images such as scalar invariant feature transform (SIFT) [6], [11], [17], [18] and histogram of oriented gradient (HOG) [7]. They use Paillier cryptographic system [8] or somewhat homomorphic encryption [9] to protect the privacy of images without affecting the extraction of image features. The application of privacy-preserving data in the ciphertext domain has been extended to the areas of multimedia content retrieval [10] and face recognition [11].

Image features are widely used in various fields such as object detection [12], image retrieval [13], [14], information hiding [15], fingerprint detection [16], etc. LBP (local binary pattern) features are widely used in many fields of computer vision because of the simple calculation of LBP features and good effect. In this paper, we propose a secure method to extract the LBP features from the encrypted image. The images are typically encrypted by block permutation, pixel permutation, and image segmentation. The specially-designed encryption can support direct extraction of the LBP features even from the encrypted images at the cloud. Additionally, it can be made that the extracted feature is also encrypted but support similarity computation.

The contributions of outsourced secure LBP extraction are summarized as follows.

(1) As far as we know, our proposed secure LBP extraction scheme is the first privacy-protected LBP extraction scheme in the ciphertext domain. This algorithm can be used for many LBP-based applications while protecting privacy.

(2) We encrypt images by block permutation, pixel permutation, and image segmentation. These three steps can well protect the image content without affecting the direct extraction of LBP features. In this way, our secure LBP feature can be extracted on multiple cloud servers without requiring additional communication between the image owner and the cloud server.

(3) Other features extracted by the privacy preserving image feature extraction scheme are encrypted and can only be used after the feature has been decrypted by users. In contrast, LBP features extracted in our scheme can be used directly without decryption and the application effect on image retrieval is not bad.

The rest of this paper is organized as follows: In Section II, we elaborate on the re-search status of the privacy-preserving of image feature extraction. In the next section, we introduce system model, security model and some preliminaries. In Section IV, we formally present the scheme design. In Section V, we describe analysis privacy, of the proposed scheme. In Section VI, we describe evaluation of the correctness, of the proposed scheme. Finally, conclusions and future work are given in Section VII.

II. RELATED WORKS

As far as we know, Lowe [17] is the first to propose a secure SIFT [18] feature extraction method. The encryption method they use is Paillier homomorphic encryption. Homomorphic encryption [19] can satisfy the addition and multiplication in the ciphertext domain and can be used to solve convolution operations in SIFT computation, but it cannot solve the comparison operation. Therefore, the author designed a ciphertext comparison scheme where the data owner generates a number of encryption thresholds for secondary communication, but this requires huge storage, computation and communication costs. In addition, Paillier homomorphic encryption cannot calculate square root in the ciphertext domain. Therefore, the author modified the step of feature descriptor calculation, which caused the extracted SIFT to lose its original characteristics.

Wang *et al.* [7] proposed two schemes for shape-based feature extraction of encrypted images, one is more practical and easier to implement but lacks proof of security, while the other has proven security. However, the experimental analysis provided by the author is too short and not persuasive. Qin *et al.* [20] found that the location of SIFT feature points would reveal the shape information of the image. They proposed a secure SIFT extraction method using order-preserving encryption and random permutation. They modified the original steps of SIFT to facilitate feature extraction, but reduced the number of directional features of the original SIFT.

Recently, Dalal and Triggs [21] proposed a secure SIFT feature extraction method by using somewhat homomorphic encryption. The authors proposed two security protocols; batch safety multiplication protocol and batch security

comparison protocol, to handle SIFT extraction in encrypted domain and improve efficiency. As in the previous paper [20], this approach modifies the original SIFT steps and reduces the number of directional features. Considering the similarity between HOG (histogram of oriented gradients), SURF (speeded up robust features) and SIFT features in convolution operations and comparison operations, in their other works [22], [23], the authors applied these two protocols to implement the secure extraction of HOG [24] and SURF [25].

Li *et al.* [26] proposed a double decryption-based privacy-preserving SIFT scheme over the encrypted domain. The authors use BCP (Bresson, Catalano and Pointcheval) [27] as their encryption scheme, which is an additively homomorphic scheme with double decryption mechanisms and is actually a variant of Paillier homomorphic encryption. They used secure multiparty computation [28] to perform comparisons during the positioning of extreme points. Their scheme reveals the location of the key points. Their scheme requires huge storage and compute costs and they still do not address the issue of a reduction in the number of directional features resulting from modifying SIFT extraction steps. Jiang *et al.* [29] proposed an effective and practical privacy-preserving scale-invariant feature transform (SIFT) scheme for encrypted image. It uses leveled homomorphic encryption [30] based on new encoding schemes, new homomorphic comparison, division and derivative encryption. Their scheme can realize higher computing efficiency, greatly reduce communication cost and interactive times between user and server, and perform correct feature point detection, accurate feature point description and image matching.

There are some shortcomings in the above researches. Their schemes require huge storage and compute costs and they still do not address the issue of a reduction in the number of directional features resulting from modifying SIFT extraction steps. Since the feature extraction operations include mathematical calculations such as addition, subtraction, multiplication and division, the selection of encryption algorithms is particularly important. To satisfy mathematical calculations, some schemes use homomorphic encryption, but additional comparison schemes are needed to extract extreme points. Some schemes use order-preserving encryption to ensure that the ciphertext order is the same as the plaintext order to facilitate comparison of ciphertext sizes. The time complexity of homomorphic encryption is too high, and the image owner takes a lot of time to encrypt the image locally. The extracted features must be decrypted before they can be used, which increases communication costs. It is not guaranteed that the original properties of SIFT are not changed. The existing privacy-preserving feature extraction algorithms mainly focus on SIFT or similar features to SIFT.

In this paper, we seek for secure outsourcing of another prevalent feature extraction method of LBP which has been widely employed in many applications.

