This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2016.2590944, IEEE Transactions on Information Forensics and Security

A Privacy-preserving and Copy-deterrence Content-based Image Retrieval Scheme in Cloud Computing

Zhihua Xia, *Member, IEEE,* Xinhui Wang, Liangao Zhang, Zhan Qin, *Member, IEEE,* Xingming Sun, *Senior Member, IEEE,* and Kui Ren, *Fellow, IEEE*

***Abstract*—With the increasing importance of images in people's daily life, Content-based Image Retrieval (CBIR) has been widely studied. Compared with text documents, images consume much more storage space. Hence, its maintenance is considered to be a typical example for cloud storage outsourcing. For privacy-preserving purposes, sensitive images, such as medical and personal images, need to be encrypted before outsourcing, which makes the CBIR technologies in plaintext domain to be unusable. In this paper, we propose a scheme that supports CBIR over encrypted images without leaking the sensitive information to the cloud server. Firstly, feature vectors are extracted to represent the corresponding images. After that, the pre-filter tables are constructed by locality-sensitive hashing to increase search efficiency. Moreover, the feature vectors are protected by the secure kNN algorithm, and image pixels are encrypted by a standard stream cipher. In addition, considering the case that the authorized query users may illegally copy and distribute the retrieved images to someone unauthorized, we propose a watermark-based protocol to deter such illegal distributions. In our watermark-based protocol, a unique watermark is directly embedded into the encrypted images by the cloud server before images are sent to the query user. Hence, when an illegal image copy is found, the unlawful query user who distributed the image can be traced by the watermark extraction. The security analysis and experiments show the security and efficiency of the proposed scheme.**

***Index Terms*—Searchable encryption, content-based image retrieval, secure kNN, copy deterrence, watermark.**

I. Introduction

With the development of the imaging devices, such as digital cameras, smartphones, and medical imaging equipments, our world has been witnessing a tremendous growth in quantity, availability, and importance of images. The needs of efficient image storage and retrieval services are reinforced by the increase of large-scale image databases among all kinds of areas. Meanwhile, after more than twenty years of development, CBIR techniques show the potential of usefulness in many real-word applications. For example, clinicians can use CBIR to find similar cases of patients and facilitate clinical decision-making processes. However, a large image database usually consists of millions of images. Therefore, CBIR services typically incur high storage and computation complexities. Cloud computing offers a great opportunity for the on-demand access to ample computation and storage resources, which makes it an attractive choice for the image storage and CBIR outsourcing. By outsourcing CBIR services to the cloud server, the data owner is relieved from maintaining local image database and interacting with database users online.

Despite the tremendous benefits, image privacy becomes the main concern with CBIR outsourcing. For example, patients may not want to disclose their medical images to any others except to a specific doctor in medical CBIR applications. To formulate the problem, this paper considers two types of privacy threats. Firstly, a curious cloud server may look into the owner's database for additional information. Secondly, after receiving the retrieved images, the query user may illegally distribute these images to someone unauthorized for benefits.

Contribution. This paper protects the privacy of image data in CBIR outsourcing applications against a curious cloud server and the dishonest query users. The main contributions are summarized as follows:

1) An index of two layers is constructed. Four typical visual descriptors, which are defined in MPEG-7, are employed in our scheme. Meanwhile, the feature vectors are encrypted by the secure kNN algorithm.
2) The existing searchable encryption schemes usually consider that the query users are fully trustworthy. This is not necessarily true in real-world applications. To the best of our knowledge, this paper is the first work that proposes a searchable encryption scheme, considering the dishonest query users who may distribute the retrieved images to those who are unauthorized. A watermark-based protocol is designed for the copy-deterrence purpose. Specifically, after completing the search operation requested by an image user, a unique watermark associated with the image user is imperceptibly embedded into the retrieved images. Then, the watermarked images are sent to the image user. When an illegal copy of the image is found, the unlawful query user who made the illegal distribution can be traced by the watermark extraction. This will help to deter the illegal distribution.
3) An elaborate watermark-based protocol in the encryption domain is designed for copy-deterrence in a cloud com-

Zhihua Xia, Xinhui Wang, Liangao Zhang and Xingming Sun are with Jiangsu Engineering Center of Network Monitoring, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, and School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, China. (E-mails: xia_zhihua@163.com, wxh_nuist@163.com, zlo2010@163.com, sunnudt@163.com.

Zhan Qin and Kui Ren is with the Department of Computer Science and Engineering, State University of New York at Buffalo. (E-mails: zhanqin@buffalo.edu, kuiren@buffalo.edu).

puting scenario. Different from common watermarking techniques, the proposed protocol needs to embed the watermark directly into the encrypted images via the cloud server. After receiving the encrypted and watermarked images, the query user needs to decrypt the images directly. And the decryption should not affect the watermark in the images.

The rest of this paper is organized as follows. Section II introduces the related works. Section III gives a brief introduction to the system and threat models, design goals, and preliminaries. The proposed scheme is described in Section IV. The security of the scheme is analyzed in Section V. The performance evaluations are presented in Section VI. Section VII gives the conclusions.

II. Related works

Searchable encryption (SE) schemes enable the query user to search over the encrypted data collections. Most of the existing SE schemes focus on the retrieval of text documents. Some early schemes explore the Boolean search to identify whether or not a query term is present in the encrypted text documents [1]–[5]. Afterwards, plenty of methods have been proposed under different threat models to achieve various search functionalities, such as similarity search [6]–[8], multi-keyword ranked search [9]–[12], dynamic search [11], [13]–[15], etc. However, few of these schemes are straightforwardly feasible to an image retrieval task. Shashank *et al*. [16] propose a Private Content-based Image Retrieval (PCBIR) scheme which protects the privacy of the query image, but exposing the unencrypted image database to the server directly. Some researchers outsource the computation of image feature extraction to the cloud server in a privacy-preserving manner [17]–[21], which can be the key techniques to the privacy-preserving CBIR outsourcing. Nevertheless, the index construction and similar search on the encrypted features need to be further addressed. In addition, the homomorphic-encryption based schemes usually incur high computation and storage burden [17]–[19].

In the area of privacy-preserving CBIR schemes, Lu *et al*. [22] constructed the first privacy-preserving CBIR scheme over the encrypted images. The authors extracted the visual words to represent the images, and then calculated the Jaccard similarity between the two sets of visual words so as to evaluate the similarity between the two corresponding images. The order-preserving encryption and min-hash algorithm are employed to protect the information of the visual words. In another work, Lu *et al.* [23] investigated three image feature protection techniques, i.e. the bitplane randomization, random projection, and randomized unary encoding. The features encrypted with the bitplane randomization and the randomized unary encoding can be used to calculate the Hamming distance in the encryption domain. The features encrypted with the random projection can be used to calculate the L1 distance in the encryption domain. Cheng *et al.* [24] designed a CBIR system by combining the bitplane randomization and random projection. Xia *et al.* [25] proposed a privacy-preserving CBIR scheme using local features and earth mover's distance (EMD). A linear transformation is applied to protect the sensitive information in the calculation of EMD. Ferreira *et al.* [26] proposed an image encryption method which is suitable for the privacy-preserving CBIR outsourcing. In [26], the texture information is separated from the color information. The texture information is encrypted by a probabilistic cryptosystem to protect the image content, but the color information is encrypted by a deterministic cryptosystem to support the color-feature based CBIR. Cheng *et al.* [27] proposed a markov-process based retrieval scheme for encrypted images. The image data is encrypted by a stream cipher, and the markov features can be directly extracted from encrypted data.

The aforementioned works make good steps to the CBIR over encrypted images. However, the search efficiency still remains to be the main issue. Nevertheless, none of these schemes consider the dishonest query users who may illegally distribute the retrieved images. Actually, it is difficult to design a method to completely prevent illegal distributions. However, it is possible to design certain techniques to deter such illegal behaviors. Watermarking techniques have been widely studied for the copy deterrence in buyer-seller scenarios [28]–[32]. For the copy-deterrence purpose, the seller inserts a unique watermark into the image before it being sold to the buyer. If the buyer distributes the copies of the watermarked image, the illegal buyer can be traced by examining the watermark in image. The watermarking techniques can prevent the illegal distributions to some extent. However, there are still several problems that need to be settled to implement watermark-based copy deterrence in our scheme:

1) For the copy-deterrence purpose, we need to embed watermark into all the retrieved images for each query request. It requires high computational complexity and thus is expected to be completed by the cloud server. Therefore, we need to employ a watermarking method that is efficient and can embed the watermark directly in the encrypted image by the cloud server. After receiving the encrypted and watermarked images, the query user should be able to directly decrypt the watermarked images with the predefined secret keys. After the decryption, the watermark is still contained in the image.
2) In a watermark-based copy deterrence protocol, the image owner may slander a query user by embedding the watermark related to the user in an original image. The proposed protocol needs to prevent this type of illegal behavior.
3) After obtaining the watermarked images, the query user may change them by regular image processing operations before illegal distributions, e.g., JPEG-compression. In this case, the watermark bits could not be extracted with 100% accuracy. Thus, the trace with the extraction errors needs to be fully discussed.

III. Problem formulation

A. System model

The system model in this paper involves four different types of entities: the image owner, image user, cloud server and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2016.2590944, IEEE Transactions on Information Forensics and Security

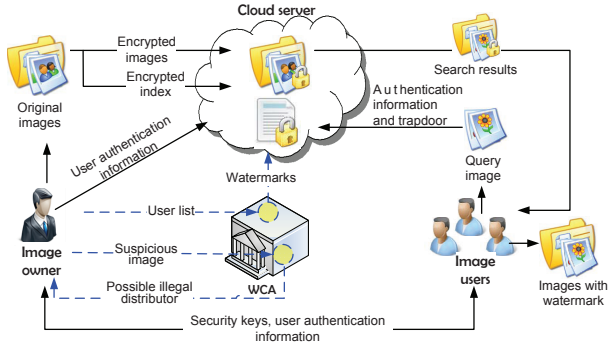


Fig. 1. Framework of the privacy-preserving and copy-deterrence CBIR scheme

watermark certification authority (WCA), as illustrated in Fig. 1.

