Cheng et al. EURASIP Journal on Information Security (2016) 2016:1
DOI 10.1186/s13635-015-0028-6

EURASIP Journal on Information Security
a SpringerOpen Journal

RESEARCH Open Access

Markov process-based retrieval for encrypted JPEG images

Hang Cheng[1,2*], Xinpeng Zhang[1], Jiang Yu[1] and Fengyong Li[3]

Abstract

This paper develops a retrieval scheme for encrypted JPEG images based on a Markov process. In our scheme, the stream cipher and permutation encryption are combined to encrypt discrete cosine transform (DCT) coefficients for protecting JPEG image content's confidentiality. And thus, it is easy for the content owner to achieve the encrypted JPEG images uploaded to a database server. In the image retrieval stage, although the server does not know the plaintext content of a given encrypted query image, he can still extract image feature calculated from the transition probability matrices related to DCT coefficients, which indicate the intra-block, inter-block, and inter-component dependencies among DCT coefficients. And these three types of dependencies are modeled by the Markov process. After that, with the multi-class support vector machine (SVM), the feature of the encrypted query image can be converted into a vector with low dimensionality determined by the number of image categories. The encrypted database images are conducted similarly. After low-dimensional vector representation, the similarity between the encrypted query image and database image may be evaluated by calculating the distance of their corresponding feature vectors. At the client side, the returned encrypted images similar to the query image can be decrypted to the plaintext images with the help of the encryption key.

Keywords: Image retrieval, Image encryption, Markov process, JPEG

1 Introduction

As cloud computing becomes increasingly popular, more and more customers want to outsource their multimedia data into the cloud server for cost saving and flexibility. In order to protect privacy of sensitive data, the content owners tend to convert the multimedia data into unrecognizable data before outsourcing, which may be not compliant with plaintext-based traditional information retrieval techniques. So, the need for privacy-preserving and effective information retrieval becomes urgent.

The problem of information retrieval in encrypted domain has been investigated for many years. In particular, secure searchable mechanisms for text documents have become an active research area that considers an application scenario where the content owners supply encrypted text documents and indices, while the servers provide search service without knowing the plaintext contents. In early studies, secure searchable encryption mechanisms only support conventional Boolean keyword search. In [1], Song et al. propose their early method to determine whether a given query keyword exists in encrypted documents by using two-level encryption based on single word. To do better in efficiency, Goh [2] constructs an efficient secure index based on Bloom filter for each encrypted text document. Using this method, the search performance that is linear to the number of database files can be obtained while supporting addition and deletion of files in dynamic fashion. Curtmola et al. [3] employ a keyword-based inverted index to aid the system for returning the files containing the required query keyword at a faster search speed compared to the method [2]. Differing from the previous schemes [1–3] based on symmetric encryption, the scheme by Boneh et al. [4] is the first to address the problem of the privacy-assured text retrieval with a public key system and allows multiusers to participate. Recent studies mainly focus on the diversity of functionalities for secure searchable systems, such as multidimensional range search [5], multi-keyword ranked search [6], privacy-assured similarity search [7], and so on [8, 9].

* Correspondence: hcheng@fzu.edu.cn
[1]School of Communication and Information Engineering, Shanghai University, Shanghai, China
[2]College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China
Full list of author information is available at the end of the article

For the same application scenario, there are few schemes to perform image retrieval in the encrypted domain. Lu et al. [10] introduce the three schemes to realize image retrieval in encrypted domain without first decrypting a query image of a ciphertext, where the encryption methods can remain the original relevance between the images approximately. In another work [11], two efficient secure search indices instead of visual features adopted by [10] are built by exploiting order-preserving encryption and min-hash function, respectively. Based on the works [10, 11], the literature [12] gives a comprehensive discussion on privacy-preserving image retrieval. With these methods [10–12], the content owners can extract the features from the images and then encrypt the features and images separately before outsourcing. However, the feature extraction/encryption will lead to higher computation overload and greater storage cost for users. In contrast, the authors [13] propose a histogram-based retrieval method for encrypted JPEG images, in which the feature extraction/encryption is needless. But there is one disadvantage with this scheme, namely, file size for the encrypted image will increase as discrete cosine transform (DCT) coefficients are shuffled.

It is known that there exist the intra-block, inter-block, and inter-component dependencies among DCT coefficients of a color JPEG image. Moreover, in some sense, the three types of dependencies are similar between similar images. Based on the analysis given above, we propose a novel scheme for encrypted JPEG images, where intra-block, inter-block, and inter-component dependencies among DCT coefficients are introduced. With this scheme, the encrypted JPEG images can be obtained through a combination of the stream cipher and permutation encryption and outsourced to a server. And also, with the given encrypted query image and the encrypted database images, it is easy for the server to calculate their similarities in encrypted domain by employing the techniques of a Markov process and multi-class support vector machine (SVM). In general, the contributions of this paper are summarized as follows:

1. The major originality of this paper lies that it utilizes the intra-block, inter-block, and inter-component dependencies of the encrypted color JPEG image to construct retrieval feature and reduces feature dimensionality by combining with multi-class SVM.
2. For a color JPEG image, this scheme can ensure both format compliance and file size preservation before and after encryption.
3. The scheme realizes privacy-preserving effective image retrieval in encrypted domain. Experiment results show that the overall retrieval performance of the proposed scheme is significantly improved.