III. SYSTEM OVERVIEW AND PRELIMINARIES

A. SYSTEM MODEL

For the feature extraction system that protects users' privacy, this paper takes into account the scenario shown in Fig. 1.

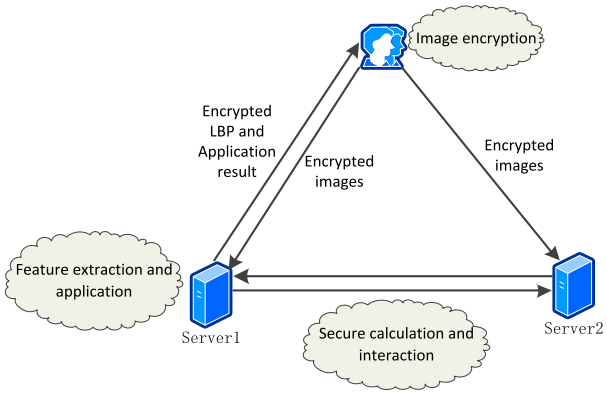

FIGURE 1. System models.

Data owner holds a large-scale image databases $\mathcal{I} = \{I_i\}_{i=1}^{n}$, to be outsourced for cost saving and efficient utilization. For privacy preserving, the image database needs to be encrypted before being uploaded, generating two encrypted image sets $\mathcal{C}_1 = \{C_{1i}\}_{i=1}^{n}$ and $\mathcal{C}_2 = \{C_{2i}\}_{i=1}^{n}$. Except the image encryption, the image owner would like to outsource the computation and storage tasks to the cloud server as many as possible. Moreover, the data owner sends the application request to the cloud server.

Cloud server stores the encrypted image for the data owner and provides the LBP feature extraction service for the data owner. In our system, the expected image processing results are a set of LBP features and LBP-based application results. The cloud consists of two entities, which are independent cloud server providers. The functions of these entities are as follows:

Server 1 and Server 2 receive the encrypted images. They receive different encrypted images respectively. Then they calculate the received encrypted images respectively. Server 2 sends the result to Server 1. The LBP feature is finally extracted by Server 1. In the following article, Server 1 and Server 2 are respectively written as S_1 and S_2.

In addition, the cloud server in our solution also provides LBP application services. In order to achieve the purpose of protecting privacy, the cloud server does not have to understand the results from the request sent by the data owner. The cloud returns extracted image features or feature-based operational results to the data owner. In other words, the cloud server is powerful in completing the requested task, but does not compromise data privacy

In our model, the user only needs to prepare a copy of the encrypted image as an input and then send it to the server for other operations. Moreover, the server generates the LBP through a secure multiparty computation framework without

having to know or learn anything to compromise the privacy of the user.

B. SECURITY MODEL

Our goal is to protect the privacy of image content while enabling the cloud server to execute the LBP algorithm on it. Specifically, we treat the image content's information (pixel values and descriptive features extracted from images) as the data owner's personal information.

In our security model, we consider the cloud server to be ''honest-but-curious'' and ''independent''. That is, the cloud server correctly implements the security LBP algorithm. However, the cloud server tries to learn additional information from encrypted data and all of operations performed by it. In our scheme, the data owner uploads the encrypted images to the cloud server, which performs all operations on the encrypted images. However, the cloud server has no other information than the encrypted LBP features, and the encrypted LBP features do not reveal any information about the image. Therefore, the privacy of image content can be preserved from the cloud server. Similar to a secure multi-party computing scenario, we assume that cloud entities are ''independent'' of each other. Here, S_1 and S_2 would explicitly state non-collusion.

C. LOCAL BINARY PATTERN

Local binary pattern (LBP) was firstly proposed in 2002 for the texture representation [31]. In the calculation process of LBP, the image is divided into overlapping blocks with a fixed size such as 3×3. The center and its 8 neighbors in each block are compared with their gray value. The position is marked as '1' if the corresponding gray value is larger than the center pixel value; otherwise, the position is encoded by '0'. Then, the 8 points in the 3×3 block produce an 8-bits binary number, and the LBP value is obtained. The extraction process is illustrated in Fig. 2. Finally, the image is represented by the histogram of LBP values.

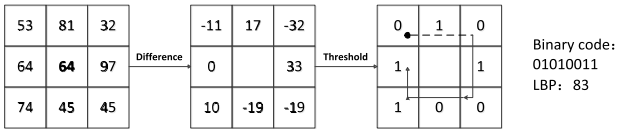

FIGURE 2. Extraction of the original local binary pattern.

A more formal LBP operation can be defined as

$$\mathrm{LBP}\,(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(i_p - i_c), \tag{1}$$

where (x_c, y_c) denotes the position of the center pixel, i_c and i_p denote the brightness of the adjacent pixels. $s(\cdot)$ denotes a symbolic function:

$$s\,(\chi) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & else. \end{cases} \tag{2}$$

This description method allows researchers to capture the details of the image well. In fact, researchers can use it to get the most advanced level in texture classification. Because LBP features have the ability to depict local texture features of images, they are widely used in the fields of image retrieval [32] and face recognition [33], [34].

IV. THE PROPOSED SCHEME

There are three entities in the proposed scheme, the image owner, the cloud server S_1 and the cloud server S_2. The image owner needs to save the image to the remote server. To protect privacy, the owner needs to encrypt the images before uploading the images. In our scheme, we generate two encrypted image sets. Both servers receive different sets of encrypted images. After receiving the encrypted images, the cloud servers perform the same calculation operations on the encrypted image sets, and then a communication between S_1 and S_2 will be carried out. Finally, the LBP feature is extracted from the encrypted image by S_1. We call our scheme as PPLBP.