Image owner wants to outsource his local data, i.e., a collection of n images $\mathcal{M} = \{m_1, m_2, ..., m_n\}$, to the cloud server in the encrypted form $\mathcal{C} = \{c_1, c_2, ..., c_n\}$, while enabling the ability to search over the encrypted images. Firstly, the image owner extracts the feature vectors $\mathcal{F} = \{f_1, f_2, ..., f_n\}$ from $\mathcal{M}$, and then constructs a secure searchable index $\mathcal{I}$ on $\mathcal{F}$. Next, both the encrypted image collection $\mathcal{C}$ and index $\mathcal{I}$ are outsourced to the cloud server. The image owner also takes the responsibility to authorize image users through a certain secure method, which is orthogonal to our schemes and will not be discussed in this paper as many previous SSE schemes [2]–[4], [6]–[14], [22]–[24], [26], [27], [33]. The image owner sends the authentication information of authorized users to the cloud server who will take the responsibility to verify the identity of user in search requests. In addition, the image owner sends the identities of the authorized users to WCA for watermark generation.

In our scheme only a single image owner is considered. However, if there are multiple image owners in our scheme and all the owners have the same set of users, the owners can encrypt the indexes under the same cryptosystem and secret keys so that the users can search images from all of these owners. But if the sets of authorized users are different for each of image owners, the owners need to encrypt their images and indexes with their particular keys. Accordingly, the user can only search from the corresponding owners. In addition, if some image owners share a part of users, one can resort to some sophisticated methods to efficiently manage the authorization of users. Attribute-based encryption methods could be a good choice.

Image users are the authorized ones to retrieve images from the cloud server. To request a search, the image user firstly generates a trapdoor *TD* for the query image, and then submits the trapdoor *TD* and his identity to the cloud server. After receiving the resulting images, the user can decrypt them with the secret key shared by the image owner.

Cloud server stores the encrypted image collection $\mathcal{C}$ and the index $\mathcal{I}$ for the image owner and processes the query requests from image users. Besides, in order to support copy deterrence, the cloud server takes the responsibility to embed the watermark into the retrieved images.

Watermark certification authority (WCA) is a trusted agency who takes the responsibilities to generate watermarks for the authorized query users and execute the arbitration through the watermark extraction algorithm.

B. Threat model

In our scheme, all the image owner, image user, and cloud server could trigger security problems. In this paper, two types of security issues are mainly considered.

Data privacy. Similar to the previous SSE schemes [2]–[4], [6]–[14], [22]–[27], [33], we consider the cloud server to be "honest-but-curious", which means the cloud server correctly follows the protocol specification, but keeps and analyzes the communication data so as to obtain the sensitive information. Thus, the privacy of the image content, image features and trapdoors needs to be properly protected.

Copyright. In the proposed scheme, we consider the dishonest image users who correctly follow the protocol specification, but may distribute the retrieved images to the unauthorized others for benefits. The watermarking technology is adopted to deter the illegal distribution. In addition, in a watermark-based protocol, the data owner may frame an innocent user by embedding the user's watermark into the original images. This kind of behavior should be prevented in the proposed scheme. To the best of our knowledge, our work is the first SSE scheme that takes the dishonest user into consideration.

We assume that there is no collusion among the cloud server, image owner, image user, and WCA. This assumption is necessary to construct an efficient scheme. Specifically, the cloud server and WCA will not leak the watermarks, the embedding algorithm, and the secret keys to the image owner. The image users will not leak the secret keys used in the trapdoor generation and image decryption to the cloud server. In addition, just as the previous SSE schemes [2]–[4], [6]–[9], [11], [13], [14], [22]–[27], [33], it is easy to deduce that the images m_i and m_j are similar to each other if both the images m_i and m_j have the high similarity scores to the same query image. This type of information leakage is not considered in this paper.

C. Design goals

Efficiency. The linear search is quite inefficient and computationally impracticable for a large database. The proposed scheme aims to achieve a better-than-linear search efficiency through constructing an efficient index. For the copy-deterrence purpose, the cloud server needs to watermark all of the retrieved images for each query. Thus, a high efficient watermark algorithm is preferred in a CBIR scenario.

Security. According to the threat model, the proposed scheme is expected to meet the following security requirements:

1) *Data privacy.* The plaintext data regarding the image content, image features, and trapdoors needs to be kept unknown to the cloud server.
2) *Copy deterrence.* A watermark-based protocol in encryption domain needs to be designed to deter the illegal

distribution, and the image owner is prevented from framing the image users.

D. Preliminaries

MPEG-7 visual descriptors. MPEG-7 is a multimedia content description standard which offers a comprehensive set of descriptors for the multimedia data description [34]. In this paper, four MPEG-7 descriptors are utilized:

1) Scalable color descriptor (SCD) is defined in the hue-saturation-value (HSV) color space. SCD uses a Haar transform encoding that facilitates the scalability for the feature extraction.
2) Color structure descriptor (CSD) aims to identify the localized color distribution using a small structuring window. To ensure interoperability, the color structure histogram is constructed in the hue-min-max-difference (HMMD) color space.
3) Color layout descriptor (CLD) provides information about the spatial color distribution within images. After an image is divided into 64 blocks, CLD descriptor is extracted from each of these blocks based on the discrete cosine transform.
4) Edge histogram descriptor (EHD) captures the spatial distribution of edges. The distribution of edges is a good texture signature for the image matching even when the underlying texture is not homogeneous.

These descriptors can be represented as feature vectors, and the image similarity could be measured by Euclidean distance between the feature vectors. For more specific descriptions about these descriptors, please refer to [34].

Locality-sensitive Hashing. Locality-sensitive hashing (LSH) has the property that close items will collide with a higher probability than distant ones, which can be applied in approximate queries [35]. A hash function family $\mathcal{H} = \{h : \mathcal{S} \to \mathcal{U}\}$ is called (c, cr, p_1, p_2)-sensitive for any $x, y \in \mathcal{S}$ if

$$\begin{cases} Pr\{h(x) = h(y)\} \geq p_1 & for \quad d(x,y) \leq r \\ Pr\{h(x) = h(y)\} \leq p_2 & for \quad d(x,y) \geq cr, \end{cases} \tag{1}$$

where the constant $c > 1$ and probabilities $p_1 > p_2$.

To enlarge the gap between p_1 and p_2, multiple hash functions can be jointed to construct another function family $\mathcal{G} = \{g : \mathcal{S} \to \mathcal{U}^\lambda\}$ where $g(v) = (h_1(v), h_2(v), ...h_\lambda(v)), h_i \in \mathcal{H}$ is the concatenation of λ LSH functions. In practice, multiple hash tables can be constructed with multiple $g_i \in \mathcal{G}$. The final set of results is a union from these multiple hash tables.

In our scheme, LSH based on the p-stable distribution is utilized to construct the pre-filter tables. A p-stable LSH $h_{a,b} : R^l \to Z$ maps an l-dimensional vector v into an integer [35], and can be formulated as $h_{a,b}(v) = \lfloor \frac{(a \cdot v + b)}{r} \rfloor$, where a is an l-dimensional random vector with the entries following a p-stable distribution, b is a real number chosen uniformly from the range $[0, r)$, and r is an positive integer.

IV. The Proposed scheme

A. Overview of the proposed scheme

The proposed scheme consists of a tuple of algorithms that are executed by different entities. A brief description of the algorithms is presented in Fig. 2.

Image owner side:

- $\mathcal{K} \leftarrow \mathsf{KeyGen}(1^\kappa)$ is the key generation algorithm that takes as input the security parameter κ, and returns the secret key set $\mathcal{K}$.
- $\mathcal{I} \leftarrow \mathsf{IndexGen}(\mathcal{K}, \mathcal{M})$ is the index generation algorithm that takes as input the secret key set $\mathcal{K}$ and the image collection $\mathcal{M}$, and returns the index $\mathcal{I}$.
- $\mathcal{C} \leftarrow \mathsf{ImgEnc}(\mathcal{K}, \mathcal{M})$ is the image encryption algorithm that takes as input the secret key set $\mathcal{K}$ and the image collection $\mathcal{M}$, and returns the encrypted image collection $\mathcal{C}$.

Image User side:

- $TD \leftarrow \mathsf{TrapdoorGen}(\mathcal{K}, m_q)$ is the trapdoor generation algorithm that takes as input the key set $\mathcal{K}$ and the query image m_q, and returns the query trapdoor TD.
- $\mathcal{M}_q \leftarrow \mathsf{ImgDec}(\mathcal{K}, \mathcal{R}')$ is the decryption algorithm that takes as input the secret key set $\mathcal{K}$ and a retrieved set of encrypted and watermarked images $\mathcal{R}'$, and returns the set of watermarked images $\mathcal{M}_q$.

Cloud server side:

- $\mathcal{R} \leftarrow \mathsf{Search}(\mathcal{I}, \mathcal{C}, TD)$ is the search algorithm that takes as input the encrypted collection $\mathcal{C}$, the index $\mathcal{I}$, and the trapdoor TD, and returns the temporary search result set $\mathcal{R}$.
- $\mathcal{R}' \leftarrow \mathsf{WatermarkEmb}(\mathcal{R}, w)$ is the watermark embedding algorithm that takes as input the temporary search result set $\mathcal{R}$ and the watermark w, and returns the set of watermarked images $\mathcal{R}'$.

WCA side:

- $\{w_i\} \leftarrow \mathsf{WatermarkGen}(\{\mathsf{UID}_i\})$ is the watermark generation algorithm that generates a unique watermark w_i for each UID_i.
- $w_t \leftarrow \mathsf{WatermarkExtra}(m_t, m_o)$ is the watermark extraction algorithm that takes as input doubtable image m_t and its original version m_o, and returns the extracted watermark w_t which is used for the arbitration.

Fig. 2. Overview of the algorithms in the proposed scheme

Given an image collection $\mathcal{M}$, the image owner runs KeyGen, IndexGen, and ImgEnc to generate the set of secret keys $\mathcal{K}$, the secure index $\mathcal{I}$, and the encrypted image collection $\mathcal{C}$, respectively. After that, the image owner outsources the index $\mathcal{I}$ and the collection $\mathcal{C}$ to the cloud server, and then sends the key set $\mathcal{K}$ to the authorized image users. In addition, the image owner sends the set of user identities $\{\mathsf{UID}_i\}$ to WCA. After receiving $\{\mathsf{UID}_i\}$, WCA generates a unique watermark w_i for each query user by WatermarkGen, and then sends the set of watermarks $\{w_i\}$ to the cloud server.