The remainder of the paper is organized as follows. The scheme of the proposed method is elaborated in Section 2. Experimental results and analysis are given in Section 3. Finally, some conclusions are drawn in Section 4. The paper is an extended version of [14], where inter-component dependency of color JPEG image is newly employed to further improve the retrieval performance. And also, the paper also includes novel performance result and comparison.

2 Proposed scheme

Consider a privacy-preserving image retrieval scheme which involves three parties: content owner, authorized user, and server. The content owner encrypts images in the JPEG format and then stores them into cloud servers. The authorized user, may be a content owner, has desire to retrieval images similar to the encrypted query image from encrypted database images. When receiving the encrypted query image, the server can calculate the distances between the encrypted query image and database images and then returns encrypted images similar to the query image in plaintext content, without knowing anything about the plaintext contents of the involved encrypted images. In the following, we will present the mechanisms of image encryption and retrieval in detail.

2.1 Image encryption

As the purpose of our scheme is to address the problem of image retrieval in encrypted domain while preserving the file size and format compliance for JPEG images, here, we first take a partial image encryption technique into account to encrypt JPEG images. The problem is difficult to solve for the traditional cryptography. The most existing partial encryption techniques [15–17] for JPEG images are mainly based on blocks shuffle, DCT coefficient permutation, and encrypting the signs of DCT coefficients. Recent work by Qian et al. [18] presents a novel partial encryption method based on a JPEG bit-stream, which aims to implement reversible data hiding in an encrypted gray JPEG image. The proposed encryption method in [18] cannot only meet the requirements of format compliance and file size preservation but also provide valuable information regarding the length of each variable length integer (VLI) code for DCT coefficients. More importantly, the encryption method in [18] can make the length of each VLI code remain unchanged before and after encryption. It means that one can still obtain the original length of any VLI code related to DCT coefficients from an encrypted JPEG image. Due to the dependencies of DCT coefficients in each component, their corresponding VLI code length may have similar relationships, which can be exploited to generate feature for image retrieval. Therefore, this paper

designs a novel retrieval scheme for encrypted JPEG images, in which the encryption method in [18] is extended to color JPEG image.

Before presenting the details of image encryption, it is necessary to show the encoding process of a color JPEG image as most of the operations in the proposed scheme, such as image encryption and feature generation, are based on the encoded binary sequences of the DCT coefficients. The color JPEG-encoding process is summarized as follows.

As commonly known, a color JPEG image is composed of Y, U, and V components, each of which is partitioned into non-overlapped blocks sized 8×8. In each block, there are 64 DCT coefficients, namely, one DC and 63 AC coefficients. According to JPEG standard [19], DC and AC coefficients can be transformed into intermediate symbols by utilizing the one-dimensional predictor and the run length coding (RLC), respectively, and then are further Huffman-coded into binary sequences, each of which consists of two parts: the Huffman code and the VLI code. Obviously, the generation of the above-mentioned binary sequences is conditioned by the Huffman and VLI code tables, which are beforehand stored in the JPEG file header. In general, the Huffman code of the DC coefficient only contains the information about the length of the VLI code. But the Huffman code for the AC coefficient also has other information about the number of consecutive zero AC coefficients before the next nonzero AC coefficient in the zigzag sequence. The final JPEG bit-stream will be formed by concatenating the JPEG file header and binary sequences of all DCT coefficients of all components. As a matter of fact, the JPEG bit-stream is also a binary sequence and thus converts into a JPEG file when writing to a file byte by byte.

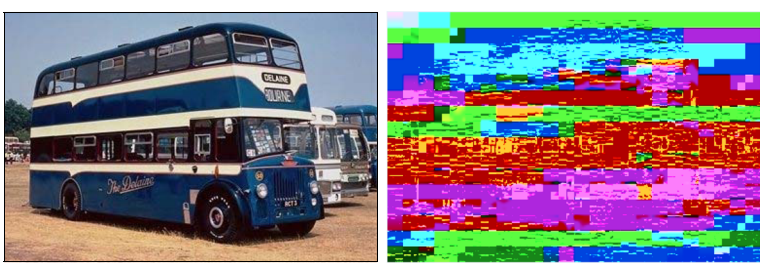
Based on the above knowledge about color JPEG image encoding, the procedure of performing the color JPEG image encryption is described as follows.

First, parse the JPEG bit-stream and extract entropy-encoded binary sequences associated with DCT coefficients for each certain component. The step is easily handled according to the JPEG file header information comprising image size, Huffman tables, and so on. Next, concatenate the VLI codes of all the AC coefficients of the Y component to form a new binary sequence and then encrypt by using the stream cipher. For example, denote the binary sequence needed to be encrypted as $A = [a_1, a_2, \ldots, a_l]$ and let a pseudo-random binary sequence be $E = [e_1, e_2, \ldots, e_l]$ with the same bit length as the binary sequence A. Then calculate the encrypted version $A' = [a_1', a_2', \ldots, a_l']$ through an exclusive-or operation

$$a_i' = a_i \oplus e_i, \qquad i = 1, 2, \ldots, l \tag{1}$$

where e_i is determined by an encryption key using a standard stream cipher. Instead of the bits a_i, the encrypted bits a_i' are placed at the original positions of bits a_i.

