Privacy Preserving Search of Multimedia

Wenjun Lu, Avinash L. Varna, Ashwin Swaminathan, and Min Wu
Department of Electrical and Computer Engineering
and Institute for Advanced Computer Studies
University of Maryland, College Park

Abstract

The advancement of information technology is rapidly integrating the physical world where we live and the online world where we retrieve and share information. One immediate example of such integration is the increasing popularity of storing and managing personal data using third-party web services, as part of the emerging trend of cloud computing. Secure management of sensitive data stored online is becoming one of the critical research issues in cloud computing and online privacy protection. In this paper, we propose techniques to achieve content based multimedia retrieval over encrypted databases, which can be used for online management of multimedia data while preserving data privacy. We propose two types of secure retrieval schemes by combining cryptographic techniques, such as order preserving encryption and randomized hash functions, with image processing and information retrieval techniques, such as visual words representation, inverted index, and min-hash. The first type of retrieval schemes scramble visual features extracted from images and allow similarity comparison of the features in their encrypted forms. The second type of schemes encrypt the state-of-the-art search indexes without significantly affecting their search capability. The two types of schemes are complementary and represent different trade-offs between user-side computational complexity and communication overhead. Retrieval results on an encrypted color image database and security analysis under different attack models show that retrieval performance comparable to conventional plaintext retrieval schemes can be achieved over encrypted databases while ensuring data confidentiality.

Index Terms: Content based image retrieval, secure search, secure cloud computing, visual words, min-hash, random projection, order preserving encryption

I. Introduction

The advancement of information technology has been rapidly integrating the physical world where we live and the online world that we rely on for retrieving, sharing, and managing information. Online services

Email: {wenjunlu, varna, minwu}@umd.edu, ashwins@gmail.com. Ashwin Swaminathan was with University of Maryland, College Park, when participating with this work; he is now with Qualcomm Research, San Diego, CA. Preliminary results from this work were published in SPIE Media Forensics and Security Conference, January 2009 [1] and ICASSP, April 2009 [2].

and web applications emerge everyday and benefit our life in almost every aspect: from information retrieval using search engines to sharing user generated content through social networks, from personal information management, such as webmail and online photo albums, to online backup services. With the arrival of the cloud computing paradigm, the Internet stores not only information for sharing, but also sensitive personal data that should be carefully protected against any unauthorized access. Secure management of personal data stored online is an increasingly important issue that can help achieve the data confidentiality and availability requirements of cloud computing. Technologies that can enable secure online data management are going to play a critical role in the future of the internet.

Traditional privacy protection for online personal data focuses on access control and secure data transmission, which ensure that the data can be securely transmitted to the server and no unauthorized people can access the data. However, once the data arrives at the server, the server decrypts the data and operates on plaintext in order to provide services to users, such as categorization, search, and data analysis. This makes the user's private information vulnerable to untrustworthy service providers and malicious intruders. For example, most personal emails are stored online as plaintext data and can be viewed by the system administrator. Given the trend that an increasing amount of personal data will be stored at a third-party server, it is both desirable and necessary to develop technologies that can better protect users' privacy without sacrificing the usability and accessibility of the information.

Information retrieval over encrypted databases is a promising technological capability for privacy protection in online information management. Encryption of the data stored on the server helps protect content privacy against untrustworthy service providers and malicious intruders, but using traditional cryptographic ciphers alone makes it difficult for the server to process the data, and for the user to retrieve information from the encrypted database. The goal of information retrieval over an encrypted database is to provide efficient and accurate search capability over encrypted documents without decrypting them first. An example application is that the user stores his/her private data in encrypted form on remote servers and later wants to search and retrieve data similar to a query in a privacy preserving manner, so that the server cannot learn the content of the query and the retrieved images. Due to the widespread use of digital cameras and portable camcorders, multimedia data constitute a significant part of today's personal data collections. Storing and managing this large volume of multimedia data online is becoming a desirable option for convenient data access anywhere anytime. Given such a trend of more semantic and multimedia-rich Internet, technologies that can enable content-based retrieval over encrypted multimedia databases will play an important role in helping people manage their multimedia data both effectively and securely. This is the main focus of the current paper.

A. Related work

Prior work in the area of information retrieval in the encrypted domain focused on text documents. Song et al. [3], Brinkman et al. [4], and Boneh et al. [5] explored Boolean search to identify whether a query term is present in an encrypted text document. Recent work by Swaminathan et al. [6] proposed a framework for rank-ordered search over encrypted text documents, so that documents can be returned in the order of their relevance to the query term. Secure text retrieval techniques can also be applied to keyword based search of multimedia data. However, keyword search relies on having accurate text description of the content already available, and its search scope is confined to the existing keyword set. In contrast, content-based search over an encrypted multimedia database provides more flexibility, whereby sample images, audios or videos are presented as queries and documents with similar audio-visual content in the database are identified.

An emerging area of work related to secure multimedia retrieval is secure signal processing, which aims at performing signal processing tasks while keeping the signals being processed secret. Erkin et al. [7] provided a review of related cryptographic primitives and some applications of secure signal processing in data analysis and content protection. However, applying cryptographic primitives to content-based multimedia retrieval is not straightforward. Effective multimedia retrieval typically relies on evaluating the similarity of two documents using the distance between their visual features, such as color histograms, shape descriptors, or salient points [8]. Traditional cryptographic primitives typically do not preserve the distance between feature vectors after encryption. In addition, efficiency and scalability are critical for multimedia retrieval but can be difficult to achieve using cryptographic primitives alone. Another work by Shashank et al. [9] addresses the problem of protecting the privacy of the query image when searching over a public database, where the images in the database are not encrypted. By appropriately formulating the query message and response message during multiple rounds of communication between the user and the server, the server is made oblivious to the actual search path and thus unaware of the query content.

Recent work in the general area of secure computation for privacy protection addressed different but related problems under various application settings [10]–[14]. Yiu et al. [10] considered privacy preserving range query over geospatial coordinates using a kd-tree. Extending such a technique to multimedia retrieval is difficult because features used for content-based multimedia retrieval are high dimensional vectors and kd-tree is known to be inefficient in high dimensional spaces. Wong et al. [11] proposed secure k-NN computation that exactly preserves the original dot-product between two vectors. This algorithm identifies the larger of the two distances but keeps the actual distance values secret from the server. Erkin et al. [12] and Sadeghi et al. [13] addressed the problem of privacy preserving face recognition, where one party wants to verify the existence of a given face image in a database hosted on another party's servers. The two parties want to keep their own data secret from each other. Additive homomorphic encryption

schemes are used to allow similarity computation in the encrypted domain. Similar techniques are also used by Jiang et al. [14] to identify the existence of similar text documents between two parties. As there have been no efficient homomorphic encryption schemes yet that allow both addition and multiplication, multiple rounds of communication between the two parties are required to compute the Euclidean distance between the query and each database entry. The main difference between the scenarios considered in the above mentioned related works [12]–[14] and in this paper is that in the current application, the user stores encrypted multimedia data on a remote server, and the server does not own the data and merely performs computation on the encrypted data on behalf of the user to allow efficient content-based retrieval over a potentially large database. The current high level of computational complexity and communication overhead of schemes being built on homomorphic encryption make them less feasible for the application considered in this paper. Efficient encryption schemes that can enable fast retrieval with little communication overhead are required for private database management over the cloud.

B. Contributions and paper organization

Our work focuses on content-based multimedia retrieval over encrypted databases, where both the query and database documents are encrypted and their privacy is protected, in contrast to the work of Shashank et al. [9] where the database images are publicly known. The techniques proposed in this paper enable efficient retrieval directly in the encrypted domain, without multiple rounds of communications between the user and the server, as compared to [12], [13]. We demonstrate the proposed techniques using image databases in this paper, although these techniques are applicable to other multimedia modalities such as video. By analyzing the requirements of secure retrieval scenarios, we propose retrieval techniques from two different viewpoints. The first group of techniques focuses on visual feature protection that allows similarity comparison among encrypted features, while the second group of techniques aim at encrypting the search indexes directly, where the search indexes are typically generated from visual features and carefully designed to enable efficient search over large multimedia databases. The two groups of techniques are complementary and represent different trade-offs between user-side computational complexity and communication overhead. We jointly exploit cryptography, image processing, and information retrieval techniques to ensure that the encrypted features and indexes preserve search accuracy. Our experiments on an encrypted color image database show that data confidentiality can be preserved while retaining good retrieval performance. The proposed schemes are computationally efficient and provably secure against semi-honest adversaries in the ciphertext only attack model. To the best of our knowledge, this work is among the first endeavors on content-based multimedia retrieval in the encrypted domain and has promising online applications for secure multimedia management.

The paper is organized as follows: Section II formulates the secure retrieval problem and analyzes

the properties of a desirable secure retrieval scheme. Section III presents the proposed secure retrieval schemes, based on feature protection and index encryption, respectively. Section IV summarizes experimental results on retrieval over an encrypted color image database, followed by security analysis under different attack models in Section V. Conclusions are drawn in Section VI.

II. Problem Formulation

In order to protect data privacy, images need to be encrypted before being transferred to the remote server. Image encryption can be done using state-of-the-art ciphers such as AES or RSA by treating images as ordinary data, or using image specific encryption techniques such as selective and format-compliant encryption [15], [16], [17] to enable post-processing such as transcoding of encrypted images. As these techniques are built upon established cryptographic encryption tools, it is computationally difficult for an adversary to decrypt the encrypted image files.

In modern image retrieval techniques, content similarity is typically evaluated using search indexes or visual features, such as color histograms and salient points, instead of comparing images pixel by pixel. Therefore, encryption of images alone is not sufficient for privacy protection because search indexes or image features in plaintext may reveal information about image content. For example, a color histogram with large values for the blue components would indicate the presence of sky or ocean, and SIFT descriptors [18] may reveal information about distinctive objects in the image. In order to be able to search through the encrypted database without leaking information from the plaintext search indexes or image features, we devise schemes to generate and appropriately encrypt image features and search indexes on the user side using a secret key and then transfer them to the server, where the encrypted features or indexes are used to evaluate image similarity. A system model for the secure search scenario is shown in Figure 1, where the left part depicts the database construction stage and the right part depicts the retrieval stage. In the overall system of secure image retrieval, images are encrypted using standard ciphers, and visual features or search indexes extracted from the images are protected by the encryption schemes proposed in this paper. During retrieval, the search index from the query image is suitably encrypted to allow similarity comparison in the encrypted domain. The block "Build search index" corresponds to encrypting the features in the feature protection schemes or building secure indexes in the secure indexing schemes, which will be described in Section III .