In order to retrieve similar images, the authorized image user runs TrapdoorGen to generate a query trapdoor TD, and then submits the trapdoor TD, UID, and authentication key to the cloud server. Upon receiving the search request, the cloud server firstly verify the identity of the user with the UID and authentication key. If successfully verified, the cloud server runs Search to obtain a temporary result set $\mathcal{R}$ including the top-k most similar images. Next, the cloud server finds the watermark w according to the user's UID, and embeds the watermark w into each of the images in $\mathcal{R}$ by WatermarkEmb. Finally, the watermarked image set $\mathcal{R}'$ is generated and sent to the query user. After receiving $\mathcal{R}'$, the query user runs ImgDec to obtain the set of decrypted images $\mathcal{M}_q$. Please note that the decrypted images will still contain the watermark in them.

If an image m_t in $\mathcal{M}_q$ is illegally distributed by a query user, and then found by the image owner, the owner can submit m_t and its original version m_o to WCA. Note that image owner can easily obtain his image collection from the cloud server. WCA then extracts watermark w_t from m_t through WatermarkExtra and identifies the possible illegal query user whose associated watermark is similar to the extracted watermark w_t.

In the following, we will present the details of our privacy-preserving and copy-deterrence CBIR scheme. For clarification, the description of the scheme is divided into two parts. At first, we present the privacy-preserving CBIR scheme. Next, the copy-deterrence functionality is added to the scheme.

B. Privacy-preserving CBIR protocol

In this subsection, we present the details of the algorithms that are involved in the privacy-preserving CBIR protocol, including KeyGen and IndexGen on the image owner side, TrapdoorGen on the image user side, and Search on the cloud server side.

- $\mathcal{K} \leftarrow \mathsf{KeyGen}(1^\kappa)$ is the key generation algorithm that takes as input the security parameter κ, and returns the set of secret keys $\mathcal{K} = \{\mathbf{S}, \mathbf{M}_1, \mathbf{M}_2, \{g_j\}_{j=1}^L, \{k_j\}_{j=1}^L, k_{img}\}$. Here, $\mathbf{S}$ is a vector of $l+1$ bits, $\mathbf{M}_1$ and $\mathbf{M}_2$ are two invertible matrices with the size $(l+1) \times (l+1)$, $\{g_j\}_{j=1}^L$ is the set of LSH functions, $\{k_j\}_{j=1}^L$ is the set of secret keys for the bucket encryption, and k_{img} is the secret key for the image encryption.
- $\mathcal{I} \leftarrow \mathsf{IndexGen}(\mathcal{K}, \mathcal{M})$ is the index generation algorithm that takes as input the secret key set $\mathcal{K}$ and the image collection $\mathcal{M}$, and returns the index $\mathcal{I}$. For clarification, we divide the process of index generation into two steps: the generation of unencrypted index and the index encryption.

  Step1: the generation of unencrypted index. At the beginning, a feature vector $f_i = (f_{i,1}, f_{i,2}, ..., f_{i,l})^T$ is extracted from each image $m_i \in \mathcal{M}$ using the feature extraction methods described in subsection III-D. Then, the similarity between the two images depends on the similarity between the two corresponding feature vectors. Intuitively, a one-to-one map index can be constructed for the search purpose. However, this will cause a linear search time. In order to get better search efficiency, a two-layer index is constructed. Specifically, the bottom layer is the one-to-one map index which is illustrated in Table I. The upper one consists of the pre-filter tables which are constructed on basis of the one-to-one map index. During the search process, most of the dissimilar images will be discarded quickly according to the pre-filter tables. Then, the similarity scores of remaining images to the query image are calculated and ranked according to the one-to-one map index.

  The construction of the one-to-one map index is quite simple. The pre-filter tables are constructed using the LSH which is introduced in subsection III-D. Specifically, the image owner randomly chooses λ LSH functions $h_1, h_2, ..., h_\lambda \in \mathcal{H}$ and applies $g(f_i) = (h_1(f_i), h_2(f_i), ..., h_\lambda(f_i))$ to the features in $\{f_i\}_{i=1}^n$ so as to build a pre-filter table. As introduced in subsection III-D, the function $g(\cdot)$ maps an l-dimensional vector into λ integers, which forms a λ-dimensional vector called bucket. The images with the same bucket value can be considered as a cluster of similar images. To provide more possible results, this process is repeated L times, generating L pre-filter tables. To sum up, the set of buckets can be denoted as $\{Bkt_{j,b}\}_{j\in[1,L],b\in[1,N_j]}$, where N_j refers to the total number of buckets in the j-th pre-filter table. In the proposed scheme, each image $m_i \in \mathcal{M}$ is mapped into L buckets. An example of the pre-filter table is illustrated in Table II.

  Setp2: the index encryption. The image features in plaintext may reveal information about the image content [22]–[27]. For example, a color histogram with large blue component would indicate the likely presence of the sky or ocean, and the shape descriptors may disclose the information about the likely object in the image. Therefore, the feature vectors in the one-to-one map index need to be encrypted. The key point is to maintain that the encrypted feature vectors can be still used to calculate and rank the similarity scores. Intuitively, the homomorphic encryption techniques can be employed here. However, the homomorphic encryption is generally time-consuming and leads an additional round of communication between the cloud server and the query user [36]. As an alternative, the secure kNN algorithm [37] is employed to protect the feature vectors. Specifically, for a feature vector $f_i = (f_{i,1}, f_{i,2}, ..., f_{i,l})^T$, we modify it into $\widehat{f}_i = (f_{i,1}, f_{i,2}, ..., f_{i,l}, \|f_i\|^2)^T$, where $\|f_i\|$ is the Euclidean norm of f_i. Next, we split $\widehat{f}_i$ into two random vectors $\{\widehat{f}_{ia}, \widehat{f}_{ib}\}$ according to $\mathbf{S}$ as: if $\mathbf{S}[j] = 0$, $\widehat{f}_{ia}[j]$ and $\widehat{f}_{ib}[j]$ are set equal to $\widehat{f}_i[j]$; if $\mathbf{S}[j] = 1$, $\widehat{f}_{ia}[j]$ and $\widehat{f}_{ib}[j]$ are set as two random values whose sum equals to $\widehat{f}_i[j]$. Finally, the encrypted feature vector is generated as $f_i' = \{\mathbf{M}_1^T \widehat{f}_{ia}, \mathbf{M}_2^T \widehat{f}_{ib}\}$.

  In addition, LSH does not necessarily have the one-way property. Thus, we cannot directly outsource the pre-filter tables to the cloud server as the bucket values may disclose the information about the features. To enhance

TABLE I
THE ONE-TO-ONE MAP INDEX

Image identity	Feature vector
$\mathsf{ID}(m_1)$	f_1
$\mathsf{ID}(m_2)$	f_2
...	...
$\mathsf{ID}(m_i)$	f_i
...	...
$\mathsf{ID}(m_n)$	f_n

TABLE II
THE j-TH PRE-FILTER TABLE

Bucket value	Image identities
$Bkt_{j,1}$	$\mathsf{ID}(m_3)$,$\mathsf{ID}(m_{27})$,$\mathsf{ID}(m_{51})$,$\mathsf{ID}(m_{115})$
$Bkt_{j,2}$	$\mathsf{ID}(m_{16})$,$\mathsf{ID}(m_{66})$,$\mathsf{ID}(m_{132})$,$\mathsf{ID}(m_{343})$
...	...
Bkt_{j,N_j}	$\mathsf{ID}(m_{24})$,$\mathsf{ID}(m_{43})$,$\mathsf{ID}(m_{432})$,$\mathsf{ID}(m_{456})$

the security, the bucket values are protected by a one-way hash function. Finally, the encrypted index $\mathcal{I}$, including the one-to-one map index and the pre-filter tables, is uploaded to the cloud server for secure CBIR.

- $TD \leftarrow \mathsf{TrapdoorGen}(\mathbf{S}, \mathbf{M}_1, \mathbf{M}_2, \{g_j\}_{j=1}^{L}, \{k_j\}_{j=1}^{L}, m_q)$. To retrieve similar images, a query user generates a trapdoor and sends it to the cloud server. The trapdoor should not reveal the information about the query image but can be used to search similar images on index $\mathcal{I}$. The procedure of TrapdoorGen is defined as follows:
  1. Calculate the feature vector f_q from the query image m_q.
  2. For each $j \in [1, L]$, compute the bucket value $Bkt_j = g_j(f_q)$, and then encrypt Bkt_j to be $\phi(Bkt_j, k_j)$ using the secret key k_j.
  3. Modify the query vector $f_q = (f_{q,1}, f_{q,2}, ..., f_{q,l})^T$ into $\widehat{f}_q = (-2f_{q,1}, -2f_{q,2}, ..., -2f_{q,l}, 1)^T$, and then split $\widehat{f}_q$ into two random vectors $\{\widehat{f}_{qa}, \widehat{f}_{qb}\}$ as: if $\mathbf{S}[j] = 0$, $\widehat{f}_{qa}[j]$ and $\widehat{f}_{qb}[j]$ are set as two random values whose sum equals to $\widehat{f}_q[j]$; if $\mathbf{S}[j] = 1$, $\widehat{f}_{qa}[j]$ and $\widehat{f}_{qb}[j]$ are set equal to $\widehat{f}_q[j]$. Note that the split operation here is a little different from that in IndexGen. Then, the query vector is encrypted as $f'_q = \{\gamma \mathbf{M}_1^{-1} \widehat{f}_{qa}, \gamma \mathbf{M}_2^{-1} \widehat{f}_{qb}\}$, where $\gamma \in R$ is a random positive value.
  4. Finally, the trapdoor is generated as $TD = \{\{\phi(Bkt_j, k_j)\}_{j=1}^{L}, f'_q\}$.
- $\mathcal{R} \leftarrow \mathsf{Search}(\mathcal{I}, \mathcal{C}, TD)$. Upon receiving a trapdoor TD from an image user, the cloud server executes Search to find the similar images.
  1. Firstly, the cloud server fetches the image IDs from L pre-filter tables according to the encrypted bucket values $\{\phi(Bkt_j, k_j)\}_{j=1}^{L} \in TD$. With the property of locality-sensitive hashing, these images are likely to be similar to the query image. This step filters out lots of dissimilar images very efficiently.
  2. Next, the distances of the images fetched above to the query image are calculated and ranked, which helps to reduce the communication burden by just sending the top-k most similar images to the query user. The distance between a database feature vector f_i and the query feature vector f_q is calculated as follows,

$$\begin{aligned} {f'_q}^T f'_i &= (\gamma M_1^{-1} \widehat{f}_{qa})^T M_1^T \widehat{f}_{ia} + (\gamma M_2^{-1} \widehat{f}_{qb})^T M_2^T \widehat{f}_{ib} \\ &= \gamma (\widehat{f}_{qa})^T \widehat{f}_{ia} + \gamma (\widehat{f}_{qb})^T \widehat{f}_{ib} \\ &= \gamma (\widehat{f}_q)^T \widehat{f}_i \\ &= \gamma (\|f_i\|^2 - 2 \sum_{j=1}^{l} f_{i,j} f_{q,j}) \\ &= \gamma (\|f_q - f_i\|^2 - \|f_q\|^2). \end{aligned} \tag{2}$$