The U and V components can be done similarly. But different encryption keys are adopted. After that, encrypt the quantization tables stored in the JPEG file header by using the stream cipher. In brief, the binary sequences with respect to the quantization tables from the file header are encrypted according to formula (1). In particular, different encryption keys are employed to different quantization tables for protecting the privacy of the quantized DCT coefficients. For better encryption, we further pseudo-randomly permute encoded binary sequences of DC coefficients in a same component when keeping their frequency position intact. And also, the encrypted bits within the same binary sequence stay their original unencrypted positions. Finally, an encrypted JPEG bit-stream is generated. Figure 1 shows an original plaintext JPEG image and its encrypted version. Since the JPEG file structure, Huffman codes, and length of JPEG bit-stream remain intact, the resulting encrypted file is still in the JPEG format while keeping the file size unchanged, which means that the file size and format compliance are preserved. With the help of unencrypted Huffman/VLI coding tables stored in the JPEG file header, one can readily decompress the encrypted JPEG image. In addition, since the proposed encryption method can preserve statistical



(a) (b)

Fig. 1 **a** An original plaintext JPEG image and **b** its encrypted version

invariance of the lengths of the VLI codes with respect to the DCT coefficients before and after encryption, our scheme can support different multiple users with different encryption keys.

2.2 Image retrieval

The key idea behind our retrieval mechanism is that color JPEG images belonging to the same category are characterized by the similar intra-block, inter-block, and inter-component dependencies among DCT coefficients. With the proposed retrieval mechanism, the server without the encryption keys can perform image retrieval in encrypted domain. The details of the image retrieval process are as follows.

Considering an encrypted image, the server first parses its corresponding encrypted JPEG bit-stream to extract all Huffman codes for DCT coefficients of each component. The extraction operation, in this step, is readily accomplished because of file format compliance before and after encryption. Next, exploiting the Huffman tables obtained from the file header to decode Huffman codes, we can obtain the length of each VLI code next to the Huffman code. That is, any DC or nonzero AC coefficient can be represented as a nonnegative integer that is equal to the length of the corresponding VLI code. Meanwhile, the number of consecutive zero-valued AC coefficients between two adjacent nonzero AC coefficients can also be achieved during the Huffman decoding. Based on the above results, we can construct the three two-dimensional matrices, denoted by D^{Y}, D^{U}, and D^{V}, respectively, with the same size as the corresponding components, i.e., Y, U, and V components. More specifically, an element of each matrix is one-to-one mapping to a DCT coefficient of a corresponding component in the same position. The mapping rule is that the elements for DC and zero-valued AC coefficients are set zero and those for nonzero AC coefficients are represented by the lengths of related VLI codes separately. As described above, the three matrices, in fact, are derived from the DCT coefficients of the Y, U, and V components, respectively. Furthermore, there exist intra-block and inter-block dependencies among DCT coefficients [20, 21]. Therefore, it is logical and reasonable to argue that these dependencies can also be found in the matrices D^x ($x \in \{\mathrm{Y}, \mathrm{U}, \mathrm{V}\}$) separately. As is well known, the natural color JPEG image has an intrinsic inter-component dependency, which is beneficial to clarify the difference between different images. So, except for intra-block and inter-block dependencies originated from a certain component, this paper also considers the inter-component dependency in order to further improve retrieval performance. In short, the dependencies among the corresponding matrices D^x ($x \in \{\mathrm{Y}, \mathrm{U}, \mathrm{V}\}$) are considered as well. Inspired by [20, 21], the above three types of dependencies can be modeled by using a first-order Markov process and followed by transition probability matrix reflection. The elements of the resulting transition probability matrices serve as features for image retrieval. The features, in this paper, are mainly composed of three parts: intra-block, inter-block, and inter-component features. The feature extraction is done through the following three steps.

Step 1: Extract intra-block features. We explain the feature extraction only from the matrix D^{Y} as the other two matrices are processed similarly. To reduce the complexity, the values of elements of the matrix D^{Y} are set to T when they are larger than the threshold T. In other words, the range of values of elements in the matrix D^{Y} is [0, T].

Without loss of generality, let $d\ m,n(u,v)$ denote (u,v)th element in the 8×8 blocks of the matrix D^{Y} sized $M \times N$, where $u,v = 0, \ldots,7$. m and n are the indices of blocks ($1 \le m \le M$, $1 \le n \le N$). Again, we denote the pair $d\ m,n(u_k,v_k)$ and $d\ m,n(u_{k+1},v_{k+1})$ $(0 \le k \le 62)$ as two immediately neighboring elements in a 8×8 block along the zigzag scanning order, where u_k and v_k indicate the coordinates in the block for the kth element in zigzag order ($0 \le u_k, v_k \le 7$). For a given k, the value of u_k (or v_k) is fixed. As shown in Fig. 2, the two green dots are denoted as the 2nd ($k = 1$) and 3rd ($k = 2$) elements along the zigzag scanning direction (the red arrows), respectively. Based on the above definition, we can know that the coordinates of these two elements within the 8×8 block are, correspondingly, $(u_1,v_1) = (0,1)$ and $(u_2,v_2) = (1,0)$. Excluding the element related to DC coefficient, we can obtain 62 mode element pairs. In this case, the value of k ranges from 1 to 62 and just maps to number the order of the element pair in zigzag order, i.e., the kth element pair refers to the pair $d\ m,n(u_k,v_k)$ and $d\ m,n(u_{k+1},v_{k+1})$. Applying the Markov process to model each element pair, we will gain 62 transition probability matrices of size $(T+1) \times (T+1)$. For kth element pair, its corresponding transition probability matrix is calculated by