The first approach for secure image retrieval is to encrypt the feature vectors of each image and store those encrypted features on the server, as described in Section III-A. The server can use these encrypted features as naïve indexes if the database is small, or the server can build efficient indexes upon the encrypted features for improved search efficiency. Since the similarity of two images is typically measured by computing the distance between features extracted from them [8], the encryption of image

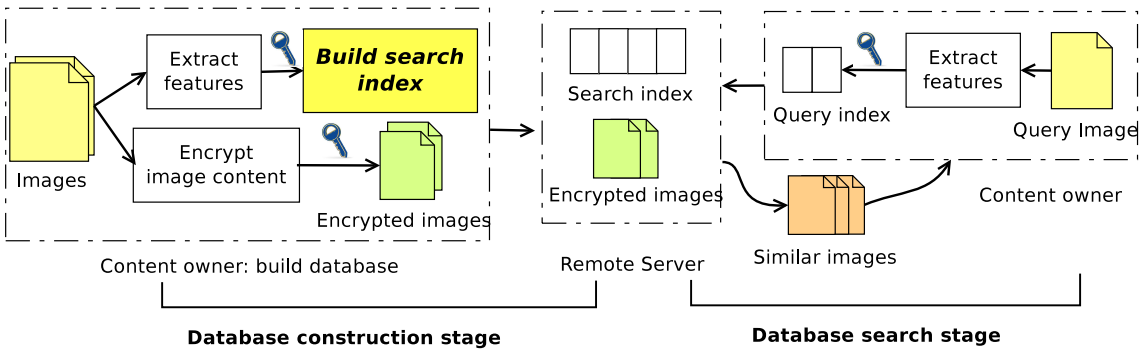

Fig. 1: System model for secure image retrieval

features should approximately preserve their distances. Suppose we represent image features as vectors in $\mathbb{R}^n$, we seek an encryption function $\mathcal{E}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, such that given two feature vectors $\mathbf{f}$ and $\mathbf{g}$, $d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g})) \approx c \cdot d(\mathbf{f}, \mathbf{g})$, where $d_{\mathcal{E}}(\cdot, \cdot)$ and $d(\cdot, \cdot)$ are some appropriate distance measures on ciphertext and plaintext, respectively, and c is a constant scaling factor. The approximate distance preserving encryption scrambles the visual features for content protection and allows servers to perform similarity comparison in the encrypted domain.

Since efficiency and scalability are critical aspects for retrieval from a large database and rely on the design of search indexes, the second approach for secure image retrieval explores the possibility of encrypting the state-of-the-art multimedia search indexes without affecting their search capability. During retrieval, the content owner who knows the secret key can generate a properly encrypted query index from the query image. The server then compares the encrypted query index with the stored indexes and returns the encrypted files of the most similar images to the user for decryption and viewing. Without knowing the secret key used for encryption, it should be difficult to search the database or infer the database content. The encrypted index also helps protect the privacy of the query image.

Secure image retrieval through feature encryption and index encryption are closely related. Image features themselves can be considered as a special form of search index, where each image is represented by its feature vectors and during retrieval, the query image's feature is compared to all features in the database. On the other hand, modern indexing schemes are built upon image features and allow efficient retrieval by reducing the number of images that need to be compared. Since the encrypted features preserve the capability of similarity comparison, they can be used to build efficient indexing schemes. The content owner has the flexibility either to provide the server with encrypted features and let the server perform the time-consuming index generation, or to generate the secure index on his/her side to reduce the amount of information that needs to be sent to the server. Therefore, the two kinds of approaches represent different trade-offs between user-side computational complexity and communication overhead.

III. Proposed Feature Encryption and Matching Techniques

In this section, we propose two categories of secure retrieval schemes. As discussed in the previous section, most state-of-the-art content-based image retrieval techniques utilize low-level visual features to represent and compare image content, and these visual features can potentially reveal important information about the image content. We first discuss feature protection schemes that enable similarity comparison between features in the encrypted domain. The encrypted features along with encrypted images can protect image content privacy against untrustworthy service providers and malicious intruders.

The ability to generate encrypted indexes on the user side provides an alternative for secure retrieval with reduced communication overhead. In the second part of this section, we discuss the protection of search indexes by exploiting the visual words representation of images [19]. Visual words method hierarchically clusters features into a vocabulary tree, following which each image is indexed based on this vocabulary tree and represented as a bag of visual words. Experiments on object recognition in the recent literature [19], [20] show that visual words based representation can be scaled to large databases. We propose two secure indexing schemes based on inverted index [21] and min-hash [22]. These two schemes protect information about the image content from the adversary and at the same time achieve efficient and scalable search capability.

A. Feature Protection

Three feature protection schemes are proposed in this paper with different trade-offs among computational complexity, retrieval performance, and security.

1) ***Bit-plane Randomization****:* The most significant bits (MSB) of an image capture important information about image appearance. The concept of processing bit-planes from MSBs to LSBs has been used in multimedia signal processing such as scalable encoding to provide fine granular trade-off between bitrate and quality. Feature vectors with small distances are likely to have similar patterns among their MSB bit-planes. This motivates us to investigate the scrambling of feature vectors based on a secret key, such that the patterns in the MSB bit-planes of the feature vectors are preserved for similarity comparison, but without knowing the secret key, the bit-planes cannot be decrypted to reveal the image content.

Given a feature vector $\mathbf{f} = [f_1, \cdots, f_n] \in \mathbb{R}^n$, each component f_i is represented in its binary form as $[b_{i1}, \cdots, b_{il}]^T$, where b_{i1} is the first MSB, b_{il} is the least significant bit (LSB) or the lth MSB, and l is the total number of bit-planes. For example, the 8-bit representation of "20" is 00010100, so the 1st MSB to the 8th MSB of 20 are 0, 0, 0, 1, 0, 1, 0, 0, respectively, and the LSB is 0. The j^{th} bit-plane of $\mathbf{f}$ is composed of the j^{th} MSB of the n feature components, denoted as $[b_{1j}, b_{2j}, \cdots, b_{nj}]$. The Hamming distance between two bit-planes is preserved when each bit-plane is XORed with the same binary vector or when each is permuted using the same permutation pattern. We exploit this property to encrypt the

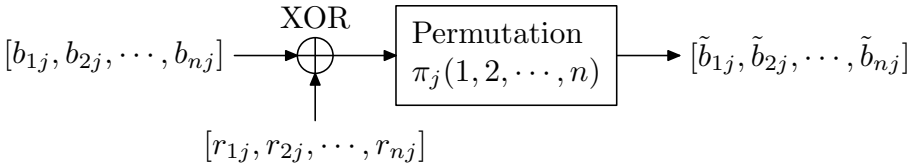

Fig. 2: Encryption of the j^{th} bit-plane

top k MSB bit-planes of the feature vectors while preserving the Hamming distances among encrypted bit-planes.

The encryption of the j^{th} bit-plane of a given feature vector is illustrated in Fig. 2. The bits comprising the bit-plane are first XORed with a pseudorandom binary sequence, which essentially randomly flips the value of each bit. As a result, each bit in the resulting bit-plane will be equally likely to be 0 or 1 and the number of '1's in the bit-plane will be approximately the same as the number of '0's. Hiding the original number of '1's in each bit-plane maximizes the entropy of the encrypted bit-planes and thus improves security. The resulting bits are then randomly permuted to obtain the encrypted bit-plane. Random permutation is an important step for improving security, which will be analyzed in Section V.

All the encrypted bit-planes are reassembled to form the encrypted feature vector $\mathcal{E}(\mathbf{f}) = [\tilde{f}_1, \cdots, \tilde{f}_n]$. Since the values $\{\tilde{f}_1, \cdots, \tilde{f}_n\}$ are completely random, traditional L_1 or L_2 norm does not capture the similarity between encrypted features. Instead, we compute the distance between two encrypted feature vectors $\mathcal{E}(\mathbf{f})$ and $\mathcal{E}(\mathbf{g})$ using a weighted sum of Hamming distances between their individual bit-planes:

$$d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g})) = \sum_{i=1}^{n} \sum_{j=1}^{l} |\tilde{b}_{ij}^{(\mathbf{f})} - \tilde{b}_{ij}^{(\mathbf{g})}| \times w(j). \tag{1}$$

Here, $\tilde{b}_{ij}$ is the i^{th} bit in the j^{th} encrypted bit-plane, and $w(j)$s are the weights assigned to the bit-planes to reflect their unequal importance, which are chosen to be 2^{-j} in this paper. Since using the same permutation and XOR pattern on corresponding bit-planes of two feature vectors preserves their Hamming distance, we have

$$d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g})) = \sum_{i=1}^{n} \sum_{j=1}^{l} |b_{ij}^{(\mathbf{f})} - b_{ij}^{(\mathbf{g})}| \times 2^{-j} \geq \sum_{i=1}^{n} \left| \sum_{j=1}^{l} (b_{ij}^{(\mathbf{f})} - b_{ij}^{(\mathbf{g})}) \times 2^{-j} \right| = \|\mathbf{f} - \mathbf{g}\|_1. \tag{2}$$

The distance $d_{\mathcal{E}}(\cdot, \cdot)$ between encrypted features is an upper bound on the L_1 distance between the original features. The bound is mostly tight but occasionally large errors between $d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g}))$ and $\|\mathbf{f} - \mathbf{g}\|_1$ may arise as some feature vectors with small L_1 distance may have large distance under $d_{\mathcal{E}}(\cdot, \cdot)$. For example, $8 = (1000)_2$ and $7 = (0111)_2$ have L_1 distance 1 but $d_{\mathcal{E}}(8, 7) = 15$. Fortunately, such cases occur with a relatively low probability, and experimental results in Section IV show that bit-plane

randomization leads to only a slight reduction in retrieval accuracy, as a trade-off for security.

*2) **Random Projection**:* An alternative to treating the feature vector as separate bit-planes is to consider the vector as a whole and perform some random transformation that preserves the distance. Random projection accomplishes this goal based on the idea that close points in a high dimensional space will be mapped to close points in a low dimensional space with high probability. Due to this property, random projection has been used as a building block in locality sensitive hashing [23] for efficient search over large multimedia databases.