  The distance $\|f_q - f_i\|^2$ is hidden by the secret scalar γ and the unknown $\|f_q\|^2$. And ${f'_q}^T f'_1 > {f'_q}^T f'_2$ implies $\|f_q - f_1\|^2 > \|f_q - f_2\|^2$, which means that the cloud server can directly find the closest feature vectors by simply sorting the set of vector products ${f'_q}^T f'_i$, without knowing the original feature vectors.
  3. Finally, the cloud server puts the top-k most similar encrypted images into the temporary set $\mathcal{R}$. After being watermarked, these images will be sent to the query user.

C. Copy-deterrence CBIR protocol

The watermarking technology is employed for the copy deterrence in the proposed scheme. At the beginning, a unique watermark associated with the query user is embedded into the encrypted images by the cloud server. Then, the encrypted and watermarked images are sent to the query user. After receiving the images, the query user can directly decrypt these images. The watermark is still preserved after the decryption. When an illegal copy of the image is found, the unlawful user who made the illegal distribution can be traced by examining the watermark in the image. This will deter the illegal distributions.

Several partial-encryption based commutative encryption and watermarking (CEW) methods have been proposed [38]–[41]. In these methods, the image data is divided into two parts. One part is encrypted to protect the image content, and the other is used to carry the watermark. The encryption and watermarking operations in these methods do not interfere with each other, which is suitable for our application scenario. However, the watermarked part has not been protected well and will leak information about the images.

Some watermarking methods are constructed based on homomorphic cryptosystems [28]–[32], [42], [43]. In these methods, the watermark can be embedded in the encrypted images and extracted in the decrypted ones, which fits our application. However, the homomorphic-cryptosystem based secure watermarking methods are not quite suitable for the application scenario of image retrieval. Firstly, the homomorphic encryption is quite time-consuming. Generally, a batch of images will be returned for each search request. It is unbearable to embed watermark in homomorphic-encryption domain for each search request. An alternative solution is to prepare a distinct watermarked image for each image user in advance. But this will greatly increase the storage burden to the cloud server.

Some researchers have designed the client-side watermarking method which transmits the same encrypted version to all the clients but lets the clients to decrypt the content with the client-specific keys. The watermark is implicitly embedded during the decryption [44], [45]. In [44], [45], the data is encrypted by addition operation which is suitable for embedding robust spread-spectrum watermarking. However, the choice of using the addition comes at the cost of losing the provable plain-text confidentiality [44]. It is usable in a general distribution scheme, but is not secure enough for a curious cloud server.

For security and efficiency, a CEW algorithm proposed by Zhang [46] is exploited in our scheme. Since the data owner can easily get the original images, we modify the embedding and extraction procedure of [46] to achieve a better watermark extraction accuracy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2016.2590944, IEEE Transactions on Information Forensics and Security

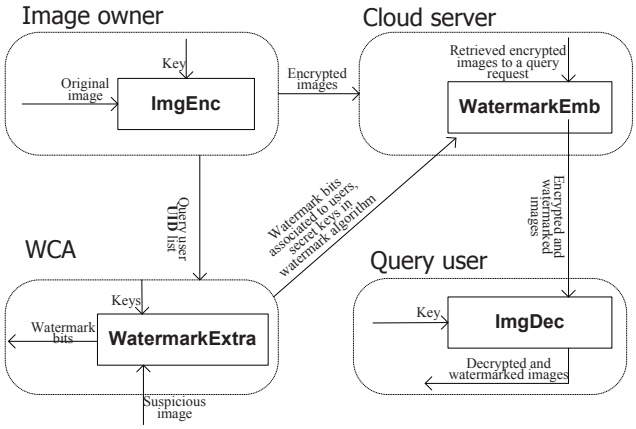


Fig. 3. Framework of the watermark-based protocol

1) Overview of Zhang's algorithm: In [46], each pixel in an grayscale image is composed of 8 binary bits. In the embedding process of Zhang's algorithm, the pixel bits of image are encrypted into random bits through the exclusive-or operation with a standard stream cipher. Then, the encrypted image is segmented into nonoverlapping blocks and a part of them are randomly chosen to carry watermark bits. Next, the pixels in each of chosen blocks are randomly divided into two sets S_0 and S_1 according to a secret key. If the watermark bit is 0, flip the 3 least significant bits (LSBs) of the pixels in S_0. Otherwise, flip the 3 LSBs of the pixels in S_1. In this way, an encrypted and watermarked image is generated. The encrypted and watermarked image can be decrypted by the same stream cipher. The decrypted image still contains the watermark in it.

The extraction of watermark bits is based on the fact that the fluctuation of an original image block is generally lower than that of a flipped one. Firstly, the blocks carrying the watermark bits are found according to the secret key. Secondly, the pixels of each block are divided into two sets according to the secret key. Finally, 3 LSBs of pixels in sets S_0 and S_1 are flipped separately. By observing the change of fluctuation, we can guess the embedded bit is 1 or 0. Note that, it is not guaranteed that each watermark bit can be correctly extracted. For more information about Zhang's work, please refer to [46].

2) The proposed watermark-based protocol: In our watermark-based protocol, we improve Zhang's watermarking algorithm in [46] to increase the robustness. In the extraction process, we extract the watermark bits by comparing the watermarked image and its corresponding original version. A sketch of our watermark-based copy-deterrence protocol is illustrated in Fig. 3.

With an image collection $\mathcal{M}$, the image owner runs ImgEnc to generate an encrypted image set $\mathcal{C}$ with a standard stream cipher, and then outsources $\mathcal{C}$ to the cloud server. Upon receiving a query request, the cloud server retrieves the temporary search result $\mathcal{R}$ according to *TD*, and gets the watermark w related to UID. Then, the cloud server embeds the watermark w into the images in $\mathcal{R}$ through WatermarkEmb, generating the watermarked-encrypted image collection $\mathcal{R}'$ which will be sent to the query user as the final result set. After receiving $\mathcal{R}'$, the image user decrypts the images in $\mathcal{R}'$ to get the image set $\mathcal{M}_q$ including the watermarked and decrypted images. If a unauthorized copy of image m_t is found, the image owner submits both the unauthorized copy m_t and the corresponding original version m_o to WCA which then exacts watermark w_t by WatermarkExtra. Finally, the extracted watermark w_t is used to identify the illegal user whose watermark is similar to w_t. The proposed watermarking algorithm is expected to achieve a better extraction accuracy than that in [46]. The detailed algorithms are described in Fig. 4 and 5. Please note that, these algorithms are designed for grayscale images. The color image with 3 color channels can be simply divided into 3 grayscale ones before using the algorithms here.

$\mathcal{C} \leftarrow \mathsf{ImgEnc}(\mathcal{K}, \mathcal{M})$

1. For each grayscale image $m \in \mathcal{M}$,
   1) Generate the secret key with a one-way pseudorandom number generator as $StreamKey \leftarrow PRNG(k_{img}, \mathsf{ID}(m))$. The size of $StreamKey$ is the same as the image m, and the numbers in $StreamKey$ are integers in the range of [0,255] as the pixels in grayscale images.
   2) For each pixel in m,
      a) Denote the pixel value at position (i,j) as $p_{i,j}$, and the bits of $p_{i,j}$ are computed as $p_{i,j,k} = \lfloor p_{i,j}/2^k \rfloor \bmod 2, k = 0, 1, ..., 7$. Similarly, denote the value of $StreamKey$ at position (i,j) as $b_{i,j}$, and the bits of $b_{i,j}$ are computed as $b_{i,j,k} = \lfloor b_{i,j}/2^k \rfloor \bmod 2, k = 0, 1, ..., 7$.
      b) Encrypt the pixel value as $e_{i,j,k} = p_{i,j,k} \otimes b_{i,j,k}$, for $k = 0, 1, ..., 7$, where $\otimes$ is the exclusive-or operator.
2. Output the encrypted image collection $\mathcal{C}$.

$\mathcal{M}_q \leftarrow \mathsf{ImgDec}(\mathcal{K}, \mathcal{R}')$

1. For each image $m' \in \mathcal{R}'$,
   1) Obtain the secret key as $StreamKey \leftarrow PRNG(k_{img}, \mathsf{ID}(m'))$.
   2) For each pixel in m',
      a) Denote the pixel value of m' at position (i,j) as $e'_{i,j}$, and the bits of $e'_{i,j}$ are computed as $e'_{i,j,k} = \lfloor e'_{i,j}/2^k \rfloor \bmod 2, k = 0, 1, ..., 7$. Similarly, denote the value of $StreamKey$ at position (i,j) as $b_{i,j}$, and the bits of $b_{i,j}$ are computed as $b_{i,j,k} = \lfloor b_{i,j}/2^k \rfloor \bmod 2, k = 0, 1, ..., 7$.
      b) Decrypt the pixel value as $p'_{i,j,k} = e'_{i,j,k} \otimes b_{i,j,k}$, for $k = 0, 1, ..., 7$.
2. Output decrypted image collection $\mathcal{M}_q$.