A. OVERVIEW OF THE PROPOSED SCHEME

The proposed scheme includes three algorithms which are respectively executed by three entities, i.e., GenKey and EncImg algorithms executed by the image owner, LBP feature extraction executed by cloud servers. Then, other algorithms like image retrieval or face recognition are performed by using LBP features.

First of all, the image owner generates a set of secret keys $\mathcal{K}$ by GenKey algorithm. Then, the owner runs EncImg to encrypt the image database $\mathcal{I} = \{I_i\}_{i=1}^n$ and generates two encrypted image databases $\mathcal{C}_1 = \{C_{1i}\}_{i=1}^n$ and $\mathcal{C}_2 = \{C_{2i}\}_{i=1}^n$. The image owner sends the encrypted image database $\mathcal{C}_1$ and $\mathcal{C}_2$ to S_1 and S_2. Besides, the set of secret keys $\mathcal{K}$ is reserved by the image owner. A summarization of the algorithms is presented in Fig. 3.

Image owner side:

$\mathcal{K} \leftarrow$The key secret generation algorithm takes the security parameters as the inputs and returns the secret key $\mathcal{K}$. This step is abbreviated as GenKey.

$\mathcal{C} \leftarrow$The image encryption algorithm takes the secret key $\mathcal{K}$ and the image database $\mathcal{I}$ as inputs, then returns the encrypted image databases $\mathcal{C}_1$ and $\mathcal{C}_2$. This step is abbreviated as EncImg. Finally, the data owner uploads $\mathcal{C}_1$ and $\mathcal{C}_2$ to S_1 and S_2.

Cloud servers side:

$\mathcal{F} \leftarrow$LBP is the feature extraction algorithm which takes as inputs the encrypted image databases $\mathcal{C}_1$ and $\mathcal{C}_2$, then returns the set of features $\mathcal{F}$.

FIGURE 3. Overview of the algorithms in the secure image LBP extraction scheme.

In the following sections, we present a detailed introduction of our PPLBP protocol.

B. ENCRYPTION

In order to protect the image owner's image privacy, the image is encrypted before outsourced to the cloud sever, as described in follows. The image consists of two types of information, color and texture information, which require appropriate protection [35]. In our scheme, the color information is protected by image segmentation, and the texture information is protected by shuffling the pixel position.

The images are divided into 3×3 non-overlapping blocks which are shuffled by random permutation. Following, the pixel locations except for the center point in each block are shuffled in the same order. Therefore, we need to generate a pseudo-random permutation generator and a secret key for the image segmentation as follow:

$$\mathcal{K}_{Enc} = \left\{RandGen, k_{bp}, k_{pp}, k_{seg}\right\}. \tag{3}$$

The encryption process is illustrated in Fig. 4.

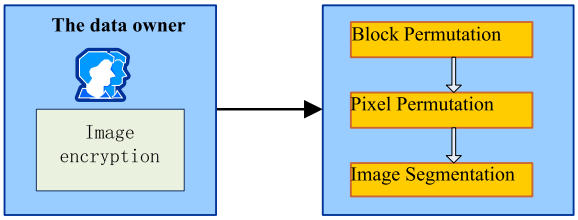

FIGURE 4. Overview of the encryption process.

1) BLOCK PERMUTATION

The secret key k_{bp} is used to generate random permutations from the range $[1 \ldots blocknum]$, where $blocknum$ is the total number of non-overlapped blocks in an image. Here, the random permutation generated is used to shuffle image blocks. The random permutation is generated with k_{bp} as follow:

$$rand\_pmt_{bp} \leftarrow RandGen(k_{bp}, [1, \ldots, blocknum]). \tag{4}$$

The total number of non-overlapped blocks in image $\mathcal{I}$ is calculated as $= \frac{imagesize}{9}$.

The image $\mathcal{I}$ is divided into $blocknum$ 3×3 non-overlapping image blocks. We denotes the non-overlapping image blocks as $block$. We define the permutated image as $\mathcal{I}'$ and divide $\mathcal{I}'$ into the blocks denoted as $block'$. For $\forall block'[i] \in \mathcal{I}'$, do $block'[i] \leftarrow \text{block}\left[rand_{pmt\,bp}[i]\right]$. The process of block permutation is defined in Algorithm 1.

2) PIXEL PERMUTATION

The secret key k_{pp} is used to generate random permutations. The k_{pp} is used to generate random permutation from the range $[1 \ldots 8]$, where the number of pixels around the center pixel is 8. We define the permutated image as $\mathcal{I}''$ and divide $\mathcal{I}''$ into blocknum 3×3 non-overlapping image blocks denoted as $block''$. For j-th block $block''_j \subset \mathcal{I}''$, the random permutation is used to shuffle pixels and generated as follows:

$$rand\_pmt_{pp} \leftarrow RandGen(k_{pp}, [1, \ldots, 8]). \tag{5}$$

Algorithm 1 BlockPMT

Input: $\mathcal{I}$, k_{bp}
Output: $\mathcal{I}'$
1: Calculate the total number of non-overlapped blocks in image $\mathcal{I}$ as $blocknum = \frac{imagesize}{9}$;
2: Generate the random permutation $rand_pmt_{bp} \leftarrow RandGen(k_{bp}, [1, \ldots, blocknum])$;
3: Divide the image $\mathcal{I}$ in to non-overlapped blocks denoted as $block$;
4: Define the permutated image as $\mathcal{I}'$, divide $\mathcal{I}'$ into the blocks denoted as $block'$;
5: for $\forall block'[i] \in \mathcal{I}'$ do
6: $block'[i] \leftarrow block\left[rand_{pmt\,bp}[i]\right]$
7: end for