The idea of random projection is briefly described as follows. Given a feature vector $\mathbf{f} \in \mathbb{R}^n$, we generate a key-dependent Gaussian random matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ with independent standard Gaussian components. The encryption function is defined as

$$\mathcal{E}(\mathbf{f}) = \mathbf{R} \cdot \mathbf{f} = [\mathbf{r}_1 \cdot \mathbf{f}, \cdots, \mathbf{r}_m \cdot \mathbf{f}] \in \mathbb{R}^m, \tag{3}$$

where $\mathbf{r}_i \cdot \mathbf{f}$ is the dot product between the i^{th} row of $\mathbf{R}$ and $\mathbf{f}$. The distance preserving property of random projection can be derived by considering the L_1 distance of encrypted features, i.e., $d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g})) = \|\mathcal{E}(\mathbf{f}) - \mathcal{E}(\mathbf{g})\|_1$. Using equation (3), we have

$$\begin{aligned} d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g})) &= \sum_{i=1}^{m} |\mathbf{r}_i \cdot \mathbf{f} - \mathbf{r}_i \cdot \mathbf{g}| = \sum_{i=1}^{m} |\mathbf{r}_i \cdot (\mathbf{f} - \mathbf{g})| = \sum_{i=1}^{m} \|\mathbf{r}_i\|_2 \cdot \|\mathbf{f} - \mathbf{g}\|_2 \cdot |\cos(\theta_i)| \\ &= \|\mathbf{f} - \mathbf{g}\|_2 \cdot \sum_{i=1}^{m} \|\mathbf{r}_i\|_2 \cdot |\cos(\theta_i)| \approx c \cdot \|\mathbf{f} - \mathbf{g}\|_2 \end{aligned} \tag{4}$$

Here, θ_i is an independent and identically distributed random variable representing the angle between the vector $\mathbf{f} - \mathbf{g}$ and the random vector $\mathbf{r}_i$. By the law of large numbers, $\|\mathbf{r}_i\|_2 \approx$ const and $\sum_{i=1}^{m} |\cos(\theta_i)| \approx$ const. Thus, the distance $d_{\mathcal{E}}(\cdot, \cdot)$ between encrypted features is proportional to the L_2 distance between the original feature vectors with high probability. By increasing the dimension m of the projected feature vector, the error $|d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g})) - c \cdot \|\mathbf{f} - \mathbf{g}\|_2|$ can be made arbitrarily small, leading to better approximation of the original L_2 distance. The projection dimension m controls the trade-off between retrieval performance and storage, as will be shown by the experimental results in Section IV.

In image retrieval literature, L_1 norm is also widely used and is shown to achieve slightly superior performance over L_2 norm in retrieval based on color histogram [23]. Random projection can also be used to preserve the L_1 distance between the original feature vectors when the projection is performed on the square-root of the feature vector,

$$\mathcal{E}(\mathbf{f}) = \mathbf{R} \cdot \sqrt{\mathbf{f}}, \text{ where } \sqrt{\mathbf{f}} = \left[\sqrt{f_1}, \cdots, \sqrt{f_n}\right]. \tag{5}$$

To prove that the encryption in (5) preserves L_1 distance, we introduce the concept of unary encoding

of an integer vector. Given $\mathbf{f} = [f_1, \cdots, f_n]$, its unary encoding $\mathcal{U}(\mathbf{f})$ is defined as follows:

$$\mathcal{U}(\mathbf{f}) = [\mathcal{U}(f_1), \mathcal{U}(f_2), \cdots, \mathcal{U}(f_n)], \text{ where } \mathcal{U}(f_i) = \underbrace{11\cdots11}_{f_i}\underbrace{00\cdots00}_{M-f_i}. \tag{6}$$

Here M is the maximum possible value for any component of $\mathbf{f}$. Since the L_1 and L_2 norms for a binary vector are the same, we can perform random projection on $\mathcal{U}(\mathbf{f})$ so that

$$\|\mathbf{R}\cdot\mathcal{U}(\mathbf{f}) - \mathbf{R}\cdot\mathcal{U}(\mathbf{g})\|_1 \approx c\cdot\|\mathcal{U}(\mathbf{f}) - \mathcal{U}(\mathbf{g})\|_2 = c\cdot\|\mathbf{f}-\mathbf{g}\|_1. \tag{7}$$

Note that the projection of $\mathcal{U}(\mathbf{f})$ onto a vector of standard Gaussian random variables results in a Gaussian random variable with variance $\sum_{i=1}^{m} f_i$. This is equivalent to the projection of $\sqrt{\mathbf{f}}$ onto a vector of standard Gaussian random variables.

The security of random projection based scheme is due to the fact that without knowing the secret key and therefore the projection matrix $\mathbf{R}$, it is theoretically impossible to reconstruct the exact original features from the projected ones. For $m < n$, $\mathbf{y} = \mathbf{R}\cdot\mathbf{x} \in \mathbb{R}^m$ is an under-determined equation and there are infinitely many $\mathbf{x}$ that can give the same output $\mathbf{y}$. For $m \geq n$, the equation $\mathbf{y} = \mathbf{R}\cdot\mathbf{x}$ can be solved by using pseudo-inverse, but a different choice of $\mathbf{R}$ will give a different $\mathbf{x}$. Therefore, without knowing $\mathbf{R}$, it is theoretically impossible to obtain the exact $\mathbf{x}$ by knowing $\mathbf{y}$. We provide formal security proof in Section V.

3) ***Randomized Unary Encoding****:* Key-dependent random projection is an efficient algorithm for feature protection and preserves the distance between feature vectors with high probability. However, since the projection is a linear operation, using a reasonable amount of known plaintext features and their encrypted versions, an attacker can obtain the projection matrix. As will be shown by the security analysis in Section V, this poses security threats in the known plaintext attack model (KPA), where the attacker has access to a set of plaintext and ciphertext pairs. This motivates us to add an additional layer of security by introducing nonlinear operations into the feature encryption.

Given a feature vector $\mathbf{f} = [f_1, \cdots, f_n]$, we perform unary encoding $\mathcal{U}(\mathbf{f}) = [\mathcal{U}(f_1), \mathcal{U}(f_2), \cdots, \mathcal{U}(f_n)]$. The nonlinearity of the encryption is achieved by performing XOR of $\mathcal{U}(\mathbf{f})$ with a vector of binary random variables $\mathbf{r}$ and then randomly permuting the resulting binary vector. As discussed in Section III-A1, XOR and random permutation preserve the Hamming distance among $\mathcal{U}(\mathbf{f}), \forall\mathbf{f}$, which also equals the L_1 distance between original feature vectors. Denoting the encryption by XOR and permutation as $\mathcal{E}_1(\cdot)$, we have $\|\mathcal{E}_1(\mathcal{U}(\mathbf{f})) - \mathcal{E}_1(\mathcal{U}(\mathbf{g}))\|_2 = \|\mathbf{f}-\mathbf{g}\|_1$. By using efficient shuffling algorithms, $\mathcal{E}_1(\cdot)$ takes $O(nM)$ time, where M is the maximum possible value for any component of $\mathbf{f}$. One disadvantage of using unary encoding is the storage increase from $O(n\log M)$ bits to $O(nM)$ bits. To reduce storage, we further apply random projection on $\mathcal{E}_1(\mathcal{U}(\mathbf{f}))$, which also plays an important role in enhancing the

security of the scheme, as will be shown in Section IV.

Denote the encryption by XOR and permutation as $\mathcal{E}_1(\cdot)$ and random projection as $\mathcal{E}_2(\cdot)$. The overall encryption function $\mathcal{E}(\cdot)$ is now $\mathcal{E}_1(\cdot)$ followed by $\mathcal{E}_2(\cdot)$, and can be written as $\mathcal{E}(\mathbf{f}) = \mathcal{E}_2(\mathcal{E}_1(\mathcal{U}(\mathbf{f}))) \in \mathbb{R}^m$. Considering L_1 distance of encrypted features, we have the approximate distance preserving property

$$d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g})) \approx c \cdot \|\mathcal{E}_1(\mathcal{U}(\mathbf{f})) - \mathcal{E}_1(\mathcal{U}(\mathbf{g}))\|_2 = c \cdot \|\mathbf{f} - \mathbf{g}\|_1. \tag{8}$$

The randomized unary encoding scheme effectively preserves the L_1 distance of original feature vectors with high probability and provides enhanced security, as will be shown in Section V.

B. Secure Indexes

Once the image features are encrypted using the above methods, they can be stored onto the server and provide search capability in the encrypted domain without revealing information about the database content. However, since the image features are usually high dimensional vectors, comparing every pair of such vectors is computationally prohibitive for a large database. Modern image retrieval techniques often achieve efficiency and scalability through well-designed search indexes. In the following, we propose two secure indexing schemes, – secure inverted index and secure min-hash, which can retain the efficient search capability of the plaintext search indexes.

1) ***Secure Inverted Index****:* Inverted index [21] is a widely used indexing structure in text document retrieval, where each keyword has an associated inverted index listing the documents that contain this keyword and the number of occurrences of this word in each of these documents. Only those documents that appear in the query word's inverted index need to be considered during retrieval. By utilizing the visual words representation of images [19], inverted index can be constructed for image documents and facilitates efficient search and retrieval over large image databases.

As discussed in Section II, in order to protect the privacy of query image and minimize the amount of database information leaked to the server during the search process, inverted indexes should be generated and protected on the user side before being transferred to the server. In order to generate inverted index, a vocabulary tree is first created, where each node in the tree denotes a representative feature vector and each leaf node represents a visual word. Generating a vocabulary tree requires large set of training images and computationally intensive hierarchical clustering. Therefore, we assume that the vocabulary tree will be generated by the service provider, who usually has large computational resources, and is then provided to each user. To build secure search indexes from the vocabulary tree, the content owner extracts the visual features from each image, assigns each feature to the closest visual word in the vocabulary tree, and finally updates and encrypts the inverted indexes for those visual words. This procedure of index generation is illustrated in Figure 3.

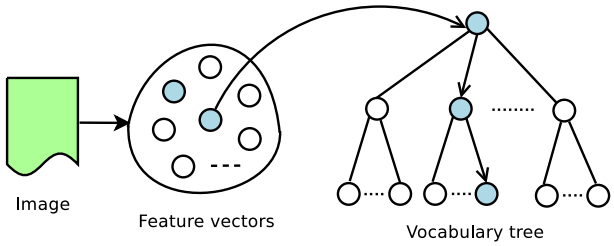

Fig. 3: Inverted index generation by content owner

Word ID	i			
Image ID	I_1	I_2	$\cdots$	I_{N_i}
Word frequency	w_1	w_2	$\cdots$	w_{N_i}

Fig. 4: Data structure of inverted index

Consider a total of N visual words and suppose N_i images contain the i^{th} visual word. Figure 4 shows the content of the inverted index of the i^{th} visual word, where w_j is the number of times the i^{th} word appears in image I_j. Given any query image Q and database image D, their bags of visual words are denoted as $\{Q_1, Q_2, \cdots, Q_N\}$ and $\{D_1, D_2, \cdots, D_N\}$, respectively. Here Q_i and D_i are the number of times the i^{th} word appears in the query and the database image, respectively. Normalization is typically applied so that $\sum_{i=1}^{N} Q_i = \sum_{i=1}^{N} D_i$ for all database images. After normalization, Q_i and D_i can take non-integer values and represent the relative frequency of occurrence of the i^{th} word. In the conventional non-secure setting, $\{D_1, D_2, \cdots, D_N\}$ is used to update the inverted indexes during index generation and $\{Q_1, Q_2, \cdots, Q_N\}$ is used to search the database for similar images.