Fig. 4. The image encryption and decryption algorithms in the watermark-based protocol

V. Security analysis

In the proposed scheme, three types of entities, i.e., the data owner, image user, and cloud server, could trigger security problems. Here, two kinds of security issues are mainly considered.

A. Data privacy problem

Similar to the previous SE scheme [22]–[27], the cloud server is considered to be "honest-but-curious", which means that the cloud server will correctly follow the protocol

$\mathcal{R}' \leftarrow \mathsf{WatermarkEmb}(\mathcal{R}, w, k_{emb1}, k_{emb2}, k_{emb3})$

1. For each encrypted image $c \in \mathcal{R}$
   1) Divide c into $s \times s$ sized nonoverlapping blocks. The watermark is a sequence of binary bits denoted as $w = w_1, w_2, ..., w_{N_w}$. A set of blocks $\{\boldsymbol{BK}_i\}_{i=1}^{N_w}$ are chosen by a pseudorandom function with the secret key k_{emb1}. Each block will carry one bit of the watermark.
   2) For each watermark bit $w_i, i \in [1, ..., N_w]$,
      a) The pixels in block $\boldsymbol{BK}_i$ are divided into two sets S_0 and S_1 according to a pseudorandom function with the secret key k_{emb2};
      b) If $w_i = 0$, flip the bits of pixels in S_0. Otherwise, flip the pixel bits in S_1. In order to preserve the image quality, we make less flipping on higher bit-planes. We denote the ratios of flipped bits on 8 bit-planes as $\epsilon = [\epsilon_1, \epsilon_2, ..., \epsilon_8]$. That is to say, for the i-th bit-plane, there are $N_w \times s^2 \times \epsilon_i/2$ bits will be flipped randomly. The flipped positions are determined by k_{emb3}.
2. Output the encrypted and watermarked image set $\mathcal{R}'$.

$w_t \leftarrow \mathsf{WatermarkExtra}(m_t, m_o, k_{emb1}, k_{emb2}, k_{emb3})$

1. Divide m_t into nonoverlapping blocks with the size $s \times s$.
2. Locate the set of blocks $\{\boldsymbol{BK}_i\}_{i=1}^{N_w}$ that carries the watermark bits $w = w_1, w_2, ..., w_{N_w}$ according to the secret key k_{emb1}.
3. For each $i \in [1, N_w]$,
   1) Divide the pixels in $\boldsymbol{BK}_i$ into two sets S_0 and S_1 according to the secret key k_{emb2};
   2) Flip the pixels in S_0 and S_1 respectively according to $[\epsilon_i]_{i=1}^8$ and k_{img3} to get two blocks BK_i^0 and BK_i^1. Construct the corresponding block BK_i from the original image with the secret key k_{emb1}. Calculate $\delta_0 = \sum_{p_j \in BK_i, p_j^0 \in BK_i^0} (p_j^0 - p_j)^2$ and $\delta_1 = \sum_{p_j \in BK_i, p_j^1 \in BK_i^1} (p_j^1 - p_j)^2$. If $\delta_0 < \delta_1$, the watermark bit is extracted as '0'. Else, the watermark bit is extracted as '1'.
4. Output the extracted watermark w_t.

Fig. 5. The watermark embedding and extraction algorithm

specification but keep curious to analyze information about the images. Thus, the privacies of the image content, image features, and trapdoors need to be properly protected. The security properties of the scheme depend on several classic techniques. Thus, we do not formally define and prove the security properties of the scheme.

1) **The privacy of image content.** The images stored on the cloud server are encrypted with a standard stream cipher. The keystreams are generated by a one-way pseudorandom number generator which takes as input a secret key and the unique image identity. The keystreams for different images are different with each other. In this case, the standard stream cipher is secure against Chosen-plaintext Attack (CPA) model [47]. Thus, the privacy of image content in our scheme is well protected.
2) **The privacy of image features.** The image features may reveal the information about image content. In our scheme, the feature vectors are encrypted by the secure kNN algorithm which is proved to be secure against the Chipertext-only Attack (COA) model [37]. In addition, the bucket values mapped from the feature vectors are further protected by a one-way hash function. Thus, the information of feature vectors will not be leaked from these bucket values.
3) **The privacy of trapdoors.** Similar to the feature vectors in index, the feature vectors in trapdoors are encrypted by the secure kNN algorithm. And the bucket values in trapdoors are also further protected to prevent the information leakage. Thus, the privacy of trapdoors is also well protected.
4) **The leakage of similarity information.** In our scheme, the similarity information among images are leaked. For instance, if the images m_i and m_j are returned as the search results to the same query, it is easy to deduce that the images m_i and m_j are similar to each other. In addition, we use the pre-filter tables to group the similar images to improve the search efficiency. Thus, the cloud server knows those images in the same bucket are similar to each other. This type of information leakage is a compromise for the efficiency.

B. Copyright problem

In the proposed scheme, we recall the assumption that the query users will correctly follow the protocol specification, but may distribute the retrieved images to someone unauthorized for benefits. The watermarking techniques are adopted to deter the illegal distribution. On the other side, the data owner may try to frame an innocent user by embedding the user's watermark into an original image. This illegal operation should be also prohibited in the proposed scheme.

1) Framing problem: In order to frame an image user, the image owner needs to know the user's unique watermark, the embedding algorithm, and the embedding secret keys. In our scheme, the watermarks are generated and sent to the cloud server by the trusted WCA. Only the cloud server and WCA know the watermarks and the embedding keys. Under the assumption that there is no collusion among the image owner, cloud server, and WCA, the watermarks and embedding secret keys are kept secret from the image owner. In this case, the image owner cannot frame the image users.

2) Trace to the illegal distributor: If the image owner finds his image is exposed to someone unauthorized, the owner can submit the illegal copy and the corresponding original image to WCA. WCA takes the responsibility to extract the watermark from the doubtful image. Then, the extracted watermark is used to identify the illegal user. However, after obtaining the watermarked images, the user may change it by some regular image processing operations, e.g., JPEG compression. In this case, the watermarks could not be extracted with 100% accuracy. Then, the trace under the extraction errors needs to be discussed.

Definition 1: For any two watermarks w_a and w_b with the length N_w, the similarity between them is defined as $\frac{N_s}{N_w}$, where N_s is the number of same bits of two watermarks at the same positions.

Take the watermarks '0100110110' and '1111101011' as an example. We can see that the lengths of watermarks are ten, and the bits of two watermarks at the second, fifth and ninth

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2016.2590944, IEEE Transactions on Information Forensics and Security

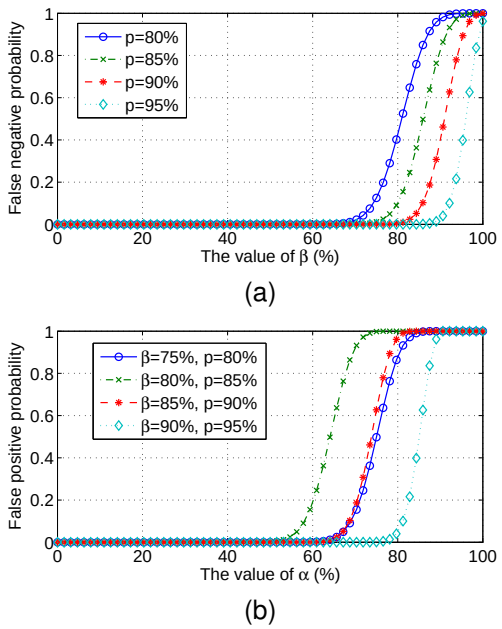


Fig. 6. The probability of (a) false negative and (b) false positive under different p, α, β.

positions are the same, respectively. So, the similarity between the two watermarks is calculated to be 0.3. The proposed scheme is designed so that the similarity between any two watermarks w_a and w_b is not larger than α.

Definition 2: Denote the w_e as the watermark extracted from an illegal distributed image and w_u as the watermark associated with a query user. If the similarity between w_e and w_u is larger than β ($\beta > \alpha$), the query user will be judged to have distributed the image.

Definition 3: The false negative is the case that an illegal user is missed by the watermark-based protocol. The false positive is the case that an innocent user is wrongly judged to be an illegal distributor.

The probability of false negative. We assume that the extractions of watermark bits are independent with each other, and the probability of correct extraction (called extraction accuracy) of each bit is p. According to **Definition 2**, when the number of rightly extracted watermark bits is larger than $\beta \times N_w$, there will be no false negative. Then, the false negative probability of the watermark-based protocol is

$$P_\beta = 1 - \sum_{i=\beta N_w}^{N_w} C_{N_w}^i p^i (1-p)^{N_w - i}. \quad (3)$$

Fig. 6(a) shows the distribution of P_β when $N_w = 64$, which indicates that the smaller β and larger p introduces less false negative.

The probability of false positive. Since the watermark bits may be wrongly extracted, a wrong user could be traced by the watermark-based protocol. The proposed scheme is designed so that the similarity between any two watermarks w_a and w_b is not larger than α. Thus, the case of false positive will not occur until the number of wrongly extracted bits is larger than $(\beta - \alpha) \times N_w$. Then, the upper bound of false positive possibility of our scheme is

$$P_\alpha = \sum_{i=(\beta-\alpha)\times N_w}^{N_w} C_{N_w}^i (1-p)^i p^{N_w - i}. \quad (4)$$

Fig. 6(b) shows the distribution of P_α when $N_w = 64$, which indicates that a smaller α introduces less false positive.

We can set the parameters α and β flexibly according to real world applications. Table III lists the settings of α and β in different cases. With these settings, both the false negative and false positive are smaller than 10^{-6}.