Algorithm 2 PixelPMT

Input: $\mathcal{I}'$, k_{pp}
Output: $\mathcal{I}''$
1: Denote the permutated image as $\mathcal{I}''$;
2:Divide $\mathcal{I}'$ into the blocks denoted by$block'$;
3: Divide the $\mathcal{I}''$ into non-overlapped blocks denoted by $block''$;
4: Generate the random permutation for j-th block $block''_j \subset \mathcal{I}''$ as $rand\_pmt_p \leftarrow RandGen(k_{pp}, [1, \ldots, 8])$
5: for $\forall block''_j \subset \mathcal{I}''$ do
6: for $\forall block''_j[i] \in block''_j$ do
7: $block''[i] \leftarrow block'\left[rand\_pmt_{pp}[i]\right]$
8: end for
9: end for

Then, for $\forall block''_j[i] \in block''_j$, do $block''[i] \leftarrow block'\left[rand\_pmt_{pp}[i]\right]$. The process of pixel permutation is defined in Algorithm 2.

3) IMAGE SEGMENTATION

The secret key k_{seg} is used to split the images and generated as follows:

$$k_{seg} \leftarrow RandGen(imgseg). \tag{6}$$

We denote the final encrypted image as $\mathcal{C}$. For an image I''_i with size of $n \times n$ pixels, use k_{seg} selects n^2 integers from $[0, 255]$ as C_{2i}. For $\forall I''_i \subset \mathcal{I}''$, do $C_{1i} = I''_i + C_{2i}$.

Finally, the encrypted images are uploaded to the cloud server. The process of image segmentation is defined in Algorithm 3 and the whole process of image encryption is defined in Algorithm 4.

C. FEATURE EXTRACTION

The feature extraction operation requires interaction between S_1 and S_2 to complete the comparison operation. One of the most famous protocols for comparing private data of two parties is Yao's millionaires' problem [36]. Inspired by their ideas, we also use secure multiparty computation to build our comparative agreement.

Algorithm 3 ImgSeg

Input: $\mathcal{I}''$, k_{seg}
Output: $\mathcal{C}_1$ and $\mathcal{C}_2$
1: Denote the final encrypted image as $\mathcal{C}$;
2: Generate the secret key as $k_{seg} \leftarrow ImgSeg$;
3: For an image I_i'' with size of $n \times n$ pixels, use k_{seg} selects n^2 integers from [0, 255] as C_{2i}
4: for $\forall I_i'' \subset \mathcal{I}''$ do
5: $C_{1i} = I_i'' + C_{2i}$
6: end for

Algorithm 4 Image Encryption

Input: $\mathcal{I}$; $\mathcal{K}_{Enc}$
Output: $\mathcal{C}_1$ and $\mathcal{C}_2$
1: for $\forall I_i \subset I$ do
2: $I_i' = \text{BlockPMT}(I_i, k_{bp})$;
3: $I_i'' = \text{PixelPMT}(I_i', k_{pp})$;
4: $\mathcal{C}_1, \mathcal{C}_2 = \text{ImgSeg}\ (I_i'', k_{seg})$;
5: end for

After receiving the encrypted images, the cloud servers divide the images as the image owner does. S_1 calculates the pixel difference between the center of the block and the pixels around it as $(C_{1i} - C_{1j})$. S_2 calculates the pixel difference between the center of the block and the pixels around it as $(C_{2i} - C_{2j})$, then sends the differences to S_1. After obtaining the above information, S_1 subtracts the received differences from its own differences. We will explain the principles of mathematics as follow:

$$\begin{aligned}\left(C_{1i} - C_{1j}\right) &= \left(I_i'' + C_{2i}\right) - \left(I_j'' + C_{2j}\right) \\ &= \left(I_i'' - I_j''\right) + (C_{2i} - C_{2j}), \end{aligned} \tag{7}$$

$$\left(C_{1i} - C_{1j}\right) - \left(C_{2i} - C_{2j}\right) = \left(I_i'' - I_j''\right). \tag{8}$$

So S_1 gets the difference between the center of the image and the surrounding pixels in the image I''.

Then, the encrypted LBP feature is extracted from the encrypted image in the same way as in plaintext image. The only difference is that, in plaintext domain, the features are calculated from overlapping blocks, but in the encryption domain, the features are calculated from non-overlapping blocks. The whole process of feature extraction is defined in Algorithm 5. The feature extraction process is illustrated in Fig. 5.

Algorithm 5 Feature Extraction

Input: $\mathcal{C}_1$, $\mathcal{C}_2$ and size of blocks
1: At $\mathcal{C}_1$, use size of blocks to block the images.
2: for $\forall block$ do
3: $C_{1i} - C_{1j}$, C_{1i} is the center of the block, C_{1j} is an inner non center point
4: end for
5: At $\mathcal{C}_2$, use size of blocks to block the images.
6: for $\forall block$ do
7: $C_{2i} - C_{2j}$, C_{2i} is the center of the block, C_{2j} is an inner non center point
8: end for
9: S_2 sends all differences to S_1
10: for $\forall block$ do
11: $\left(C_{1i} - C_{1j}\right) - \left(C_{2i} - C_{2j}\right)$
12: end for

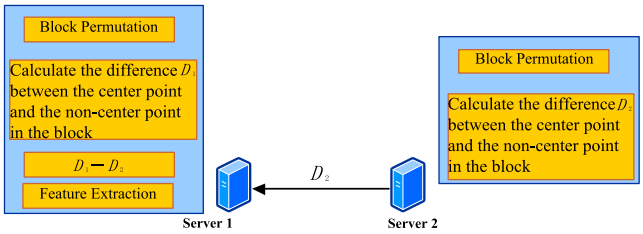

FIGURE 5. The process of feature extraction.

V. SECURITY ANALYSIS

The image content in our scheme is protected by three times of block permutation, pixel replacement and image segmentation. Our security proofs follow the paradigm in secure multi-party computations [37].