$$\begin{aligned}\mathbf{D}_{u_k,v_k,u_{k+1},v_{k+1}}(x,y) &= \Pr\left(d_{m,n}(u_{k+1},v_{k+1}) = y \middle| d_{m,n}(u_k,v_k) = x\right) \\ &= \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\delta\left[d_{m,n}(u_k,v_k) = x, d_{m,n}(u_{k+1},v_{k+1}) = y\right]}{\sum_{m=1}^{M}\sum_{n=1}^{N}\delta\left[d_{m,n}(u_k,v_k) = x\right]}\end{aligned} \tag{2}$$

and

$$\delta(C) = \begin{cases} 1, & \text{if } C \text{ holds} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $x, y \in [0, T]$. To reduce the feature dimensionality, we average the transition probability matrices generated

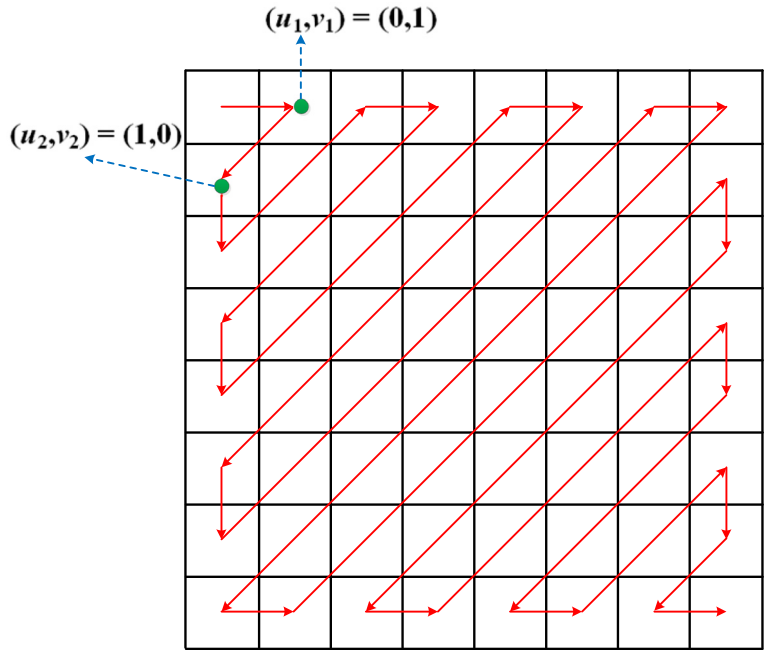


Fig. 2 Example of two immediately neighboring elements along the zigzag scanning direction

from the 62 element pairs to generate a new matrix with $(T+1)\times(T+1)$ elements, all of which are used as features. Thus, we have a $(T+1)\times(T+1)$ dimensional feature vector for the matrix D^{Y}. In a similar manner, we can obtain another two feature vectors for the matrices D^{U} and D^{V} separately. Concatenate the three feature vectors to form a total of $3\times(T+1)\times(T+1)$ dimensional feature vector, which is treated as intra-block features.

Step 2: Extract inter-block features. For simplicity, we still take the matrix D^{Y} as an example to illustrate inter-block feature extraction. In this step, we aim to find the dependency among the elements located in the same position within all 8×8 blocks. To this end, we operate the matrix D^{Y} in a way similar to that in step 1. Based on formula (4), we construct the transition probability matrices for mode element pairs, each of which is defined as two elements with the same position of two adjacent blocks along horizontal or vertical directions.

$$\begin{aligned}\mathbf{D}_{m,n,\Delta m,\Delta n,u,v}(x,y) &= \Pr\left(d_{m+\Delta m,n+\Delta n}(u,v)=y \middle| d_{m,n}(u,v)=x\right)\\ &= \frac{\sum_{m=1}^{M-\Delta m}\sum_{n=1}^{N-\Delta n}\delta\left[d_{m,n}(u,v)=x, d_{m+\Delta m,n+\Delta n}(u,v)=y\right]}{\sum_{m=1}^{M-\Delta m}\sum_{n=1}^{N-\Delta n}\delta\left[d_{m,n}(u,v)=x\right]}\end{aligned} \tag{4}$$

where $x, y \in [0, \mathrm{T}]$ and $\delta(\cdot)=1$ if and only if its arguments are satisfied. $m=1$ and $n=0$ for horizontal direction, and $m=0$ and $n=1$ for vertical direction. For each direction, there exist 63 mode element pairs based on the 63 different positions of AC coefficients in an 8×8 block. Considering the complexity, we will separately calculate the average of horizontal and vertical transition probability matrices and then jointly form $2\times(T+1)\times(T+1)$ dimensional feature vector. Similarly, the inter-block features for the matrices D^{U} and D^{V} can also be obtained. The final feature vectors with $3\times2\times(T+1)\times(T+1)$ dimensions are produced and referred to as inter-block features.