***Encryption of Inverted Index**:* Given that the service provider typically creates and thus has the knowledge of the vocabulary tree, inverted indexes in their plaintext form can potentially reveal information about the visual content of the images. We protect the inverted index by first performing a random permutation $\tau(\cdot)$ on the word IDs so that the i^{th} word will now have an ID $\tau(i)$. Computing random permutation takes $O(N)$ time and needs to be done only once on the user side. However, the server needs to guess the correct IDs from $O(N!)$ possibilities, which is computationally infeasible given the typically large value of N.

Scrambling word IDs alone is not secure, because the server can still use visual word frequencies to identify the words that appear more frequently. An example is given in Figure 5, showing the distribution of word frequencies for local color histograms extracted from the Corel image dataset of 1000 images. This statistical information can be exploited to identify many words and makes the random permutation less secure. We apply order preserving encryption (OPE) [24] to alleviate this problem. OPE has the property that for two values x and y, if $x < y$, after encryption $\mathcal{E}(\cdot)$, the order relation is preserved so that $\mathcal{E}(x) < \mathcal{E}(y)$. By applying OPE on the word frequency values, we can make the distribution of encrypted frequency values close to a uniform distribution in order to reduce the amount of information leaked to the server. At the same time, the preservation of the order information ensures that image similarity can still be compared in the encrypted domain.

To perform order preserving encryption, we map each frequency value w to an integer uniformly selected from an interval $[l_w, u_w]$. The length of each such interval is chosen by the content owner to be proportional to the number of times that the value w occurs in all inverted indexes. Note that inverted index of a particular word only stores information about images that contain this word, therefore only positive w will be considered. Intervals for different word frequency values are non-overlapping and order preserving, i.e., for $w < v$, their corresponding intervals $[l_w, u_w]$ and $[l_v, u_v]$ satisfy $u_w < l_v$, while for $w = v$, they will be mapped to two values randomly chosen from the same interval $[l_w, u_w]$. These intervals form a partition of a large overall interval, and we use [0,7800] as the overall interval in our experiments. Figure 6 shows that after order preserving encryption, the distribution of word frequency values is closer to a uniform distribution over the large overall interval.

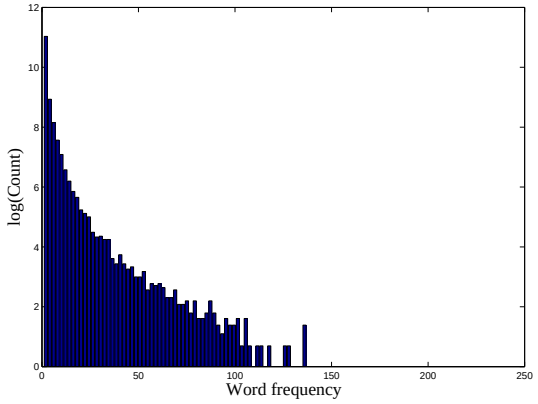

Fig. 5: Histogram of word frequencies before OPE

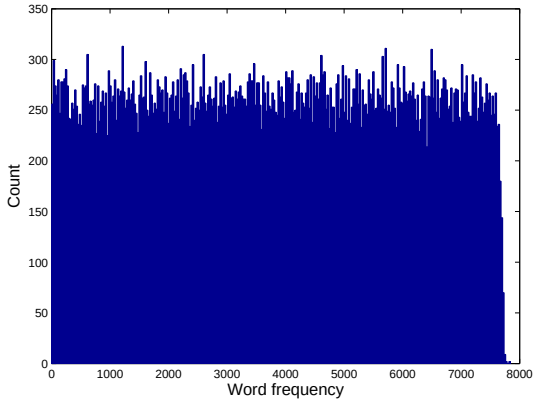

Fig. 6: Histogram of word frequencies after OPE

***Retrieval using Encrypted Index**:* After encryption, the visual words representations of the query image and an image in the database are denoted by $\{\mathcal{E}(Q_1), \cdots, \mathcal{E}(Q_N)\}$ and $\{\mathcal{E}(D_1), \cdots, \mathcal{E}(D_N)\}$, respectively, where $\mathcal{E}(\cdot)$ represents the order preserving encryption. Since visual words that are common in many images carry little discriminative information, we weigh the OPE encrypted version of each frequency value $\mathcal{E}(Q_i)$ and $\mathcal{E}(D_i)$ by its inverse document frequency (IDF) [25]. IDF is defined as IDF $= \log(\frac{M}{N_i})$, where M is the total number of images in the database and N_i is the number of images containing the word i. Commonly occurring visual words will have low IDF and receive small weights in similarity comparison. After encryption and weighting, we represent the query image and database image as

$$Q_{OPE} = \{\tilde{Q}_1, \tilde{Q}_2, \cdots, \tilde{Q}_N\}, \text{ where } \tilde{Q}_i = \mathcal{E}(Q_i) \log\left(\frac{M}{N_i}\right), \quad (9)$$

$$D_{OPE} = \{\tilde{D}_1, \tilde{D}_2, \cdots, \tilde{D}_N\}, \text{ where } \tilde{D}_i = \mathcal{E}(D_i) \log\left(\frac{M}{N_i}\right). \quad (10)$$

The similarity of two images Q_{OPE} and D_{OPE} after OPE is measured by the Jaccard similarity between $\{\tilde{Q}_1, \tilde{Q}_2, \cdots, \tilde{Q}_N\}$ and $\{\tilde{D}_1, \tilde{D}_2, \cdots, \tilde{D}_N\}$:

$$\text{Sim}(Q_{OPE}, D_{OPE}) \triangleq \frac{|Q_{OPE} \cap D_{OPE}|}{|Q_{OPE} \cup D_{OPE}|} \triangleq \frac{\sum_{i=1}^{N} \min(\tilde{Q}_i, \tilde{D}_i)}{\sum_{i=1}^{N} \max(\tilde{Q}_i, \tilde{D}_i)}. \tag{11}$$

The Jaccard similarity measures the similarity between two sets and has been used for near duplicate detection of text and image documents [26], [27]. The set operations $\cap$ and $\cup$ are extended in Equation (11) to measure the similarity between two sets of word frequency values. The functions $\min(\cdot,\cdot)$ and $\max(\cdot,\cdot)$ return the minimum and maximum value of the two input arguments, respectively. As the order information used in $\min(\cdot,\cdot)$ and $\max(\cdot,\cdot)$ is preserved by the order preserving encryption, the Jaccard similarity computed from the encrypted sets reflects the similarity of the plaintext sets, thus allowing similarity comparison in the encrypted domain.

As order preserving encryption preserves the order among encrypted word frequency values, some information about the encrypted index may be revealed, and we shall discuss the related security impact in Section V. The other limitation of using OPE on the inverted index is that the length of intervals used in OPE is determined by the distribution of word frequency values and this distribution can change when many more images are added to or deleted from the database. For example, if one word frequency value appears much more often in the newly added set of images, the corresponding OPE interval will have higher probability in the word frequency distribution than other intervals. Such a change in the distribution may reveal some of the interval ranges used in OPE and make OPE less secure. As the storage size of image indexes is typically much smaller than that of the images, this security problem with dynamic database changes can be alleviated by periodically downloading the indexes from the server to the user side, decrypting them, and encrypting them again using the new distribution information.

*2) **Secure Min-Hash Algorithm**:* The min-Hash algorithm, first proposed by Broder et al. [22], provides another efficient way to compute the Jaccard similarity between the visual words representations of two images. The min-Hash algorithm was originally developed for near duplicate detection of text documents [26]; extensions to near duplicate detection of images have been proposed recently by applying min-Hash to visual words representations [27], [28]. Here, we focus on the security aspect of the min-Hash algorithm and use it for secure ranking of image similarity.

The basic idea of the min-Hash algorithm is as follows: For any given set $\mathcal{A}$ such as the visual words representation, its min-Hash is defined as $m(\mathcal{A}, f) = \arg\min_{x \in \mathcal{A}} f(x)$, where f is a randomized hash function[1] with the property that $\Pr[f(x) < f(y)] = \Pr[f(x) > f(y)] = 0.5$, $\forall x, y \in \mathcal{A}$ and $x \neq y$. The

[1]The hash function used here is different from a cryptographically secure hash function in that it does not need to be strongly collision free.

probability that two sets have the same min-Hash value is given by their Jaccard similarity defined in Equation (11).

To compare the similarity between a given query image and an image in the database, we use their visual words representations:

$$Q_{MH} = \{\hat{Q}_1, \hat{Q}_2, \cdots, \hat{Q}_N\}, \text{ with } \hat{Q}_i = Q_i \log\left(\frac{M}{N_i}\right), \tag{12}$$

$$D_{MH} = \{\hat{D}_1, \hat{D}_2, \cdots, \hat{D}_N\}, \text{ with } \hat{D}_i = D_i \log\left(\frac{M}{N_i}\right), \tag{13}$$

where Q_i and D_i are the number of times the i^{th} visual word appears in the query and the database image, respectively. Non-zero components in Q_{MH} and D_{MH} suggest the existence of the corresponding visual word and represents the number of occurrence scaled by the inverse document frequency. In order to apply min-Hash to measure the Jaccard similarity between the sets Q_{MH} and D_{MH}, we follow the method of Chum et al. [28] and represent Q_{MH} and D_{MH} as the following sets:

$$\mathcal{A}(Q_{MH}) = \{X_1^1, \cdots, X_1^{\hat{Q}_1}, X_2^1, \cdots, X_2^{\hat{Q}_2}, \cdots, X_N^1, \cdots, X_N^{\hat{Q}_N}\}, \tag{14}$$

$$\mathcal{A}(D_{MH}) = \{X_1^1, \cdots, X_1^{\hat{D}_1}, X_2^1, \cdots, X_2^{\hat{D}_2}, \cdots, X_N^1, \cdots, X_N^{\hat{D}_N}\}. \tag{15}$$

Here, X_i^j is a unique element indexed by i and j. The min-Hash values generated from $\mathcal{A}(Q_{MH})$ and $\mathcal{A}(D_{MH})$ satisfy

$$\Pr[m(\mathcal{A}(Q_{MH}), f) = m(\mathcal{A}(D_{MH}), f)] = \text{Sim}(Q_{MH}, D_{MH}) = \frac{\sum_{i=1}^N \min(\hat{Q}_i, \hat{D}_i)}{\sum_{i=1}^N \max(\hat{Q}_i, \hat{D}_i)}. \tag{16}$$

In order to obtain a reliable estimate of $\text{Sim}(Q_{MH}, D_{MH})$, k independent hash functions $f_1, f_2, \cdots, f_k$ are used to generate k min-Hash values for $\mathcal{A}(Q_{MH})$ and $\mathcal{A}(D_{MH})$, respectively. The concatenation of the k min-Hash values for $\mathcal{A}(Q_{MH})$ forms a *sketch* of the image Q_{MH}, and a sketch of the image D_{MH} is formed similarly. The number of identical values in their sketches, denoted by $s(Q_{MH}, D_{MH}) = |\{i : 1 \leq i \leq k | m_i(Q_{MH}) = m_i(D_{MH})\}|$, follows a binomial distribution

$$\Pr[s(Q_{MH}, D_{MH}) = l] = \binom{k}{l} [\text{Sim}(Q_{MH}, D_{MH})]^l [1 - \text{Sim}(Q_{MH}, D_{MH})]^{k-l}.$$

Thus, the maximum likelihood estimate for the similarity of two images $\text{Sim}(Q_{MH}, D_{MH})$ is the fraction of identical values in their sketches, $s(Q_{MH}, D_{MH})/k$.