As shown in Fig. 6, for the ciphertext-only attack (COA) [38] model, we consider an ideal functionality $\mathcal{F}$ and the corresponding information leakages of our scheme. Ciphertext-only attack refers to exhaustive attack when only encrypted text can be accessed by adversaries. The execution of our scheme involves the interaction between cloud servers and users, which is defined as the real experiment. The honest-but-curious and independent cloud servers are defined as the adversary $\mathcal{A}$. The simulator $\mathcal{S}$ is defined to simulate the view of the adversary $\mathcal{A}$ by using the functionality $\mathcal{F}$ only.

A. SECURITY OF IMAGE CONTENT

The simulator $\mathcal{S}$ knows the number of images and the size of images. However, $\mathcal{S}$ can only fill the images with the random generated pixels.

In order to analyze the security of the image content, we observe the data of each cloud entity. For S_1, it only has the encrypted image set $\mathcal{C}_1$, and there are also differences $\left(C_{1i} - C_{1j}\right) - \left(C_{2i} - C_{2j}\right)$ between the center point within the block and the non-center point within the block after block permutation and pixel permutation have been performed. As long as the client generates a new key each time to encrypt the image, the image segmentation mechanism provides security for the image content. The strength of the ciphertext security depends on the security of the pseudorandom generator function used. $\left(C_{1i} - C_{1j}\right) - \left(C_{2i} - C_{2j}\right)$ is already encrypted by image texture protection. Its security strength is $256^{blocknum}$.

The texture information of an image is protected by block and pixel permutation. The security strengths of block permutation is equal to $blocknum!$. The security strengths of pixel permutation is equal to $blocknum \times (8!)$, respectively.

The ideal functionality $\mathcal{F}$ of our scheme as well as information leakages.

$\mathcal{F}$: StoreImage($\mathcal{I}$)

Functionality. Image owner encrypts the images $\mathcal{I}$, and generates two sets of encrypted images $\mathcal{C}_1$ and $\mathcal{C}_2$. Then, image owner uploads $\mathcal{C}_1$ to the cloud server $\mathcal{S}_1$ and $\mathcal{C}_2$ to the cloud server $\mathcal{S}_2$.

Storage leakage. The informations leaked here includes $\mathcal{C}_1$ and $\mathcal{C}_2$, the size of images, and the number of images.

$\mathcal{F}$: ExtractFeature($\mathcal{C}_1$, $\mathcal{C}_2$)

Functionality. Cloud servers extract LBP features from encrypted images in $\mathcal{C}_1$ and $\mathcal{C}_2$.

Feature leakage. The information leaked here includes the encrypted LBP histograms, the similarities among the LBP histograms, and the distributions of LBP histograms.

FIGURE 6. The functionality F and information leakage in our framework.

The image content is made up of all of these information and the security strength of permutation in our scheme can be calculated as: $blocknum! + blocknum \times (8!)$.

For S_2, it only has the encrypted image set $\mathcal{C}_2$. The encrypted image set $\mathcal{C}_2$ does not contain any image information.

B. SECURITY OF FEATURES

In our scheme, the LBP histograms are calculated from encrypted grayscale values of images. Due to the pixel permutation in blocks, the extracted LBP histograms are also the permutated ones. In this case, the cloud server cannot extract valid LBP features without the secret permutation keys. Otherwise, the server can guess the content of encrypted images by searching the database with the LBP features generated from the specially selected images. With a simulated image I^S, $\mathcal{S}$ can simulate LBP histograms of the simulated images. The computational complexity of a distinguisher D in distinguishing the histogram is 256! which means a 1684 bits security strength.

VI. PERFORMANCE ANALYSIS

This section evaluates the performance of the proposed scheme in terms of encryption effectiveness, retrieval accuracy and face recognition rate. We implement the proposed scheme with MatLab 2014 on a Linux operation system with Intel(R) Core(TM) i7-6800K CPU @ 3.40GHz and 16 G memory.

A. EFFECTIVENESS OF IMAGE ENCRYPTIONS

In our protocol, the images are encrypted by block permutation, pixel permutation and image segmentation. Fig. 7 shows

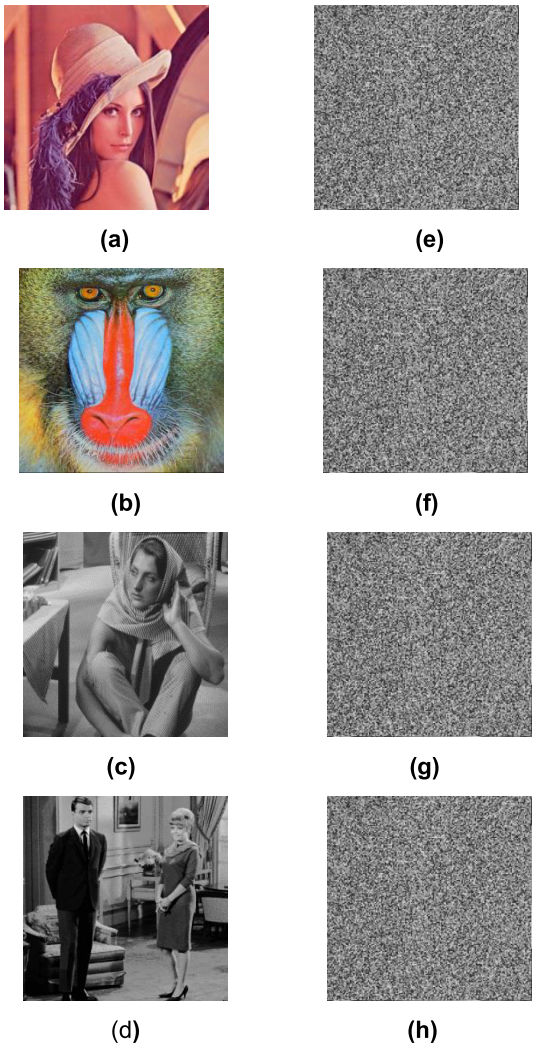

FIGURE 7. The visual effect. (a) ∼ (d) The visual effect of the original image. (e) ∼ (h) The visual effect of the encrypted image.