Step 3: Extract inter-component features. In this paper, the color JPEG images are based on a YUV 4:1:1 format, so the size of the Y component is four times that of the U or V components, namely, every four adjacent 8×8 blocks of the Y component corresponds to an 8×8 block of the U component and an 8×8 block of the V component. The mapping relation is maintained among the matrices D^{i} ($i\in\{\mathrm{Y},\mathrm{U},\mathrm{V}\}$) as well. We partition a 16×16 block into four non-overlapped 8×8 blocks, denoted as Y_1, Y_2, Y_3, and Y_4, respectively. Here, Y_i ($i\in\{1,2,3,4\}$) only represents a specific 8×8 block in the ith position in the 16×16 block. For example, Y_1 represents the top left 8×8 block in the 16×16 block. By utilizing the Y_i ($i\in\{1,2,3,4\}$) representation, the matrix D^{Y} is decomposed into four sub-matrices D_i^{Y} ($i\in\{1,2,3,4\}$), each of which only contains the same type of block and has the same size as D^{U} or D^{V} with $(M/2)\times(N/2)$ dimensions. Below, the inter-component features are calculated to capture the correlation between two components, namely,

{Y, U}, or {Y, V}, or {U, V}. We denote by $d(s)$ $m,n(u,v)$ the (u,v)th element in the (m,n)th 8×8 block from some matrix $D^{(s)}$, where $u, v \in [0, 7]$, $m \in [1, M/2]$, $n \in [1, N/2]$, and $s \in \{Y_1, Y_2, Y_3, Y_4, U, V\}$. For specific u, v, and all m,n, we can obtain the transition probability matrices given as follows:

$$\begin{aligned}\mathbf{D}^{(s_1,s_2)}_{m,n,u,v}(x,y) &= \Pr\left(d^{(s_1)}_{m,n}(u,v) = x | d^{(s_2)}_{m,n}(u,v) = y\right) \\ &= \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\delta\left[d^{(s_1)}_{m,n}(u,v) = x, d^{(s_2)}_{m,n}(u,v) = y\right]}{\sum_{m=1}^{M}\sum_{n=1}^{N}\delta\left[d^{(s_2)}_{m,n}(u,v) = y\right]}\end{aligned} \tag{5}$$

where $x, y \in [0, T]$, $s_1 \in \{Y_1, Y_2, Y_3, Y_4, U\}$, $s_2 \in \{U, V\}$, and $\delta(\cdot)$ as step 2. And also, s_2 should be set to V when $s_1 =$ U. Note that we calculate the average of four matrices, shown in formula (6), to capture the inter-component relationships between the Y and U components. The similar operation is done between the Y and V components, but not required for the U and V components.

$$\mathbf{D}^{(Y,U)}_{m,n,u,v} = \left[\mathbf{D}^{(Y_1,U)}_{m,n,u,v} + \mathbf{D}^{(Y_2,U)}_{m,n,u,v} + \mathbf{D}^{(Y_3,U)}_{m,n,u,v} + \mathbf{D}^{(Y_4,U)}_{m,n,u,v}\right]/4 \tag{6}$$

Corresponding to various u and v except for $(u,v) = (0,0)$, 63 transition probability matrices are formed for each component pair, such as {Y, U}, {Y, V}, and {U, V}. Similarly, in order to lower feature dimensions, we average the 63 matrices for the three component pairs separately. Finally, we utilize the resulting three tradition probability matrices with size $(T+1) \times (T+1)$ to form a $3 \times (T+1) \times (T+1)$ dimensional feature vector viewed as inter-component features for an encrypted color JPEG image.

Based upon the above three steps, we can transform any encrypted color JPEG image into a feature vector with $12 \times (T+1) \times (T+1)$ dimensions. Here, $T = 7$ is recommended because the VLI code length of the AC coefficient is in general less than 7. A statistical result in 1000 JPEG images from Corel image database [22] shows that about 99.89 % of all elements of the matrix D^Y are not larger than 7. The threshold operation results in $12 \times (7+1) \times (7+1) = 768$ dimensional feature vector, which can capture the intra-block, inter-block, and inter-component dependencies of any color JPEG image, and serves for image retrieval.

For further reduced computing complexity and feature dimension, this paper introduces a supervised mechanism for image retrieval when a training image set is available. The mechanism is implemented by utilizing supporting vector machine (SVM) that is widely used to solve the problem of pattern recognition and machine learning. The main idea of SVM is to find a hyperplane to separate the two classes for the purpose of maximizing the distance between the nearest point and the hyperplane. Denote training sample set as $\{(x_i, y_i), i = 1, 2,..., m\}$. x_i is a feature vector of real numbers, and y_i indicates the class of x_i, i.e., −1 or +1. Then the desired optimal hyperplane can be achieved by solving the following optimization problem:

$$\begin{aligned}&\min_{\omega,b} \quad \left(1/2\|\omega\|^2\right) + C\sum_{i=1}^{m}\xi_i \\ &\text{s.t.} \quad y_i(\omega \cdot \Phi(x_i) + b) \geq 1 - \xi_i\end{aligned} \tag{7}$$

where $\xi_i \geq 0$, $i = 1, 2,..., m$. $\Phi(x)$ is a nonlinear function and has the capability to map feature vector from low-dimensional space to high-dimensional space. When learning ω and b from the training sample set, we can make a class prediction for a new sample (x, y) according to the sign of $\omega \cdot \Phi(x) + b$. In general, $y = 1$ if $\omega \cdot \Phi(x) + b > 1$, and $y = -1$ if $\omega \cdot \Phi(x) + b \leq -1$.