The use of randomized hash functions in the min-Hash algorithm makes it possible to protect the original word frequency information from the adversary. To implement the randomized hash function, we use the input value as part of the seed to a pseudo random number generator and map the input value to a random number between $(0, 1)$ as the output. The min-Hash $m(\mathcal{A}, f) = X_{i_0}^{j_0} = \text{argmin}_{i,j} f(X_i^j)\ \forall X_i^j \in \mathcal{A}$.

In our implementation, $X_{i_0}^{j_0}$ is the output of a trapdoor function $g(i_0, j_0)$ uniquely determined by i_0 and j_0 so that it is easy to compute in one direction to obtain $g(i_0, j_0)$ given i_0 and j_0, but it is computationally difficult to compute in the opposite direction, i.e., to determine i_0 and j_0 given $g(i_0, j_0)$. The trapdoor function can be implemented through a trapdoor permutation function [29].

During index generation, the content owner creates min-Hash sketch for every image using a secret key and stores these sketches on the remote server. During retrieval, the query image is processed similarly by the content owner, who has the secret key to generate its min-Hash sketch. This sketch is then sent to the server for comparison with the sketches of the database images. Similarity between two images is computed as the percentage of identical values in their min-Hash sketches. Retrieval efficiency can be further improved by organizing similar sketches into the same bucket of another hash table [30] and comparing only to sketches with similarity higher than a certain threshold.

IV. Experimental Results

Two desirable properties of a secure image retrieval scheme are good retrieval performance that is comparable to state-of-the-art plaintext retrieval schemes and provable security so that content privacy is protected against adversaries. In this section, we demonstrate the retrieval performance of the proposed secure search schemes, and in the next section, we analyze the security of these schemes under different attack models.

A. Experiment Setup

We perform search and retrieval experiments on an image database containing 1000 color images from the Corel dataset [31]. These images are grouped by content into 10 categories, with 100 images in each category: African, Beach, Architecture, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountain, and Food. Image sizes are either 256×384 or 384×256. This database has been used as ground-truth for evaluating color image retrieval [32] and image annotation [33]. Sample images from the database are shown in Figure 7.

We use global color histogram for the feature encryption based schemes and localized color histogram for the index encryption based schemes. The color histograms are in the HSV color space. For localized color histogram, we divide an image into 256 blocks and extract a 128-dimensional color histogram from each block by quantizing the three channels of hue, saturation, and intensity value into 8, 4, and 4 levels, respectively, where finer quantization is allocated to hue as suggested by Jeong et al. [32]. For feature encryption based schemes, we have one histogram for each image, while for index encryption based schemes, we obtain a training set of 256,000 histograms from the entire database and perform hierarchical clustering to build the vocabulary tree. During clustering, we use L_1 norm to measure the

Fig. 7: Selected content of the Corel dataset (figure from [32])

distance between color histograms and take the average of each cluster as its representative feature. Each node in the vocabulary tree except the leaf nodes has 10 children and the tree has height 3, which gives 10^3 visual words.

During search and retrieval, images in the database are returned in the descending order of their similarity to the query. Retrieval performance is evaluated using precision-recall curves, where precision and recall are defined as

$$\begin{aligned} precision &= \frac{\text{\# of positive images among returned images}}{\text{\# of returned images}}, \\ recall &= \frac{\text{\# of positive images among returned images}}{\text{\# of positive images in the database}}. \end{aligned}$$

A higher precision value at a given recall value indicates better retrieval performance. Our experiments use every image in the database as a query, and positive images are those images in the same category as the query.

For comparison with prior art on color histogram based image retrieval, we choose Jeong et al.'s work [32], where different settings for image retrieval using color histograms are compared and the best retrieval performance is achieved by comparing image similarity using the intersection of global color histograms in the HSV space. Given two color histograms H_1 and H_2 in the d-dimensional space, their intersection $I(H_1, H_2)$ is defined as

$$I(H_1, H_2) = \frac{\sum_{i=1}^{d} \min[H_1(i), H_2(i)]}{\min[\sum_{i=1}^{d} H_1(i), \sum_{i=1}^{d} H_2(i)]}.$$

Images with higher intersection values are considered more similar. During retrieval, the color histogram

of the query image is compared with every histogram in the database and images with higher similarity are returned. When the L_1 norms of the histograms are the same, retrieval based on histogram intersection is equivalent to retrieval by L_1 distance of the histograms.

B. Retrieval results based on encrypted features

In contrast to the conventional retrieval scheme that uses plaintext color histograms as features for similarity comparison, we use the encrypted versions of the same features in the secure retrieval scheme. Recall that the distance defined in Section III-A1 between features after bit-plane randomization is an upper bound on the original L_1 distance between features, while random projection and randomized unary encoding preserves the original L_1 distance with high probability. Thus, we would expect our secure retrieval schemes based on encrypted features to have performance comparable to conventional schemes. The retrieval performance of the three feature protection schemes are illustrated in Fig. 8.

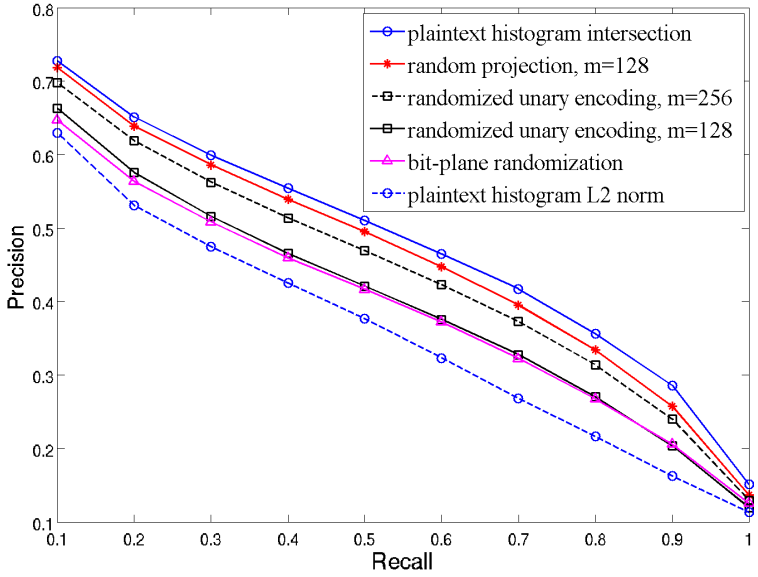

Fig. 8: Retrieval performance based on encrypted features

For comparison, we show the retrieval performance using histogram intersection and L_2 distance between plaintext histograms as the top and bottom curves in Fig. 8, respectively. For each of the precision-recall curves in Fig. 8, a higher precision value at a given recall value indicates better retrieval performance. We see that the retrieval performance based on encrypted features is better than plaintext histogram based on L_2 norm and is close to plaintext histogram intersection. This is expected because the original L_1 distance between color histograms is approximately preserved. By searching over encrypted features, we only need to retrieve about $1\% - 9\%$ more images to obtain the same number of relevant

images as in plaintext search. This shows that secure retrieval can be achieved by slightly trading off retrieval accuracy.

Among the three feature protection schemes, we observe a trade-off among retrieval performance, storage, and computational complexity. Bit-plane randomization has the largest gap to the plaintext intersection method among the three schemes. This is because the distance between features after bit-plane randomization is an upper bound to the original L_1 distance and there is some discrepancy between the distances for certain cases, as discussed in Section III. However, bit-plane randomization has the lowest complexity $O(kn)$ as compared to $O(mn)$ of random projection and $O(mnM)$ of randomized unary encoding, where k is the number of bit-planes to encrypt, m is the dimension of the projected features, and M is the largest value of the feature vector. Random projection and randomized unary encoding preserve L_1 norm with high probability, so their performance can be made arbitrarily close to plaintext scheme by increasing the projection dimension m. By doubling the projection dimension m from 128 to 256, the gap between the curves of plaintext and randomized unary encoding can be reduced by half, and the performance of random projection can be made almost the same as plaintext search (random projection with $m = 256$ is not shown in the figure). With the same m, random projection outperforms the randomized unary encoding because the latter projects the much longer unary encoded version than the original feature. M in randomized unary encoding can be quantized to a much smaller value to reduce complexity. In this paper, we quantize M from 98304 to 128 with no loss in retrieval performance. The higher complexity of randomized unary encoding is a trade-off for better security, which will be analyzed in Section V. Compared to traditional non-secure retrieval scheme, the additional step in our schemes is to encrypt the features using pseudo random permutations or random projection, which are computationally efficient and take less than 1 second per image on a dual-core 3.0GHz PC with 4GB RAM in our experiments.

C. Retrieval results based on secure indexes

The two secure indexing schemes are based on the visual words representation of the image. To establish the baseline retrieval performance of the visual words representation, we first demonstrate retrieval using the inverted index without any encryption and compare that with the performance of plaintext histogram intersection. In visual words representation, local color histograms are extracted from blocks of the image. By utilizing the vocabulary tree, each image is then represented as a bag of visual words $Q = \{\hat{Q}_1, \cdots, \hat{Q}_N\}$. Here, $\hat{Q}_i$ takes the form $\hat{Q}_i = Q_i \log\left(\frac{M}{N_i}\right)$, as shown in equations (12) and (13), where Q_i is the term frequency value and $\log\left(\frac{M}{N_i}\right)$ is the inverse document frequency weighting. In Fig. 9, we can see that plaintext histogram intersection and inverted index with term frequency-inverse document frequency (TF-IDF) weighting achieve very similar performance.