the encryption of the image. The Fig. 7 (a) ∼ (d) show the original images. They contain different shades of color and gray images. The Fig. 7 (e) ∼ (h) show the visual effect of the encrypted image. Encryption operations are based on grayscale images. Because the image segmentation will make the gray value greater than 255, the encrypted image will be a white blank image, where the normalized method is used to display the image likes Fig. 7 (e) ∼ (h). Obviously it is impossible to guess the image content from the encrypted image.

The Fig. 8 (a) ∼ (d) show the original LBP. We found that the original LBP will leak a lot of image texture information. The Fig. 8 (e) ∼ (h) show the LBP after pixel permutation and image segmentation in the encryption process. Obviously it is possible to guess the image texture without block permutation. In particular, the texture information of the face image is more pronounced likes the Fig. 8 (e) and (g). The Fig. 8 (i) ∼ (l) show the PPLBP which will not reveal the image texture information. So we use block permutation to

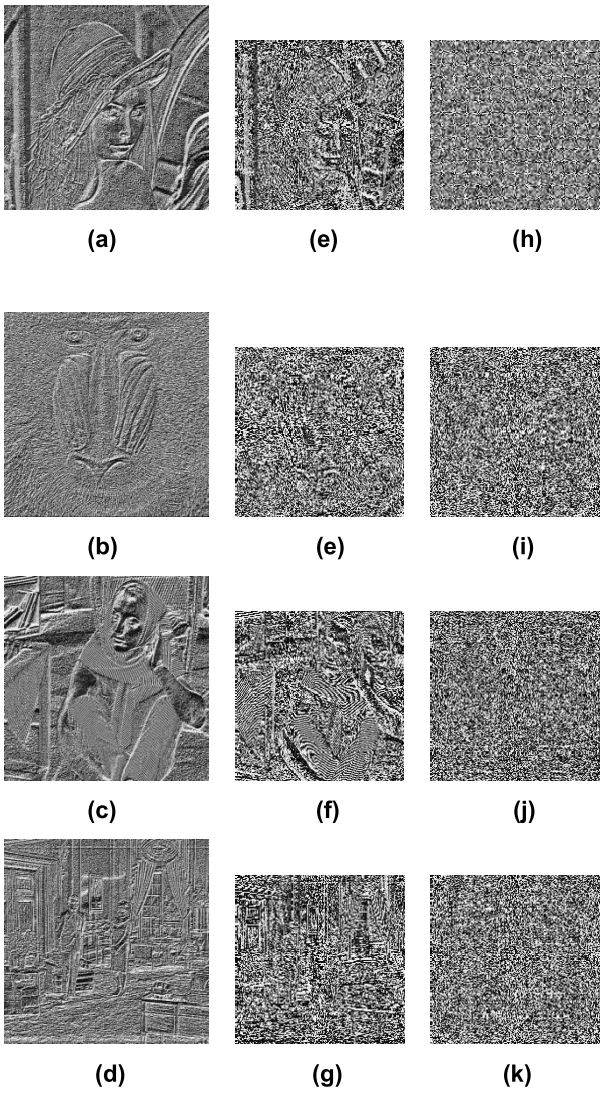

FIGURE 8. The visual effect of LBP (The size of the encrypted LBP is 1/9 the size of the original LBP). (a) ∼ (d) The visual effect of the original LBP. (e) ∼ (h) The visual effect of the LBP without block permutation. (i) ∼ (l) The visual effect of PPLBP.

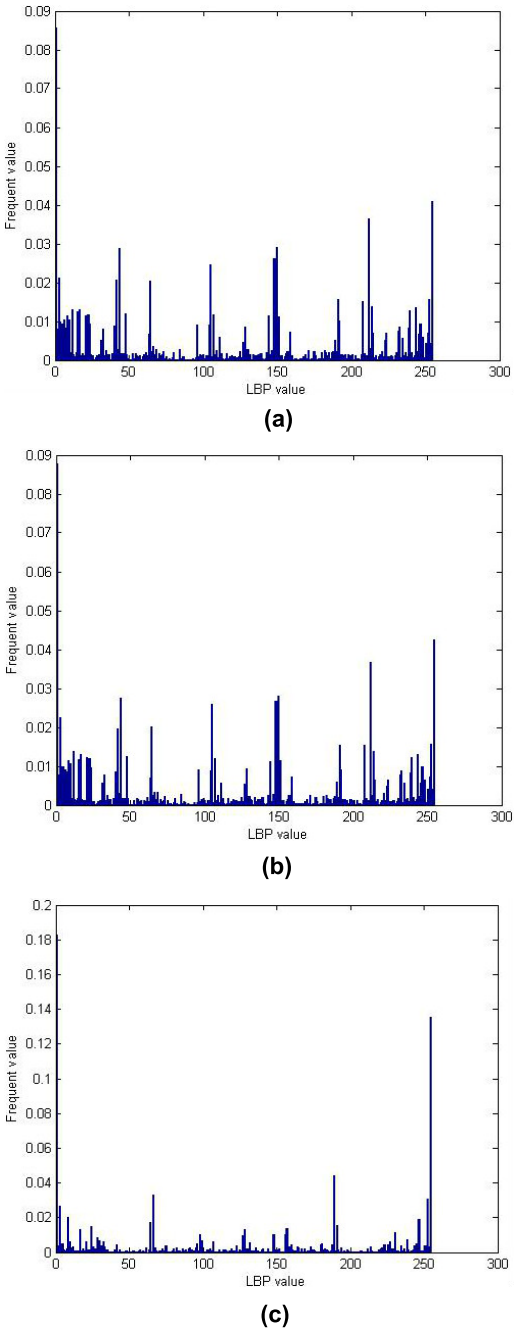

FIGURE 9. The histogram of LBP of Fig. 8 (a). (a) The histogram of the original LBP. (b) The histogram of the LBP without pixel permutation. (c) The histogram of PPLBP.

protect the texture information of the image. Since the size of the image after encryption is only one-ninth of the original size, we have magnified the size of Fig. 8 (e) ∼ (l) for more intuitive.