The multi-class SVM is an extension of SVM and can solve multi-class pattern recognition problems, where the class number is larger than two. At present, a common method to implement multi-class SVM is that multi-class problems are decomposed into a set of two class problems solved by SVM. This way, the final result of a multi-class problem can be achieved according to all involved SVM's individual decisions. Considering the diversity of image class in image retrieval, this paper uses multi-class SVM to solve the problem of high-computing complexity and feature dimensionality. More specifically, let w the number of categories of images in the training set. First, transform encrypted images of the training set to feature vectors and then train w SVM models by performing multi-class SVM, where the one-against-all strategy is adopted, i.e., when training the ith SVM model, all images belonging to the ith category are labeled positive and the labels of the remaining images are set negative. After obtaining the w SVM models, the server maps a 768-dim feature vector representing a color JPEG image to a new w-dim vector, each element of which as the decision value of the corresponding SVM model. In general, the value of w is less than the dimension of the original feature vector. For a pair of similar images, their corresponding w-dim vectors are close to each other. When having w-dim feature representation, the server may calculate the distance between the encrypted query image and the database image by the following formula:

$$d(Q,F) = \sum_{i=1}^{w}\frac{|q_i - f_i|}{|1 + q_i + f_i|} \tag{8}$$

where $Q = [q_1,..., q_w]$ is a w-dim vector associated with an encrypted query image and $F = [f_1,..., f_w]$ for any

encrypted image in database. Based on $d(Q,F)$, the server can identify the similarities between the query image and images in the image database.

Subsequently, the encrypted images ranked by the distances to the query image are returned to the users for decrypting and other purposes. It should be important to note that the users may use different encryption keys to encrypt query image but require the encryption keys shared by the content owner to decrypt them. Here, assume that the encryption keys are shared through a secure channel.

3 Experimental results

In this section, we study the performance of the proposed scheme for image retrieval in encrypted domain. The Corel image database containing 1000 images [22] is used for this experimental study. These images are grouped into 10 categories: African, beach, architecture, buses, dinosaurs, elephants, flowers, horses, mountain, and food, each of which contains 100 images of size either 384×256 or 256×384 in JPEG format. In the experiments, we first encrypt all of the images by using the encryption method served for this paper. Next, employing the rule with taking randomly 50 images from each category, the 1000 images are randomly divided into two halves for the training set and the image database, respectively. Since the category amount of images for our experiment is 10, we can obtain 10 SVM models by applying multi-class SVM to train the training set. In other words, the database images can be transformed into 10-dim vectors in the experiment. Again, we select every image from the encrypted image database as the encrypted query image. Meanwhile, the performance is evaluated by utilizing the precision-recall curve, which is broadly adopted for image retrieval. The precision-recall curve, however, relies on the query image. Therefore, we take the average precision-recall curve for all query images as a final performance measure of an image retrieval algorithm. The precision and recall are defined as follows:

$$\text{Precision} = \frac{N_P}{N_R}, \quad \text{Recall} = \frac{N_P}{N_A} \tag{9}$$

where N_P is the number of returned positive images, N_R is the number of all returned images, and N_A is the number of all positive images in the database.

Figure 3 compares the precision-recall curves of different types of dependencies, i.e., intra-block, inter-block, inter-component, intra-block + inter-block, and intra-block + inter-block + inter-component dependencies. Here, the symbol "+" denotes a combination of two different types of dependencies. It can be seen from Fig. 3 that the methods by the combinations of different types of dependencies have shown stronger ability to find similar images than that by single