TABLE III
VALUES OF α AND β WITH THE PROBABILITY OF FALSE NEGATIVE AND FALSE POSITIVE ARE SMALLER THAN 10^{-6}

Number of watermark bits N_w	Watermark extraction accuracy p	β	α
1024	0.95	0.916016	0.831055
	0.90	0.853516	0.706055
	0.85	0.795898	0.590820
	0.80	0.739258	0.477539
512	0.95	0.900391	0.798828
	0.90	0.833984	0.666016
	0.85	0.771484	0.541016
	0.80	0.714844	0.427734
256	0.95	0.878906	0.753906
	0.90	0.804688	0.605469
	0.85	0.738281	0.472656
	0.80	0.675781	0.347656
128	0.95	0.84375	0.679688
	0.90	0.765625	0.523438
	0.85	0.687500	0.367188
	0.80	0.625000	0.242188
64	0.95	0.796875	0.578125
	0.90	0.703125	0.390625
	0.85	0.625000	0.234375
	0.80	0.546875	0.078125

The number of available watermarks. A small α means a small similarity between different watermarks in the scheme, which leads to small probabilities of false negative and false positive. However, a small α also cause a very limited number of available watermarks. Furthermore, it is difficult to estimate the maximum number of available watermarks with a specific α. So far, we only find a lower bounder of the number of available watermarks. That is $2^{\lfloor \frac{1}{1-\alpha} \rfloor}$. Please note that, this is a common problem of all watermark-based schemes.

C. Collusion problems

This paper assumes that the WCA is fully trusted, the cloud server is honest-but-curious, the image users follows the protocol specification to search images but may distribute the retrieved images to the unauthorized ones, and the image owner may try to frame an image user. We assume that the image owner, cloud server, data user, and WCA will not collude with each other. These are basic assumptions in the buyer-seller protocols [28], [29], [48]–[50] and the SSE schemes [2]–[4], [6]–[9], [11], [13], [14], [22]–[27], [33]. However, our scheme is a combination of the SSE scheme and watermark-based protocol. Our threat model is slightly different from the previous SSE schemes since we consider

the not-fully trusted query users. In this situation, it is better to take the collusion problems into consideration. In following, we discuss the collusion problems and the possible ways to tackle them. We show that it will increase the computation, communication, and storage burden a lot to deal with these collusion problems.

The collusion between the image owner and cloud server. In this case, the image owner will be able to frame an image user. Here, we discuss two possible ways to deal with this problem:

1) Jun Zhang et al. [42] have proposed a secure image retrieval scheme based on the watermark-based protocol to protect the copyright of service user's query image against the image retrieval service provider. Please note that the application scenario in [42] is different from that in an SE scheme. Compared to a buyer-seller protocol, the service provider in [42] can be referred as the buyer, and the service user can be referred as the seller. Upon receiving a query request, the service provider generates a watermark and encrypts it with his public key. Next, the encrypted watermark, the public key, and the corresponding signatures are sent to the service user. The service user also generates a watermark and encrypts it with the service provider's public key. Then, both the two watermarks are embedded into the query image in the encryption domain. The service user sends the watermarked and encrypted query image to the service provider and stores the encrypted watermark, the public key, and the corresponding signatures which are needed for the arbitration. The service provider then decrypts the query image to search similar images. The decrypted query image still contain the watermark in it. In [42], the service user knows nothing about the service provider's watermark, and thus can not frame the service provider. However, the adoption of the techniques in [42] will greatly increase the burden of a privacy-preserving CBIR scheme. Firstly, it will increase the computation burden and cause storage expansions to encrypt images with a homomorphic encryption. Secondly, the cloud server needs to generate a distinct encrypted image collection for each query user with the corresponding public key. This will greatly increase the storage burden of the cloud server.
2) Another alternative is to have WCA to embed the watermark under the assumption that WCA is fully trustworthy. Under this situation, the cloud server needs to send the retrieved images to WCA in each query. Then, WCA takes the responsibilities to embed the watermark and send the watermarked images to the query user. This will increase the communication overhead of the scheme, and could be too complex for a realistic WCA. In addition, WCA may be also interested in the image privacy, and could not be fully trusted any more if he takes the responsibility to transmit the encrypted images.

The collusion between the image user and cloud server. If an image user leaks the secret key to the cloud server, the image collection will be totally exposed to the cloud server. Thus, all the existing SSE schemes are constructed on the assumption that there is no collusion between the image user and cloud server [2]–[4], [6]–[9], [11], [13], [14], [22]–[27], [33]. In order to deal with this collusion problem in our scheme, one could divide the secret keys into two parts and respectively send the two parts to the image user and WCA. Under this situation, to carry out a query, the query user firstly encrypts the feature vector with its secret key, and sends the semi-encrypted feature vector to WCA. Then, WCA further encrypts the feature vector and sends final-encrypted feature vector to the cloud server. After searching on the index, the cloud server firstly sends the retrieved images to WCA. Then, WCA decrypts the images with its secret key (maybe also embed the watermark at this time) and sends the semi-decrypted images to the query user who will further decrypt to get the fully-decrypted images. This will increase the communication overhead of the scheme and cause a complex WCA. Moreover, in this scenario, the cloud server, image user, and WCA might collude with each other to obtain the full secret keys. This is a quite similar collusion problem as with our current scheme.

TABLE IV
PARAMETERS IN THE EXPERIMENTS

Visual descriptors	Parameters			
	l	L	λ	r
CSD	128	12	2	14
SCD	128	5	2	6
CLD	120	4	2	4
EHD	80	2	2	4

VI. PERFORMANCE EVALUATIONS

In this section, we present the performance evaluations of the proposed scheme on Corel image dataset which is a benchmark dataset for the image retrieval performance test [51]. This image database includes 100 categories of images and each category contains 100 similar images. The entire scheme is implemented using C++ language on a Windows 7 with Intel Core(TM) Duo Processor 2.80 GHz. The performance of the scheme depends on several parameters, including the parameter in hash function r, the number of connected LSH functions λ, the number of pre-filter tables L, and the dimensionality of visual descriptor l. In this paper, these parameters are set empirically except l which is fixed in a specific feature extraction algorithm. The parameters in our experiments are summarised in Table IV. Please note that the parameters used here are not claimed to be the optimal ones.

A. Retrieval precision

In our experiments, the "precision" for a query is defined as that in [52]: $P_k = k'/k$, where k' is the number of real similar images in the k retrieved images. According to the Equation 2, the encryption of the feature vectors will not influence the retrieval precision. However, the pre-filter tables which are utilized to improve the search efficiency will affect the retrieval precision.

Four MPEG-7 descriptors are adopted to test the retrieval precisions. The precisions of our schemes with and without the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2016.2590944, IEEE Transactions on Information Forensics and Security

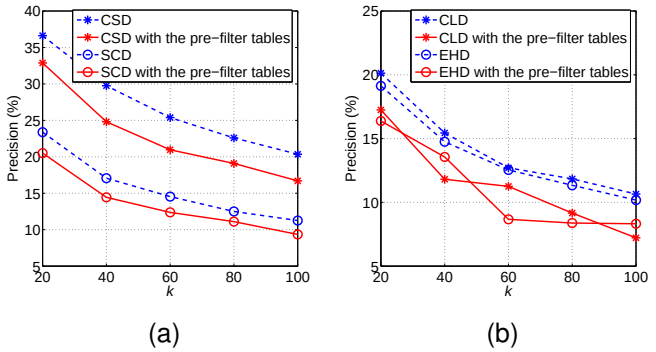


Fig. 7. Average retrieval precision when top-k results are retrieved, (a) CSD and SCD descriptors, and (b) CLD and EHD descriptors.

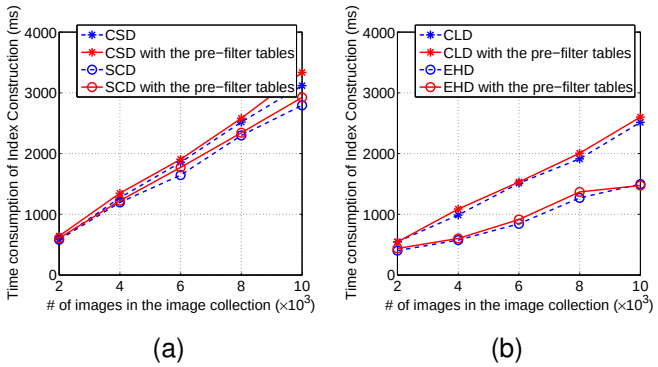


Fig. 8. Time consumption of the index construction, (a) SCD and CSD descriptors, and (b) CLD and EHD descriptors.

pre-filter tables are compared. We choose 20 image categories to test the retrieval precisions. Two samples from each category are randomly selected as the query images. Accordingly, the retrieval precisions are averaged from 40 queries for each visual descriptor.

The average precisions of the four descriptors are presented in Fig. 7. The precisions are mainly dependent on the performances of the visual descriptors. The usage of the pre-filter tables decreases the retrieval precision. The average decline rates are 15.04%, 19.90%, 13.93, and 18.59% for CSD, CLD, EHD, and SCD, respectively. These losses of precision are traded off for the search efficiency.

B. Efficiency

The time consumptions of the index construction, trapdoor generation, search operation, and watermark embedding process are tested in this subsection. In addition, the storage consumption of the index is also presented.

1) Time consumption of the index construction: Before the index construction, we have completed the feature extraction. Thus, the time consumption of index construction in our experiments mainly includes two parts: 1) the consumption for building L hash tables, and 2) the consumption for encrypting the feature vectors with a splitting operation and two multiplicative operations with the $(l+1)\times(l+1)$ matrices. In order to construct a pre-filter table, it takes $O(n\lambda l)$ time to generate the bucket values, where n denotes the total number of images, l denotes the dimensionality of feature vector, and λ denotes the number of jointed hash functions. The time complexity of the splitting operation is $O(nl)$, and the time complexity of the matrix multiplication is $O(nl^2)$. In total, the time complexity for the index construction is $O(Ln\lambda l + nl + nl^2)$. Since L, λ and l are the fixed constants in our scheme, the time consumption of the index construction is almost linear to the size of image collection, i.e., $O(n)$. The time consumption of the index construction is illustrated in Fig. 8, which shows that the construction of the pre-filter tables costs very little time.

2) Time consumption of the trapdoor generation: Similar to the index generation, the trapdoor generation incurs the calculations of bucket values, a splitting operation, and two matrix multiplications. The time complexity is $O(L\lambda l+l+l^2)$. The time cost of the trapdoor generation is mainly dependent on the dimensionality of the visual descriptor, as illustrated in Fig. 9.