B. HISTOGRAMS COMPARISON

In the application, LBP histograms are used. As shown in Fig. 9 is LBP histograms of Fig. 8 (a) with different encryption schemes. We normalized the histograms to facilitate analysis of the results. Fig. 9 (a) shows the histogram of the original LBP. Fig. 9 (b) shows the histogram of the LBP without pixel permutation. We can find out by comparing the histograms that there are some differences between the two, but they are very similar. Fig. 9 (c) shows the histogram of PPLBP. Obviously, the histogram of PPLBP differs greatly from the previous two. So we think pixel permutation can protect LBP values and histograms very well.

C. APPLICATION IN IMAGE RETRIEVAL

In the process of image retrieval, the user usually provides a sample image (Query by Example). The query system extracts the features of the query image and compares them with the features in the database. Finally, an image with similar characteristics to the query image is returned to the user.

TABLE 1. The mAPs of image retrieval.

Schemes	PPLBP	Original LBP
mAP	0.28998	0.32293

The standard Manhattan Distance is utilized to quantify the distance between the query image and the image in the cloud. In our experiment, mean average precision (mAP) is used to measure the retrieval accuracy. The Python evaluation package of Inria Holidays Dataset is directly used to calculate the map of the proposed method. Inria Holidays database [39] is used as the retrieval accuracy experiment database which contains 1491 color images with the size of 400533pixels. As show in Table 1, the mAP of our scheme is 0.28998, the mAP of the original LBP is 0.32293. The difference between the two mAP values is small, so we believe that the encryption operation does not affect the effect of the LBP feature in our scheme and our PPLBP can be applied to image retrieval.

VII. CONCLUSION

Smart campus can enjoy rich cloud computing resources by outsourcing image features extraction to cloud computing platforms. However, there are few solutions for outsourcing secure image features extraction. In this paper, we propose a secure out-sourcing LBP scheme. The proposed scheme uses block permutation, pixel permutation and image segmentation to protect the privacy of outsourced images. The secure LPB features can be extracted by the server from these encrypted images. The secure LBP features can be used directly for many applications and retain the most characteristics of the original LBP features. We analyzed and evaluated the safety of the proposed scheme. We further conducted a large number of experiments on the proposed scheme. The experimental results show that the secure LBP in this scheme can be used in image retrieval and face recognition. In the future, we plan to research more secure extraction of image features in the field of encryption.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," *Nat. Inst. Standards Technol.*, vol. 53, no. 6, p. 50, 2011.
[2] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in *Proc. Int. Conf. Theory Cryptogr.*, 2005, pp. 264–282.
[3] M. J. Atallah, F. Kerschbaum, and W. Du, "Secure and private sequence comparisons," in *Proc. ACM Workshop Privacy Electron. Soc.*, 2003, pp. 39–44.
[4] S. Salinas, C. Luo, X. Chen, and P. Li, "Efficient secure outsourcing of large-scale linear systems of equations," in *Proc. IEEE Conf. Comput. Commun.*, Apr./May 2015, pp. 1035–1043.
[5] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k-nearest neighbor query over encrypted data in outsourced environments," in *Proc. IEEE 30th Int. Conf. Data Eng. (ICDE)*, Mar./Apr. 2014, pp. 664–675.
[6] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Homomorphic encryption-based secure SIFT for privacy-preserving feature extraction," *Proc. SPIE*, vol. 7880, p. 788005, Feb. 2011.
[7] S. Wang, M. Nassar, M. Atallah, and Q. Malluhi, "Secure and private outsourcing of shape-based feature extraction," in *Information and Communications Security*, vol. 8233. Cham, Switzerland: Springer, 2013, pp. 90–99.
[8] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Prague, Czech Republic, 1999, pp. 223–238.
[9] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Proc. Conf. Cryptol.*, vol. 6841. Berlin, Germany: Springer, 2011, pp. 505–524.
[10] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-Maillet, "A privacy-preserving framework for large-scale content-based information retrieval," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 152–167, Jan. 2015.
[11] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "SCiFI—A system for secure face identification," in *Proc. IEEE Secur. Privacy*, May 2010, pp. 239–254.
[12] K. U. Sharma and N. V. Thakur, "A review and an approach for object detection in images," *Int. J. Comput. Vis. Robot.*, vol. 7, nos. 1–2, pp. 196–237, 2017.
[13] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards privacy-preserving content-based image retrieval in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 276–286, Jan./Mar. 2018.
[14] Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, "EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing," *Inf. Sci.*, vol. 387, pp. 195–204, May 2017.
[15] Y. Cao, Z. Zhou, X. Sun, and C. Gao, "Coverless information hiding based on the molecular structure images of material," *Comput. Mater. Continua*, vol. 54, no. 2, pp. 197–207, 2018.
[16] C. Yuan, X. Li, Q. M. J. Wu, J. Li, and X. Sun, "Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis," *Comput. Mater. Continua*, vol. 53, no. 4, pp. 357–371, 2017.
[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[18] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC*, vol. 9, 2009, pp. 169–178.
[19] 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.
[20] 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*, 2014, pp. 497–506.
[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2005, pp. 886–893.
[22] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer-Verlag, 2006, pp. 404–417.
[23] 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.
[24] Q. Wang, J. Wang, S. Hu, Q. Zou, and K. Ren, "SecHOG: Privacy-preserving outsourcing computation of histogram of oriented gradients in the cloud," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 257–268.
[25] Q. Wang, S. Hu, J. Wang, and K. Ren, "Secure surfing: Privacy-preserving speeded-up robust feature extractor," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Jun. 2006, pp. 700–710.
[26] P. Li, T. Li, Z.-A. Yao, C.-M. Tang, and J. Li, "Privacy-preserving outsourcing of image feature extraction in cloud computing," *Soft Comput.*, vol. 21, no. 15, pp. 4349–4359, 2017.
[27] E. Bresson, D. Catalano, and D. Pointcheval, "A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications," in *Proc. Int. Conf. Theory Appl.Cryptol. Inf. Secur.*, Taipei, Taiwan, vol. 2894, 2003, pp. 37–54.
[28] P. Bogetoft *et al.*, "Secure multiparty computation goes live," in *Proc. Int. Conf. Financial Cryptogr. Data Secur. (FC)*, Accra Beach, Barbados, vol. 5628, Feb. 2009, pp. 325–343.
[29] L. Jiang, C. Xu, X. Wang, B. Luo, and H. Wang, "Secure outsourcing SIFT: Efficient and privacy-preserving image feature extraction in the encrypted domain," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 8, p. 1, 2017.