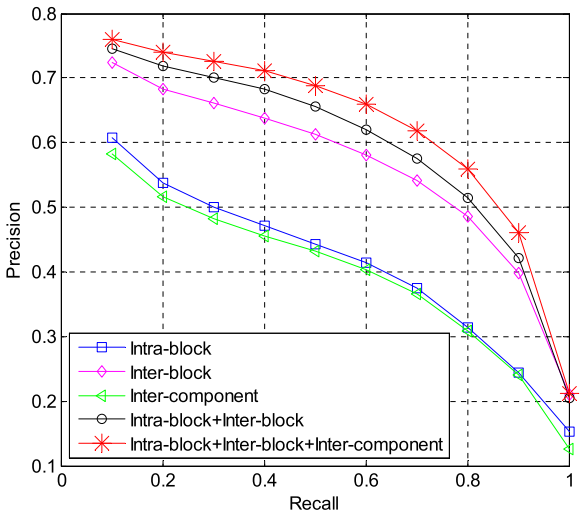


Fig. 3 Precision-recall curves with different types of dependencies

dependencies. Specially, the combination of the first three single dependencies can achieve the better result than intra-block + inter-block at all cases. Although the performance of the inter-component dependency is worst among all single dependencies, the inter-component dependency can reflect the correlation between gray value and color information at each pixel. Therefore, the inter-component dependency can further enhance the final performance when it is considered. In addition, we also observe that the performance of inter-block dependency outperforms the intra-block dependency over the provided Corel image database with an average gain more than 14 %. The main reason may be that these two types of dependencies employ the matrices D^x ($x \in \{Y, U, V\}$) to reflect intra- and inter-block dependencies. Based on subsection 2.2, we can know each element of these matrices is replaced by the length of the VLI code of its corresponding DCT coefficient at the same position. The replacement operation can conceal the actual values of the DCT coefficients, where it is very difficult to identify little changes between neighboring DCT coefficients within the block. This means that we may get the poor performance by employing the intra-block dependency, which can be shown in Fig. 3. However, for the inter-block dependency, it can reflect the overall structure of DCT coefficients of the JPEG image. And also, it is not sensitive to little changes between DCT coefficients compared to the intra-block dependency. That is why that the inter-block dependency can demonstrate better performance than the intra-block dependency.

To make the comparison fair, we make a performance comparison of our scheme and Zhang's method [13] in terms of the supervised mechanism. The comparative results are given in Fig. 4. It can demonstrate that our scheme with or without the inter-component dependency

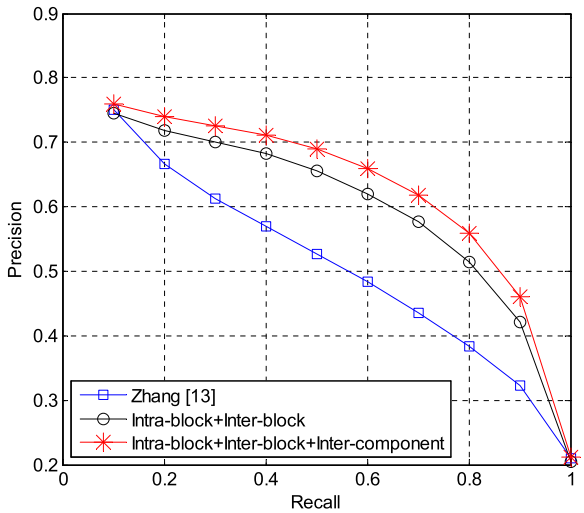


Fig. 4 Comparison with the method in [13]

can provide a better performance than Zhang's method in [13]. For example, when recall = 0.4, our scheme with the inter-component dependency can obtain a precision of 71.1 %, a precision value about 14.2 % higher than [13]. On average, our scheme with the inter-component dependency achieves nearly 12 % improvement than [13], while our scheme can also obtain 9 % improvement without consideration of the inter-component dependency. As for the file size, we also compare our scheme with Zhang [13] over all 1000 images of the Corel image database. As shown in Table 1, for each category, the average file size increase of our scheme is much less than that of Zhang's method. In fact, the proposed scheme can maintain file size unchanged. It is not surprising that little changes of the file size in our scheme are still found as this problem is mainly attributed to JPEG intrinsic structure and zero padding strategy, but not caused by the encryption method adopted in this paper. In the section, the comparisons between the proposed scheme and Lu's methods in [10–12] are not made due to the difference of their frameworks. In this paper, the feature extraction/encryption is needless at the client side. It brings convenience to the content owners or authorized users. All of Lu's methods in [10–12], however, require the content owner to perform the operations of feature extraction/encryption.

Table 1 Average file size increase with different methods

Image category	The file size increase (in bytes)	
	Proposed scheme	Zhang [13]
African	10.0	5081.4
Beach	10.5	6768.6
Architecture	8.1	6214.9
Buses	9.9	6053.7
Dinosaurs	1.5	9554.3
Elephants	12.4	5234.9
Flowers	5.3	6156.2
Horses	11.2	3960.1
Mountain	9.8	7236.9
Food	7.4	5447.3

In this paper, the proposed encryption mechanism extended from [18] mainly adopts the stream cipher to protect the privacy of JPEG images. It implies that the security of the proposed encryption method can be guaranteed by that of the stream cipher. At the same time, the pseudo-random permutation encryption is also applied to DC coefficients for increasing the security of the proposed encryption method. In addition, since the proposed encryption method can preserve statistical invariance of DCT coefficients' dependencies before and after encryption, our scheme can support different multiple users with different encryption keys. In this case, the same plaintext image can also be encrypted by different encryption keys. It means that several encrypted versions of a plaintext image are allowed to exist simultaneously in the proposed scheme, which not affects the normal operation of the proposed scheme. Through these above secure measures, it would be difficult for an adversary to perfectly infer the original content from an encrypted image.

4 Conclusions

A novel scheme for performing image retrieval in encrypted domain is introduced. With this scheme, the content owner can get encrypted images by utilizing stream cipher and permutation encryption and send them to the server for storage and retrieval service. After obtaining an encrypted query image, although the server does not know anything about the plaintext content of the query image, he/she can still extract a 768-dim feature vector from the encrypted query image by employing the transition probability matrices based on a Markov process, which reflect intra-block, inter-block, and inter-component dependencies of DCT coefficients. And then, with the aid of the multi-class SVM, the server can further reduce feature dimensions of the involved encrypted images and calculate the distances between the encrypted query image and the database images. Finally, the encrypted images ranked by the similarity to the query image are returned to the users. In future work, the image retrieval techniques for better precision-recall performance with various encryption methods deserve further investigation.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
HC carried out the main research of this work and drafted the manuscript. XZ, JY, and FL helped to modify the manuscript. All authors read and approved the final manuscript.