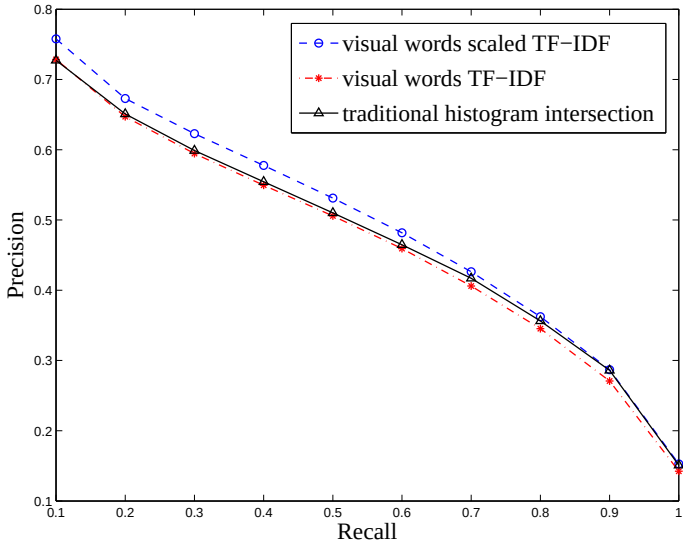

Fig. 9: Baseline retrieval performance of visual words representation

Considering that w occurrences of a word may not necessarily carry w times the significance of a single occurrence, we apply the following scaled TF-IDF weighting,

$$\hat{Q}_i = \begin{cases} (1 + \log(Q_i)) \log(M/N_i), & \text{if } Q_i \neq 0, \\ 0, & \text{if } Q_i = 0, \end{cases} \tag{17}$$

and find that the inverted index using visual words representation outperforms the histogram intersection by about 3% in precision. The comparison in Figure 9 shows that visual words representation can be used for rank-ordered retrieval of color images, while its success for object recognition using SIFT [18] features has been reported in [19], [20].

In the secure indexing scheme based on inverted index, the inverted indexes are encrypted by order preserving encryption and random permutation of word IDs. We perform the same retrieval experiment using encrypted inverted indexes and compare in Figure 10 its precision-recall curve with that of the baseline inverted index without any encryption. We can see that encryption of the index has very little impact on the retrieval performance, and the precision-recall curves before OPE and after OPE are very close to each other. This can be attributed to the use of Jaccard similarity, which is approximately preserved after the order preserving encryption. Compared to the conventional non-secure setting, generating encrypted indexes imposes additional computational cost on the content owner, but this cost is small. When performed on a dual-core 3.0GHz PC with 4GB RAM, the tasks of extracting features, creating visual words representation, and encrypting inverted indexes can be done within 2 seconds per image, and search and retrieval over the entire database of 1000 images takes less than 1 second. The use of

inverted index ensures that retrieval can be efficiently scaled to larger databases.

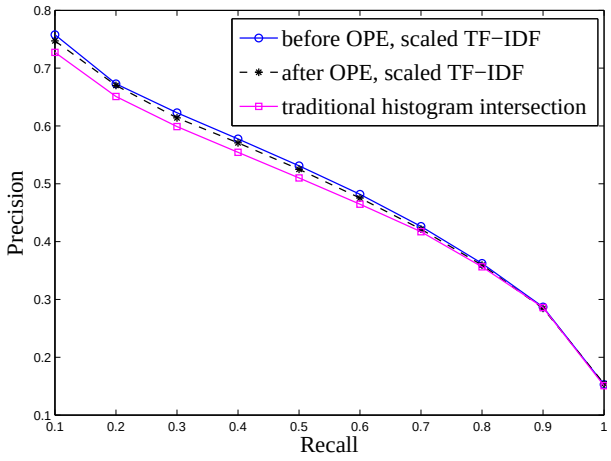

Fig. 10: Retrieval performance of OPE

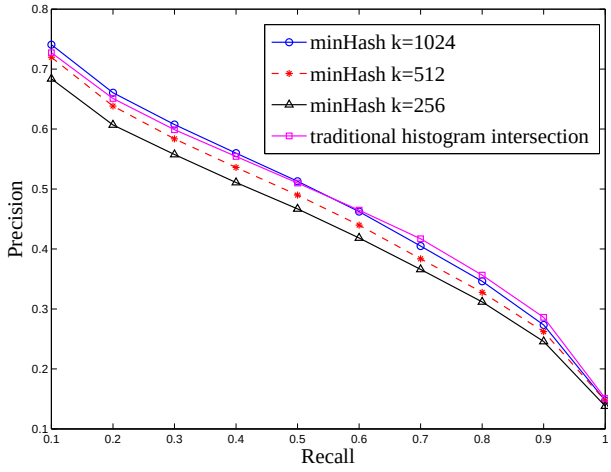

Fig. 11: Retrieval performance of min-Hash

In the secure indexing scheme based on the min-Hash algorithm, each image is represented by a sketch $\{m_1, m_2, \cdots, m_k\}$, where m_i is the min-Hash value generated by the i^{th} randomized hash function. Images are returned in the descending order of their similarity to the query, measured by the percentage of identical values between their min-Hash sketches. Retrieval performance using min-Hash sketches is shown in Figure 11, where we can see that using min-Hash sketches gives a retrieval performance comparable to those of the histogram intersection method and the inverted index scheme. This is expected because the number of identical values in two min-Hash sketches preserves the Jaccard similarity with high probability. As the length of the sketch k increases, the estimate for image similarity based on the percentage of identical values in two min-Hash sketches becomes more accurate, leading to better precision-recall curves. A sketch length of 1024 gives performances similar to that of the inverted index scheme. Min-Hash sketches can be computed very efficiently on the user side, taking less than 1 second per image on a Dual-Core 3.0GHz PC with 4GB RAM. During retrieval, we compare sketches of all the images in the database in order to obtain the precision-recall curve. In practice, typically only the most similar images are of interest, so additional hash tables can be constructed for those sketches to further improve retrieval efficiency.

V. Security Analysis

One of the key features of modern information retrieval is the separation of data from search indexes, which enables efficient search capability and convenient data management. Similarly, in the proposed secure retrieval schemes, image data themselves are encrypted separately from search indexes by cryptographic ciphers before being stored on the server. Built on top of established cryptographic primitives, it

is computationally difficult for an adversary to infer image content by breaking the cryptographic ciphers. Therefore, the security of the entire retrieval schemes is determined by the security of the encryption schemes for features or indexes. In this section, we focus our analysis on the security of the proposed feature/index encryption schemes and potential information leakage under two attack models, namely, ciphertext only attack (COA) and known-plaintext attack (KPA). We assume that the adversary is semi-honest, i.e., the adversary will follow the execution requirement of the protocol but may use what they see during the execution to infer additional information. Such a semi-honest model is applicable to such scenarios as third-party service providers. We provide security definition and proofs for the proposed secure retrieval schemes under the COA model and compare their security levels in the KPA model.

A. Ciphertext Only Attack (COA)

Ciphertext only attack model assumes an adversary has access only to the ciphertext. Since the goal of confidentiality preserving multimedia retrieval is to protect content privacy against adversaries, such as malicious intruders and untrustworthy service providers who have access to encrypted images and indexes, it is necessary for any confidentiality preserving retrieval scheme to be secure under the COA model. We first discuss the potential information leakage and define the security objective in the COA model.

Suppose M images $\{I_1, \cdots, I_M\}$ are stored in the database in their encrypted forms $\{\tilde{I}_i\}, i = 1, \cdots, M$, we denote the corresponding encrypted index or encrypted features as $\mathcal{E}(I_i)$. Trying to identify the database content, the adversary can compare $\mathcal{E}(I_i)$ for all i and divide the database into several groups such that images in each group have similar encrypted indexes or features and are likely to have similar content. As will be shown subsequently through a rigorous security analysis, if we assume for the proposed schemes that the adversary has no prior knowledge about the database content and that the features and indexes are properly encrypted, the adversary will not be able to infer the plaintext image content using just the grouping information. During retrieval, the encrypted index or feature $\mathcal{E}(Q)$ of the query image is sent to the server and the encrypted versions of images similar to the query image are returned. The adversary can thus obtain information about the retrieval history in terms of which images are retrieved each time, but no content information will be revealed.

A secure retrieval scheme should be able to prevent the adversary from learning the following: (1) the plaintext versions of the database images $\{I_i\}$ and the query image Q; (2) the secret key K used in the encryption; and (3) any function of the images $f(I_i)$ and $f(Q)$, such as the plaintext features of the images. We provide a security definition formulated using the concept of content indistinguishability, which is adapted from the indistinguishability definition [34] widely used in cryptanalysis. We show that most of the proposed schemes are provably secure under this definition.

We begin with a review of the concept of negligible function. It is used in security analysis to characterize the probability that a computationally-bounded adversary successfully breaks a computationally secure encryption scheme [29]. A function $f(\cdot)$ is *negligible* if for every polynomial $p(\cdot)$, there exists an N such that for all integers $n > N$ it holds that $f(n) < 1/p(n)$. According to modern cryptography [29], events that occur with negligible probability are unlikely to occur and they can be ignored for all practical purposes. In security analysis, the parameter n typically determines the length of the secret key and the security of the encryption scheme. With an increase in n, the probability that an adversary successfully breaks the encryption scheme decays faster than the inverse of any polynomial function of n.

We now provide the definition of content indistinguishability for characterizing the security of the proposed schemes.

Content indistinguishability: *Consider the following security experiment: an adversary chooses two images I_0 and I_1. The content owner chooses a secret key K_0 and encrypts the two images using a feature or index encryption scheme $\mathcal{E}$ to obtain the encrypted indexes $\mathcal{E}(I_0, K_0)$ and $\mathcal{E}(I_1, K_0)$. The content owner then randomly chooses a value b from $\{0, 1\}$ with equal probability and sends the encrypted index $\mathcal{E}(I_b, K_0)$ to the adversary. Using any probabilistic polynomial time algorithm, the adversary outputs a number b' as an estimate of b. The retrieval scheme satisfies the property of* ***content indistinguishability*** *if*

$$|\Pr(b' = b) - \frac{1}{2}| \leq negl(n),$$

for every choice of I_0, I_1 by the adversary, where $negl(n)$ is a negligible function of the security parameter n. The probability is taken over all possible random choices by the adversary and in the experiment, such as the secret key K_0 and the value of b.