[30] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Proc. Int. Symp. Algorithmic Number Theory.*, vol. 1423. New York, NY, USA: Springer-Verlag, 1998, pp. 267–288.
[31] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[32] Z. Shi, F. Ye, Q. He, and Z. Shi, "Symmetrical invariant LBP texture descriptor and application for image retrieval," in *Proc. Congr. Image Signal Process. (CISP)*, vol. 2, May 2008, pp. 825–829.
[33] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block LBP representation," in *Advances in Biometrics* (Lecture Notes in Computer Science), vol. 4642. Berlin, Germany: Springer, 2007, pp. 11–18.
[34] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved LBP under Bayesian framework," in *Proc. Int. Conf. Image Graph.*, Dec. 2004, pp. 306–309.
[35] B. Ferreira, J. Rodrigues, J. Leitão, and H. Domingos, "Privacy-preserving content-based image retrieval in the cloud," in *Proc. IEEE 34th Symp. Reliable Distrib. Syst.*, Sep./Oct. 2015, pp. 11–20.
[36] A. C. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. Symp. Found. Comput. Sci.*, Nov. 1982, pp. 160–164.
[37] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. IEEE Symp. Found. Comput. Sci.*, Oct. 2001, pp. 136–145.
[38] A. Biryukov, "Ciphertext-only attack," in *Computer Science and Communications Dictionary*. Boston, MA, USA: Springer, 2005, p. 207.
[39] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vision.*, vol. 5302. New York, NY, USA: Springer-Verlag, pp. 304–317.

ZHIHUA XIA (M'14) received the B.E. degree from Hunan City University, China, in 2006, and the Ph.D. degree in computer science and technology from Hunan University, China, in 2011. He is currently an Associate Professor with the School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include cloud computing security and digital forensic.

XIAOHE MA received the B.E. degree with Harbin Engineering University, China, in 2016. She is currently pursuing the M.S. degree in computer science and technology with the College of Computer and Software, Nanjing University of Information Science and Technology, China. His research interests include network and information security.

ZIXUAN SHEN is currently pursuing the bachelor's degree in computer science and technology with the College of Computer and Software, Nanjing University of Information Science and Technology, China. Her research interests include convolutional neural networks.

XINGMING SUN received the B.S. degree in mathematics from Hunan Normal University, Hunan, China, in 1984, the M.S. degree in computing science from the Dalian University of Science and Technology, Dalian, China, in 1988, and the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2001. He is currently a Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. His research interests include network and information security, digital watermarking, cloud computing security, and wireless network security.

NEAL N. XIONG received the Ph.D. degree in sensor system engineering from Wuhan University and the Ph.D. degree in dependable sensor networks from the Japan Advanced Institute of Science and Technology, respectively. He was with Georgia State University, the Wentworth Technology Institution, and Colorado Technical University (a Full Professor about five years) about 10 years. He is current an Associate Professor (third year) with the Department of Mathematics and Computer Science, Northeastern State University, OK, USA. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory.

BYEUNGWOO JEON received the B.S. degree *(magna cum laude)* in 1985, the M.S. degree in electronics engineering from Seoul National University, Seoul, South Korea, in 1987, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, in 1992. From 1993 to 1997, he was with the Signal Processing Laboratory, Samsung Electronics, South Korea, where he conducted research and development into video compression algorithms, the design of digital broadcasting satellite receivers, and other MPEG-related research for multimedia applications. Since 1997, he has been with the Faculty of the School of Electronic and Electrical Engineering, Sungkyunkwan University, South Korea, where he is currently a Full Professor. From 2004 to 2006, he served as a Project Manager of digital TV and broadcasting with the Korean Ministry of Information and Communications, where he supervised all digital TV-related research and development in South Korea. He has authored many papers in the areas of video compression, pre$\diagup$post processing, and pattern recognition. His research interests include multimedia signal processing, video compression, statistical pattern recognition, and remote sensing. He is a member of the Tau Beta Pi, the Eta Kappa Nu, SPIE, IEEK, KICS, and KSOBE. He was a recipient of the 2005 IEEK Haedong Paper Award in the Signal Processing Society, South Korea.