Acknowledgements
This work was supported by the National Natural Science Foundation of China under Grants 61472235, 61373151, and 61202367; the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, Shanghai Pujiang Program under Grant 13PJ1403200; and the Excellent University Young Teachers Training Program of Shanghai Municipal Education Commission under Grant ZZsdl15105.

Author details
[1]School of Communication and Information Engineering, Shanghai University, Shanghai, China. [2]College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. [3]College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China.

Received: 17 October 2015 Accepted: 22 December 2015
Published online: 04 January 2016

References

1. D Song, D Wagner, A Perrig, Practical techniques for searches in encrypted data, in *IEEE symp. on research in security and privacy* (IEEE, New York, 2000), pp. 44–55
2. EJ Goh, Secure indexes. IACR Cryptology ePrint Archive. 216 (2003)
3. R Curtmola, J Garay, S Kamara, R Ostrovsky, Searchable symmetric encryption: improved definitions and efficient constructions, in *Proceedings of the 13th ACM conference on computer and communications security* (ACM, New York, 2006), pp. 79–88
4. D Boneh, G Crescenzo, R Ostrovsky, G Persiano, Public-key encryption with keyword search, in *Advances in cryptology—Eurocrypt 2004* (Springer, Heidelberg, 2004), pp. 506–522
5. E Shi, J Bethencourt, TH Chan, D Song, A Perrig, Multi-dimensional range query over encrypted data, in *IEEE Symposium on Security and Privacy 2007* (IEEE, New York, 2007), pp. 350–364
6. N Cao, C Wang, M Li, K Ren, W Lou, Privacy-preserving multi-keyword ranked search over encrypted cloud data. Parallel and Distributed Systems, IEEE Trans **25**(1), 222–233 (2014)
7. C Wang, K Ren, S Yu, KMR Urs, Achieving usable and privacy-assured similarity search over outsourced cloud data, in *INFOCOM, 2012 Proceedings IEEE* (IEEE, New York, 2012), pp. 451–459
8. S Kamara, C Papamanthou, Parallel and dynamic searchable symmetric encryption, in *Financial cryptography and data security* (Springer, Berlin Heidelberg, 2013), pp. 258–274
9. D Cash, S Jarecki, C Jutla, H Krawczyk, M Rosu, M Steiner, Highly-scalable searchable symmetric encryption with support for Boolean queries, in *Advances in Cryptology-CRYPTO* (Springer, Berlin Heidelberg, 2013), pp. 353–373
10. W Lu, AL Varna, A Swaminathan, M Wu, Secure image retrieval through feature protection, in *Proc. of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, New York, 2009), pp. 1533–1536
11. W Lu, A Swaminathan, AL Varna, M Wu, International Society for Optics and Photonics. Enabling search over encrypted multimedia databases, in *IS&T/SPIE Electronic Imaging* (SPIE, 2009), pp. 725418-725418
12. W Lu, A Varna, M Wu, Confidentiality-preserving image search: a comparative study between homomorphic encryption and distance-preserving randomization. Access, IEEE **2**, 125–141 (2014)
13. X Zhang, H Cheng, Histogram-based retrieval for encrypted JPEG images, in *Signal and Information Processing (ChinaSIP), 2014 IEEE China Summit & International Conference on* (IEEE, New York, 2014), pp. 446–449
14. H Cheng, X Zhang, J Yu, F Li, Markov process based retrieval for encrypted JPEG images, in *Availability, Reliability and Security (ARES), 2015 10th International Conference on* (IEEE, Toulouse, 2015), pp. 417–421
15. S Lian, J Sun, Z Wang, A novel image encryption scheme based-on JPEG encoding, in *Proceedings of Eighth International Conference on Information Visualization* (IEEE, New York, 2004), pp. 217–220
16. B Yang, C Zhou, C Busch, X Niu, Transparent and perceptually enhanced JPEG image encryption, in *Digital Signal Processing, 2009 16th International Conference on* (IEEE, New York, 2009), pp. 1–6
17. W Li, N Yu, A robust chaos-based image encryption scheme, in *Multimedia and Expo, 2009. ICME 2009, IEEE International Conference on* (IEEE, New York, 2009), pp. 1034–1037
18. Z Qian, X Zhang, S Wang, Reversible data hiding in encrypted JPEG bitstream. Multimedia, IEEE Trans **16**(5), 1486–1491 (2014)
19. Int. Telecommunication Union, CCITT Recommendation T.81, Information technology: digital compression and coding of continuous-tone still images—requirements and guidelines 1992.
20. YQ Shi, C Chen, W Chen, A Markov process based approach to effective attacking JPEG steganography, in *Information hiding* (Springer, Berlin Heidelberg, 2007), pp. 249–264
21. C Chen, YQ Shi, JPEG image steganalysis utilizing both intrablock and interblock correlations, in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS 2008)* (IEEE, New York, 2008), pp. 3029–3032
22. The Corel image database, http://wang.ist.psu.edu/docs/related/.