The content indistinguishability definition essentially states that a computationally bounded adversary cannot distinguish between two encrypted features or indexes even if he/she has knowledge of the plaintext features or indexes. Under the COA model where the attacker is assumed to have knowledge of the ciphertext only, if a feature/index encryption scheme satisfies the above definition, the adversary will not be able to distinguish any two encrypted images in terms of their content, and thus confidentiality is preserved under such a COA model.

To prove that the proposed encryption schemes satisfy the above security definition, we carry out reduction to relate the security of the entire scheme to the security of some basic cryptographic building blocks, such as pseudorandom functions and pseudorandom permutations [29]. Since these cryptographic primitives are considered hard to break by any probabilistic polynomial time algorithm, we can prove that an encryption scheme satisfies the above security definition by showing that breaking the scheme is equivalent to breaking the cryptographic primitives. Next, we outline the security proof for the feature

protection scheme based on random projection under the security definition of content indistinguishability. The proof for other schemes can be carried out in a similar way.

A1. Proof for random projection and randomized unary encoding: We analyze the security of feature protection schemes based on random projection and randomized unary encoding together, both of which have projection onto random vectors as the last step. Given any two images I_0 and I_1, we denote their corresponding feature vectors as $\mathbf{f}_0$ and $\mathbf{f}_1$, which are normalized so that $\|\mathbf{f}_0\|_2 = \|\mathbf{f}_1\|_2 = c$ for some constant c. For the random projection based scheme, the content owner will choose a secret key K_0 and generate a pseudorandom matrix $\mathbf{R}_0$ whose elements are independent Gaussian variables from $\mathcal{N}(0,1)$. Encryption by random projection is denoted as

$$\tilde{\mathbf{f}_0} \triangleq \mathcal{E}(\mathbf{f}_0, K_0) = \mathbf{R}_0 \cdot \mathbf{f}_0 = (\mathbf{r_1} \cdot \mathbf{f}_0, \mathbf{r_2} \cdot \mathbf{f}_0, \cdots, \mathbf{r_m} \cdot \mathbf{f}_0),$$
$$\tilde{\mathbf{f}_1} \triangleq \mathcal{E}(\mathbf{f}_1, K_0) = \mathbf{R}_0 \cdot \mathbf{f}_1 = (\mathbf{r_1} \cdot \mathbf{f}_1, \mathbf{r_2} \cdot \mathbf{f}_1, \cdots, \mathbf{r_m} \cdot \mathbf{f}_1),$$

where $\mathbf{r}_i$ is the i^{th} row of the matrix $\mathbf{R}_0$.

Assuming now that the matrix $\mathbf{R}_0$ is truly random with each element as independent standard Gaussian variables, the encrypted features $\tilde{\mathbf{f}}_0$, $\tilde{\mathbf{f}}_1$ can be considered as vectors chosen uniformly at random from the distribution of vectors whose components are independent Gaussian variables $\mathcal{N}(0, c^2)$, because $\mathbf{r_i} \cdot \mathbf{f}_b$ are independent Gaussian $\mathcal{N}(0, c^2)$ for any $i \in \{1, \cdots, m\}$ and $b \in \{0, 1\}$. The conditional probability of the encrypted feature given the plaintext feature only depends on the value of c. Since all plaintext features are normalized to have the same value of c, we have $\Pr[\tilde{\mathbf{f}}_b|\mathbf{f}_b] = \Pr[\tilde{\mathbf{f}}_b]$, for any $\mathbf{f}_b$, which satisfies the definition of perfect secrecy [35]. A cryptosystem satisfies perfect secrecy if the posterior probability of the ciphertext given the plaintext is exactly the same as the prior probability of the ciphertext, for all ciphertexts and plaintexts. Therefore, the probability that any probabilistic polynomial time algorithm can distinguish $\tilde{\mathbf{f}}_0$ and $\tilde{\mathbf{f}}_1$ is exactly 1/2.

After replacing the truly random projection matrix $\mathbf{R}_0$ with a cryptographically secure pseudorandom matrix, we denote the probability that $\tilde{\mathbf{f}}_0$ and $\tilde{\mathbf{f}}_1$ are distinguished by a probabilistic polynomial time attacker as $\Pr(b' = b) = \frac{1}{2} + \epsilon(n)$. If the function $\epsilon(n)$ is not negligibly small, it would imply that there exists a polynomial time algorithm that can distinguish a truly random sequence from a pseudorandom sequence, which contradicts the definition of cryptographically secure pseudorandom sequence of numbers [29]. Thus $\epsilon(n)$ must be negligibly small implying that feature protection schemes based on random projection or on randomized unary encoding satisfy the definition of content indistinguishability. ■

Although the security analysis of the random projection scheme and the randomized unary encoding scheme are the same under the COA model, we will show in Section V-B that the additional encryption stage in the randomized unary encoding scheme makes it more secure than the random projection scheme

in the KPA model. Proof for feature protection scheme based on bit-plane randomization and index encryption using secure min-hash can be carried out in a similar fashion, by showing that the distribution of the encrypted features does not depend on the plaintext features.

A2. Security discussion for inverted index based scheme: The above security analysis can be applied to most of the proposed secure retrieval schemes, and the only scheme that does not satisfy the definition of content indistinguishability is the inverted index based scheme. In this subsection, we provide security discussion for the secure inverted index scheme and discuss the implications for practical applications.

As described in Section III-B1, the inverted index based scheme encrypts the inverted index by pseudorandom permutation and order preserving encryption. Although the order statistics are scrambled by pseudorandom permutation, the variance of the index may still be approximately preserved by order preserving encryption. Therefore, in the security experiment of content indistinguishability, an adversary can choose two images whose bag-of-words representations have very different variance and such variance can be preserved after encryption, allowing the adversary to distinguish the two images. Since order preserving encryption (OPE) reveals some information about the plaintext in terms of its variance, the retrieval scheme based on OPE does not satisfy the content indistinguishability definition. A proper security definition for OPE with clear practical implications would be desirable but is a challenging task. Only recently was a formal definition for OPE security being provided by Boldyreva et al. [36], in which a construction that can satisfy this definition was also given. It is still not clear what kind of information will be leaked by such a secure OPE construction in a practical scenario. Despite the security limitations, order preserving encryption can enable efficient indexing and range query in the encrypted domain, which would otherwise be much more difficult. We refer readers to [36] for a detailed security analysis of OPE and recommend that OPE based scheme should be used with caution in practice.

A3. Retrieval performance using a wrong key: To demonstrate the security of our secure retrieval schemes under the COA model, we perform attacks by first extracting and encrypting the features or indexes from some plaintext images using a randomly chosen key, and then searching the database using these encrypted features as a query and analyzing the retrieved encrypted images. For a secure retrieval scheme, the query index encrypted by a wrong key is equally like to be closest to any encrypted index in the database. Therefore, retrieval from an encrypted database using a wrong key is equivalent to picking images randomly from the database. For verification, we perform retrieval using every image in the database as the query but encrypt the query index or features with a key different from the one used in encryption. The precision-recall curves for all the schemes are shown in Fig. 12, Fig. 13, and Fig. 14.

Since the database has 100 images in each of the 10 categories, random selection from the database would imply a precision value around 0.1 for all recall values, which is confirmed in the above figures. Although the server can search the database using any plaintext images, it cannot learn anything about the

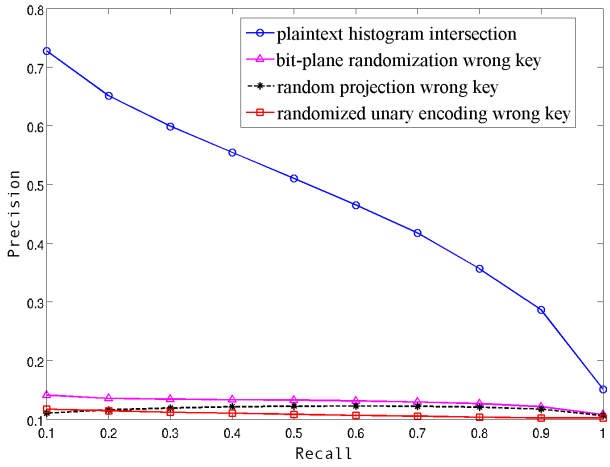

Fig. 12: Retrieval using a wrong key for feature protection schemes

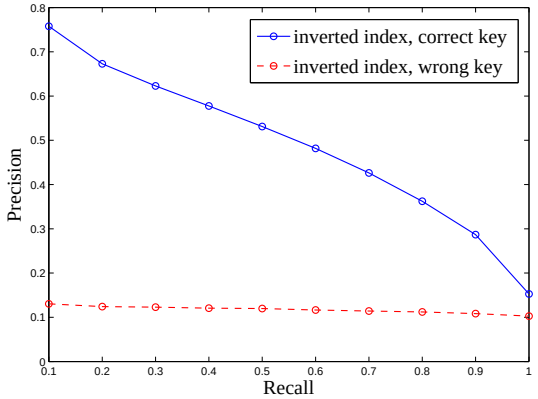

Fig. 13: Inverted index scheme using wrong key

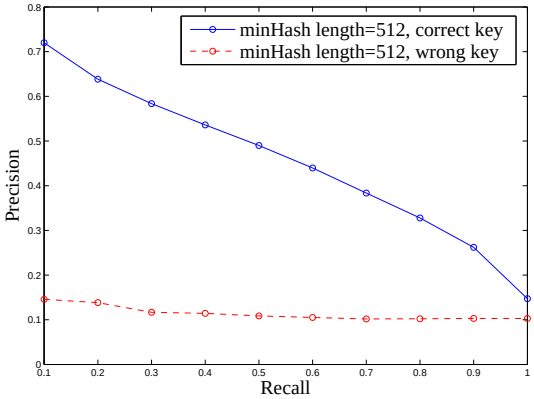

Fig. 14: Min-Hash scheme using wrong key

image content from the ranking of the returned images, as the returned images are essentially a random selection from the database.

B. Known Plaintext Attack (KPA)

In the KPA model, the attacker is assumed to know some plaintext images I and their corresponding encrypted features or indexes $\mathcal{E}(I, K_0)$, where K_0 is the secret key used by the content owner during encryption and is unknown to the attacker. In the semi-honest adversary model, the known-plaintext attack is a less serious concern as compared to the ciphertext-only attack. Hence, we provide an informal security analysis for the proposed schemes under the KPA model, as a supplement to our formal security analysis in the COA model.