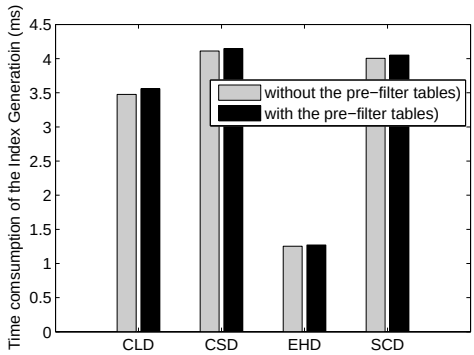


Fig. 9. Time consumption of the trapdoor generation, the data are averaged from 40 times of trapdoor generation

3) Time consumption of the search operation: During the search process, the cloud server firstly retrieves n' similar images from the pre-filter tables according to the bucket values submitted by the query user. Next, a linear search is executed over these n' images to retrieve the top-k most similar results. Thus, the search complexity of the scheme with the pre-filter tables is $O(n')$. Generally, n' is expected to be far less than the database size n. As is illustrated in Fig. 10, the scheme with the pre-filter tables achieves better efficiency than that without the pre-filter tables. The average decline rates of the time consumption are 52.32%, 67.12%, 58.86% and 53.14% for CSD, SCD, EHD and CLD, respectively. Please note that the efficiency of the scheme can be further improved with the smaller L, smaller r, and larger λ, which can be flexibly adjusted according to the real-world applications.

4) Time consumption of the watermark embedding: The proposed watermark algorithm embeds watermark bits by flipping image pixels according to each bit flipping proportion$[\epsilon_1, \epsilon_2, ..., \epsilon_8]$. The number of the flipped bits equals to $\sum_{i=1}^{8} \varepsilon_i \times N_w \times s^2/2$.The time complexity of flipping is $O(N_w \times s^2)$. However, in the execution of an algorithm, additional time is usually needed. We record the times that consumed by the flipping of bits and the whole embedding process, respectively, which are shown in Table V and VI. The results are averaged from the embedding process of 1000

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2016.2590944, IEEE Transactions on Information Forensics and Security

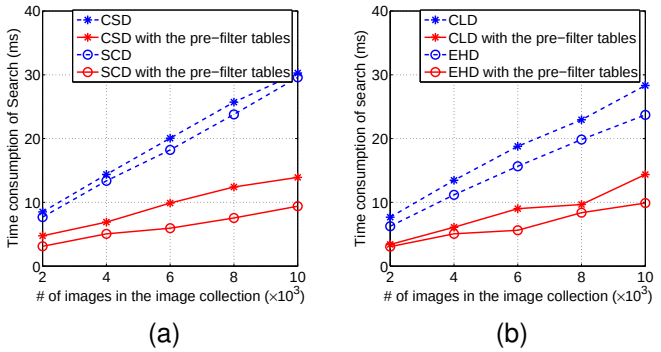


Fig. 10. Average search time for different size of image collection, (a) SCD and CSD descriptors, and (b) CLD and EHD descriptors.

TABLE V
TIME CONSUMED BY BIT FLIPPING (*ms*)

Number of watermark bits	Size of block s		
	16	24	32
16	0.122	0.268	0.475
32	0.244	0.543	0.961
64	0.492	1.092	1.823

images with the size of 384×256 or 256×384. The bit flipping ratios are set to [0, 0, 0, 0.1, 0.2, 0.4, 0.8, 0.9]. Table V and VI show that the time consumed by the flipping is only a small portion of the time consumed by the whole embedding process. Most of the time are consumed by the operations such as the image loading, image copy, and memory release.

5) Storage consumption of the index: The index in the proposed scheme consists of a one-to-one map index and L hash tables. In Table VII, we present the storage consumption of the indexes with and without the pre-filter tables. The results are calculated from 10,000 images. Table VII shows that the pre-filter tables consume a little storage compared with the one-to-one map index.

C. Watermark extraction accuracy

After receiving the watermarked image, the query user may change it by regular image processing operations, e.g., JPEG compression. In addition, with the knowledge of watermarking algorithm, the user may try to remove the watermark bits by flipping the bits of image pixel. In this case, the watermark bits could not be extracted with the 100% accuracy. In this paper, the watermark extraction accuracy is defined as $p = N_e/N_w$, where N_e is the number of correctly extracted watermark bits and N_w is the total number of the watermark bits. Please note that, the embedding parameters used here are not claimed to be the optimal ones. One can adjust the parameters flexibly to get a good tradeoff between the robustness and image equality according to the application scenarios.

To test the robustness of the proposed watermarking algorithm, we conduct bit-flipping and JPEG-compression to attack the watermarked images, and then try to extract watermark bits from these attacked images. In the embedding process, we set the number of watermark bits $N_w = 64$, the bit-flipping ratios $\epsilon = [0, 0, 0, 0.1, 0.2, 0.4, 0.8, 0.9]$ for 8 bit-plane, and the block

TABLE VI
TIME CONSUMED BY THE WHOLE WATERMARK EMBEDDING PROCESS (*ms*)

Number of watermark bits	Size of block s		
	16	24	32
16	4.933	5.121	5.154
32	5.091	5.257	5.608
64	5.471	5.836	6.657

TABLE VII
STORAGE CONSUMPTION OF INDEXES FROM 10,000 IMAGES

Index	Storage consumption (KB)			
	CSD	SCD	CLD	EHD
without the pre-filter tables	35382	35382	33195	22258
with the pre-filter tables	36389	36370	34181	23244

size s =16, 24 and 32. Fig. 11 (b) shows that the watermark leads no significant distortion to the image.

Generally, the bit-flipping ratio ϵ and flipping position can be kept as secret parameters to the attacker. We assume that the attacker only make the flipping on 5 lower bit-planes so as to preserve a satisfied image quality. The flipping is conducted randomly on these bit-planes. Fig. 12 shows that the averaged extraction accuracies are larger than 95% when 30% bits on 5 lower bit-planes are flipped. However, the distortion of image cannot be negligible in this case, as shown in Fig. 11(c). It means that the attacker cannot remove the watermark with the negligible distortion. In addition, a larger block size s helps to enhance the robustness of our watermarking algorithm.

Fig. 13 presents the average extraction accuracies under JPEG-compression attack. The results are averaged from 1000 images. It is shown that higher extraction accuracy is achieved with larger block size s, and the JPEG-compression with higher quality factor causes less extraction errors. The only exception is the case of Zhang's algorithm with the quality factor of 70%. It is because most of the original images are JPEG images with the quality factor 70%. The recompression with the same factor results in less influence on the images.

VII. CONCLUSIONS

In this paper, we presented a privacy-preserving and copy-deterrence content-based image retrieval scheme in a cloud computing scenario. The secure kNN algorithm is applied to encrypt the visual features. The similarity scores can be directly calculated with the encrypted features by the cloud server, which enables the cloud server to rank the images without the additional communication burden. The locality-sensitive hashing is utilized to improve the search efficiency. For the first time, we consider the dishonest users in an SE schemes and propose a watermark-based protocol to deter the illegal distribution of images. Overall, the image features are secure against Ciphertext-only Attack model, the image contents are secure against Chosen-plaintext Attack model, and the search efficiency is improved from $O(n)$ to $O(n')$.

As future works, there still are some aspects could be improved. Firstly, the proposed watermarking method cannot be regarded as a very robust one. Secondly, a small parameter α will cause a limited number of available watermarks. In the future, we will make more efforts to design watermarking algorithm with better robustness and embedding capacity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2016.2590944, IEEE Transactions on Information Forensics and Security

Fig. 11. The distortion of image under bit-flipping attack. The peak signal to noise ratio ($PSNR$) and structural similarity image measurement ($SSIM$) of the watermarked and attacked images are calculated to quantifiably show the image quality. (a) original image, (b) watermarked image, $PSNR = 34.2603dB$, $SSIM$=0.8531, (c) watermarked image after bit-flipping attack with 30% bits flipped, $PSNR = 27.6602dB$, $SSIM$=0.6085, and (d) watermarked image after bit-flipping attack with 40% bits flipped, $PSNR = 26.6603dB$, $SSIM$=0.5670.

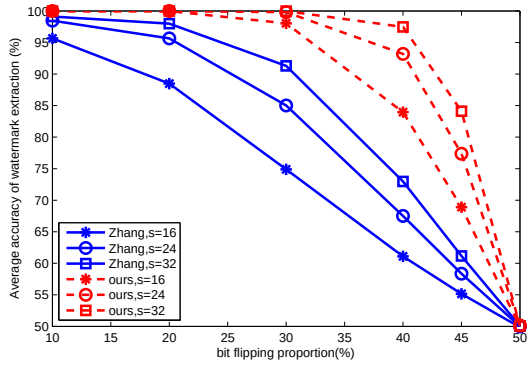


Fig. 12. The average accuracy of watermark extraction after bit-flipping attack. The results are averaged from 1000 images

ACKNOWLEDGMENT

This work is supported by the NSFC (61173141, U1536206, 61232016, U1405254, 61373133, 61502242, 61572258), BK20150925, Fund of Jiangsu Engineering Center of Network Monitoring (KJR1402), Fund of MOE Internet Innovation Platform (KJRP1403), CICAEET, PAPD fund, and the US National Science Foundation(CNS-1262277). Zhihua Xia is supported by Jiangsu Government Scholarship for Overseas Studies (JS-2014-332). The authors would like to thank Prof. Yun Q. Shi for the valuable suggestions for the revision of paper.

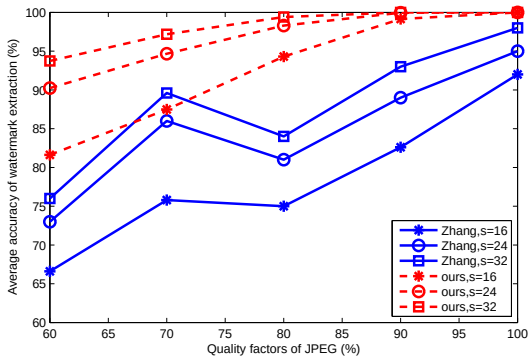


Fig. 13. The average accuracy of watermark extraction after JPEG-compression attack. The results are averaged from 1000 images

REFERENCES

[1] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-Eurocrypt*. Springer, 2004, pp. 506–522.