One immediate consequence of the information available to the attacker under the KPA model is

that the attacker can search the database using the known $\mathcal{E}(I, K_0)$, and the top images returned from the retrieval would have similar visual appearance to the known image I if the distance between their encrypted features is small. For example, the adversary can search the database using a known image of sky with the corresponding encrypted features or indexes and identify how many images in the database might be visually similar to the sky image based on the retrieval results. If the attacker has access to an increasing number of plaintext images with diverse content and their encrypted features, more information about the database content would be revealed.

To evaluate the security of our proposed retrieval schemes under the KPA model, we compare the number of plaintext-ciphertext pairs required for an attacker to recover the secret key used to encrypt features or indexes. To illustrate the trade-off between security and computational complexity, we provide analysis for the two feature protection schemes based on random projection and unary encoding. We shall see that more encryption stages can bring higher security at a higher computational cost. Security analysis for the other schemes can be carried out in a similar fashion.

B1. Analysis for random projection: Assume that the attacker knows k pairs of plaintext features $\mathbf{f}_i$ and their encrypted versions $\tilde{\mathbf{f}}_i$, $i = 1, 2, \cdots, k$. We can write $\tilde{\mathbf{F}} = \mathbf{R} \cdot \mathbf{F}$, where $\mathbf{F}$ and $\tilde{\mathbf{F}}$ have $\mathbf{f}_i$ and $\tilde{\mathbf{f}}_i$ as their i^{th} columns, respectively. The key-dependent projection matrix $\mathbf{R}$ can be uniquely determined if $\mathbf{F}$ is invertible. To make $\mathbf{F}$ invertible, the attacker needs at least n linearly independent plaintext features. Overall, to break the random projection scheme, the attacker needs $O(n)$ pairs of plaintext and ciphertext. Similar analysis has been used for evaluating the security of different image hashing schemes [37]. ■

B2. Analysis for randomized unary encoding: In randomized unary encoding, the feature vector of the image $\mathbf{f} = \{f_1, \cdots, f_n\}$ is first converted by unary encoding to a binary string $\mathcal{U}(\mathbf{f}) = [\mathcal{U}(f_1), \cdots, \mathcal{U}(f_n)]$, whose length is nM where M is the maximum possible feature value. This binary string $\mathcal{U}(\mathbf{f})$ is XORed with a random binary string, then randomly permuted, and finally projected to give the encrypted feature $\tilde{\mathbf{f}} = \mathbf{R} \cdot \mathcal{E}(\mathcal{U}(\mathbf{f}))$, where $\mathcal{E}(\cdot)$ here represents the random XOR and permutation.

If the attacker knows the intermediate stage $\mathcal{E}(\mathcal{U}(\mathbf{f}))$, then $O(nM)$ pairs of $\{\mathbf{f}, \mathcal{E}(\mathcal{U}(\mathbf{f}))\}$ are required to deduce the XOR and permutation pattern, and similarly, $O(nM)$ pairs of $\{\mathcal{E}(\mathcal{U}(\mathbf{f})), \tilde{\mathbf{f}}\}$ are required to obtain the random projection matrix $\mathbf{R}$. However, due to the concatenation of the two encryption stages, the randomized unary encoding achieves higher security than the single encryption of random projection. A brief discussion of the security increase due to multiple encryptions is provided below. Using similar ideas in differential cryptanalysis [38], in the best case for the attacker, if the known feature vectors satisfy some special property, for example, $\mathcal{U}(\mathbf{f}_1)$ and $\mathcal{U}(\mathbf{f}_2)$ differ only in 1 bit, $\mathcal{E}(\mathcal{U}(\mathbf{f}_1))$ and $\mathcal{E}(\mathcal{U}(\mathbf{f}_2))$ obtained after XOR and random permutation will have only one different component, so the attacker can obtain individual columns of the projection matrix $\mathbf{R}$. However, due to the pseudorandom permutation, given all the columns of $\mathbf{R}$, there are $O((nM)!)$ possibilities for the matrix $\mathbf{R}$. Furthermore, the system

$\tilde{\mathbf{f}} = \mathbf{R} \cdot \mathcal{E}(\mathcal{U}(\mathbf{f}))$ is typically underdetermined. Thus knowing $\mathbf{R}$ will not help the attacker uniquely determine $\mathcal{E}(\mathcal{U}(\mathbf{f}))$. Because the projection matrix $\mathbf{R}$ is typically non-invertible, the concatenation of the two encryption stages here is resilient to the meet-in-the-middle attack which undermines the security increase of double encryption using DES from a squared key space to only a doubled key space [39]. A formal analysis on the security increase from the multiple encryption is desirable but is outside the scope of the current paper. ■

In summary, randomized unary encoding exhibits much better security than the simple random projection scheme. The higher security of the randomized unary encoding comes from the use of two encryption stages, which reflects the trade-off between security and computational complexity. Similar arguments also hold for the secure min-hash scheme and secure inverted index scheme, where the two-step encryption in secure min-hash scheme provides better security than one-step encryption schemes such as random projection and secure inverted index. The chosen plaintext attack (CPA) model, where the attacker can obtain encrypted features for any chosen plaintext image, will be a severe attack for the proposed secure retrieval schemes, because the attacker in the CPA model can essentially query the database using any plaintext images and infer the database content based on the retrieval results. Based on the security requirement and computational constraints, the proposed secure retrieval schemes should be properly selected for use in different applications.

VI. Conclusions

This work is among the first in addressing the problem of content based multimedia retrieval over encrypted databases, which has become increasingly important given the current trend of storing and managing personal data collections online under the emerging cloud computing paradigm. We have proposed two complementary approaches for secure multimedia retrieval: one is to encrypt visual features of multimedia documents and allow similarity comparison of the features in their encrypted forms; and the other is to encrypt state-of-the-art search indexes without affecting their search capability. Scalability and efficiency of the search indexes are retained after the encryption. We have shown through experiments that retrieval performance comparable to plaintext retrieval can be achieved. We also introduced definitions of the security objective in the multimedia retrieval scenario and provided security proofs under the ciphertext only attack model. Security analysis demonstrates different trade-offs among computational complexity, search performance, and security for the proposed retrieval schemes. Future work will aim at achieving an improved trade-off between security and computational complexity. Exploring general multimedia processing such as format conversion and content manipulation in the encrypted domain can further support secure online multimedia management.

REFERENCES

[1] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," in *SPIE/IS&T Media Forensics and Security*. Proc. of SPIE, vol. 7254, 2009, pp. 7254–18.

[2] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection," in *IEEE Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 1533–1536.

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

[4] R. Brinkman, J. M. Doumen, and W. Jonker, "Using secret sharing for searching in encrypted data," in *Workshop on Secure Data Management in a Connected World*, 2004, pp. 18–27.

[5] D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano, "Public-key encryption with keyword search," in *Proceedings of Eurocrypt*, 2004.

[6] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M. Wu, and D. W. Oard, "Confidentiality preserving rank-ordered search," in *Proceedings of the ACM Workshop on Storage, Security, and Survivability*, 2007, pp. 7–12.

[7] Z. Erkin, A. Piva, S. Katzenbeisser, R. L. Lagendijk, J. Shokrollahi, G. Neven, and M. Barni, "Protection and retrieval of encrypted multimedia content: when cryptography meets signal processing," *EURASIP Journal on Information Security*, vol. 7, no. 2, pp. 1–20, 2007.

[8] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: ideas, influences, and trends of the new age," *ACM Computing Surveys*, 2008.

[9] J. Shashank, P. Kowshik, K. Srinathan, and C. Jawahar, "Private content based image retrieval," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[10] M.-L. Yiu, G. Ghinita, C. S. Jensen, and P. Kalnis, "Outsourcing search services on private spatial data," in *Proceedings of the 2009 IEEE International Conference on Data Engineering*, 2009, pp. 1140–1143.

[11] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proceedings of the 35th SIGMOD International Conference on Management of Data*, 2009, pp. 139–152.

[12] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," *Privacy Preserving Technologies, LNCS*, vol. 5672, pp. 235–253, 2009.

[13] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *12th International Conference on Information Security and Cryptology*, 2009.

[14] W. Jiang, M. Murugesan, C. Clifton, and L. Si, "Similar document detection with limited information disclosure," in *IEEE 24th International Conference on Data Engineering*, 2008.

[15] Y. Mao and M. Wu, "A joint signal processing and cryptographic approach to multimedia encryption," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 2061– 2075, 2006.

[16] M. Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 905–917, 2006.

[17] H. Kim, J. Wen, and J. D. Villasenor, "Secure arithmetic coding," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2263–2272, 2007.

[18] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[19] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[21] J. Zobel and A. Moffat, "Inverted files versus signature files for text indexing," *ACM Transactions on Database Systems*, vol. 23, no. 4, pp. 453–490, 1998.

[22] A. Broder, M. Charikar, A. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," in *Proceedings of the 30th ACM Symposium on Theory of Computing*, 1998, pp. 327–336.

[23] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the International Conference on Very Large Data Bases*, 1999.

[24] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proc. SIGMOD*, 2004.

[25] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[26] A. Broder, "On the resemblance and containment of documents," in *Proceedings of Compression and Complexity of Sequences*, 1997, pp. 21–29.

[27] O. Chum, J. Philbin, M. Isard, and A. Zisserman, "Scalable near identical image and shot detection," in *Proceedings of the International Conference on Image and Video Retrieval (CIVR)*, 2007.

[28] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-hash and TF-IDF weighting," in *British Machine Vision Conference (BMVC)*, 2008.

[29] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*. Chapman & Hall/CRC, 2007.

[30] M. Slaney and M. Casey, "Locality-sensitive hashing for finding nearest neighbors," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 128–131, 2008.

[31] Corel test set. [Online]. Available: http://wang.ist.psu.edu/∼jwang/test1.tar

[32] S. Jeong, C. Won, and R. Gray, "Image retrieval using color histograms generated by Gauss mixture vector quantization," *Computer Vision and Image Understanding*, vol. 94, pp. 44–66, 2004.

[33] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 394–410, 2007.

[34] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, pp. 270–299, 1984.

[35] C. E. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal*, vol. 28, pp. 656–719, 1945.

[36] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order preserving symmetric encryption," in *Advances in Cryptology - EUROCRYPT*, 2009, pp. 224–241.

[37] Y. Mao and M. Wu, "Unicity distance of robust image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 462–467, 2007.

[38] D. Coppersmith, "The data encryption standard (DES) and its strength against attacks," *IBM Journal of Research and Development*, vol. 38, May 1994.

[39] W. Diffie and M. E. Hellman, "Exhaustive cryptanalysis of the NBS data encryption standard," *Computer*, vol. 10, pp. 74–84, June 1977.