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. of IEEE Symposium on Security and Privacy*. IEEE, 2000, pp. 44–55.

[3] E.-J. Goh *et al.*, "Secure indexes." *IACR Cryptology ePrint Archive*, vol. 2003, p. 216, 2003.

[4] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proc. of 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 79–88.

[5] J. Shen, H. Tan, J. Wang, J. Wang, and S. Lee, "A novel routing protocol providing good transmission reliability in underwater sensor networks," *Journal of Internet Technology*, vol. 16, no. 1, pp. 171–178, 2015.

[6] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in *Proc. of 28th International Conference on Data Engineering*. IEEE, 2012, pp. 1156–1167.

[7] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *Proc. of INFOCOM*. IEEE, 2012, pp. 451–459.

[8] Z. Xia, Y. Zhu, X. Sun, and L. Chen, "Secure semantic expansion based search over encrypted cloud data supporting similarity ranking," *Journal of Cloud Computing*, vol. 3, no. 1, pp. 1–11, 2014.

[9] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[10] Z. Fu, X. Sun, Q. Liu, L. ZHOU, and J. SHU, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, vol. 98, no. 1, pp. 190–200, 2015.

[11] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, p. 1, 2015.

[12] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel & Distributed Systems*, vol. PP, no. Online, pp. 1–1, 2015.

[13] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," in *Financial Cryptography and Data Security*. Springer, 2013, pp. 258–274.

[14] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic searchable encryption in very large databases: Data structures and implementation," in *Proc. of Network and Distributed System Security Symposium*, vol. 14, 2014.

[15] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *Journal of Internet Technology*, vol. 16, no. 2, pp. 317–323, 2015.

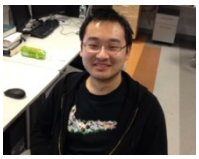
[16] J. Shashank, P. Kowshik, K. Srinathan, and C. Jawahar, "Private content based image retrieval," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[17] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Secure and robust sift," in *Proc. of 17th ACM international conference on Multimedia*. ACM, 2009, pp. 637–640.
[18] ——, "Image feature extraction in encrypted domain with privacy-preserving sift," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4593–4607, 2012.
[19] P. Zheng and J. Huang, "An efficient image homomorphic encryption scheme with small ciphertext expansion," in *Proc. of 21st ACM international conference on Multimedia*. ACM, 2013, pp. 803–812.
[20] Z. Qin, J. Yan, K. Ren, C. W. Chen, and C. Wang, "Towards efficient privacy-preserving image feature extraction in cloud computing," in *ACM International Conference on Multimedia*. ACM, 2014, pp. 497–506.
[21] Z. Xia, X. Wang, X. Sun, and B. Wang, "Steganalysis of least significant bit matching using multi-order differences," *Security and Communication Networks*, vol. 7, no. 8, pp. 1283–1291, 2014.
[22] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," in *Proc. of IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2009, pp. 725 418–725 418.
[23] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 1533–1536.
[24] B. Cheng, L. Zhuo, Y. Bai, Y. Peng, and J. Zhang, "Secure index construction for privacy-preserving large-scale image retrieval," in *Proc. of the Fourth International Conference on Big Data and Cloud Computing*. IEEE, 2014, pp. 116–120.
[25] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards privacy-preserving content-based image retrieval in cloud computing," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2015.
[26] B. Ferreira, J. Rodrigues, J. Leitão, and H. Domingos, "Privacy-preserving content-based image retrieval in the cloud," *arXiv preprint arXiv:1411.4862*, 2014.
[27] H. Cheng, X. Zhang, J. Yu, and F. Li, "Markov process based retrieval for encrypted jpeg images," in *Proc. of 10th International Conference on Availability, Reliability and Security*. IEEE, 2015, pp. 417–421.
[28] N. Memon and P. W. Wong, "A buyer-seller watermarking protocol," *IEEE Transactions on Image Processing*, vol. 10, no. 4, pp. 643–649, 2001.
[29] A. Piva, F. Bartolini, and M. Barni, "Managing copyright in open networks," *IEEE Internet Computing*, vol. 6, no. 3, pp. 18–26, 2002.
[30] S. Katzenbeisser, A. Lemma, M. U. Celik, M. Van Der Veen, and M. Maas, "A buyer–seller watermarking protocol based on secure embedding," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 4, pp. 783–786, 2008.
[31] M. Deng, T. Bianchi, A. Piva, and B. Preneel, "An efficient buyer-seller watermarking protocol based on composite signal representation," in *Proceedings of the 11th ACM workshop on Multimedia and security*. ACM, 2009, pp. 9–18.
[32] A. Rial, M. Deng, T. Bianchi, A. Piva, and B. Preneel, "A provably secure anonymous buyer–seller watermarking protocol," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 4, pp. 920–931, 2010.
[33] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in *Proc. of 8th ACM SIGSAC symposium on Information, computer and communications security*. ACM, 2013, pp. 71–82.
[34] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, 2001.
[35] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. of the twentieth annual symposium on Computational geometry*. ACM, 2004, pp. 253–262.
[36] S. Rane and P. T. Boufounos, "Privacy-preserving nearest neighbor methods: comparing signals without revealing them," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 18–28, 2013.
[37] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in *Proc. of 2009 ACM SIGMOD International Conference on Management of data*. ACM, 2009, pp. 139–152.
[38] S. Lian, Z. Liu, R. Zhen, and H. Wang, "Commutative watermarking and encryption for media data," *Optical Engineering*, vol. 45, no. 8, pp. 080 510–080 510, 2006.
[39] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. De Natale, and A. Neri, "A joint digital watermarking and encryption method," in *Electronic Imaging*. International Society for Optics and Photonics, 2008, pp. 68 191C–68 191C.
[40] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. De Natale, and A. Neri, "A commutative digital image watermarking and encryption method in the tree structured haar transform domain," *Signal Processing: Image Communication*, vol. 26, no. 1, pp. 1–12, 2011.
[41] A. Lemma, S. Katzenbeisser, M. Celik, and M. Van Der Veen, "Secure watermark embedding through partial encryption," in *Digital Watermarking*. Springer, 2006, pp. 433–445.
[42] J. Zhang, Y. Xiang, W. Zhou, L. Ye, and Y. Mu, "Secure image retrieval based on visual content and watermarking protocol," *The Computer Journal*, vol. 54, no. 10, pp. 1661–1674, 2011.
[43] T. Bianchi and A. Piva, "Secure watermarking for multimedia content protection: A review of its benefits and open issues," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 87–96, 2013.
[44] M. U. Celik, A. N. Lemma, S. Katzenbeisser, and M. Van Der Veen, "Lookup-table-based secure client-side embedding for spread-spectrum watermarks," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 3, pp. 475–487, 2008.
[45] A. Piva, T. Bianchi, and A. De Rosa, "Secure client-side st-dm watermark embedding," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 1, pp. 13–26, 2010.
[46] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.
[47] Wikipedia, "Streamcipher," 2015, [Online; accessed 29-November-2015].
[48] K. Gopalakrishnan, N. Memon, and P. L. Vora, "Protocols for watermark verification," *IEEE MultiMedia*, no. 4, pp. 66–70, 2001.
[49] C.-L. Lei, P.-L. Yu, P.-L. Tsai, and M.-H. Chan, "An efficient and anonymous buyer-seller watermarking protocol," *IEEE Transactions on Image Processing*, vol. 13, no. 12, pp. 1618–1626, 2004.
[50] Z. Xia, X. Wang, X. Sun, Q. Liu, and N. Xiong, "Steganalysis of lsb matching using differences between nonadjacent pixels," *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 1947–1962, 2016.
[51] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.
[52] H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-based image retrieval: overview and proposals," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 593–601, 2001.

Zhihua Xia received the BS degree in Hunan City University, China and PhD degree in computer science and technology from Hunan University, China, in 2006 and 2011, respectively. He works as an associate professor in School of Computer & Software, Nanjing University of Information Science & Technology. His research interests include digital forensic and encrypted image processing. He is a member of the IEEE.

Xinhui Wang received her BS in software engineering from Nanjing University of Information Science & Technology in 2012, China. She is currently pursuing her MS in computer science and technology at the College of Computer and Software, in Nanjing University of Information Science & Technology, China. Her research interest is privacy protection in cloud computing.

Liangao Zhang received his BS in network engineering from Nanjing University of Information Science & Technology in 2013, China. He is currently pursuing his MS in computer science and technology at the College of Computer and Software, in Nanjing University of Information Science & Technology, China. Her research interest is privacy protection in cloud computing.



Zhan Qin is a PhD candidate at Department of Computer Science and Engineering of State University of New York at Buffalo. He received his B.E in information engineering from Beijing Institute of Technology in 2010, his M.Sc. in electronic engineering from Columbia University in 2012. He has worked at Microsoft Research Asia in the summer of 2011. His research interests focus on security and privacy of cloud computing.

Xingming Sun received his BS in mathematics from Hunan Normal University, China, in 1984, MS in computing science from Dalian University of Science and Technology, China, in 1988, and PhD in computing science from Fudan University, China, in 2001. He is currently a professor in China-USA Computer Research Center, China. His research interests include network and information security, digital watermarking, and data security in cloud.

Kui Ren (M'07-SM'11-F'16) is a professor of Computer Science and Engineering and the director of UbiSeC Lab at State University of New York at Buffalo. He received his PhD degree from Worcester Polytechnic Institute. Kui's current research interest spans Cloud & Outsourcing Security, Wireless & Wearable Systems Security, and Mobile Sensing & Crowdsourcing. His research has been supported by NSF, DoE, AFRL, MSR, and Amazon. He was a recipient of SEAS Senior Researcher of the Year in 2015, Sigma Xi/IIT Research Excellence Award in 2012, and NSF CAREER Award in 2011. Kui has published 150 peer review journal and conference papers and received several Best Paper Awards including IEEE ICNP 2011. He currently serves as an associate editor for IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Mobile Computing, IEEE Wireless Communications, IEEE Internet of Things Journal, and IEEE Transactions on Smart Grid. Kui is a Fellow of IEEE, a Distinguished Lecturer of IEEE, a member of ACM, and a past board member of Internet Privacy Task Force, State of Illinois.