Enabling Search over Encrypted Multimedia Databases

Wenjun Lu, Ashwin Swaminathan, Avinash L. Varna, and Min Wu Department of Electrical and Computer Engineering

and Institute for Advanced Computer Studies

University of Maryland, College Park, U.S.A

email: {wenjunlu, ashwins, varna, minwu}@eng.umd.edu

ABSTRACT

Performing information retrieval tasks while preserving data confidentiality is a desirable capability when a database is stored on a server maintained by a third-party service provider. This paper addresses the problem of enabling content-based retrieval over encrypted multimedia databases. Search indexes, along with multimedia documents, are first encrypted by the content owner and then stored onto the server. Through jointly applying cryptographic techniques, such as order preserving encryption and randomized hash functions, with image processing and information retrieval techniques, secure indexing schemes are designed to provide both privacy protection and rank-ordered search capability. Retrieval results on an encrypted color image database and security analysis of the secure indexing schemes under different attack models show that data confidentiality can be preserved while retaining very good retrieval performance. This work has promising applications in secure multimedia management.

Keywords: image retrieval, secure search, visual words, min-hash

1. INTRODUCTION

The goal of information retrieval over an encrypted database is to provide efficient and accurate search capability over encrypted documents without decrypting them first. Advancements in this area can have applications in protecting the privacy of sensitive data stored on third-party servers. Examples include webmail and online backup services, where there are growing concerns that the service provider may not be entrusted with the content of personal data (such as emails, photos, and videos), and data encryption is needed to protect the privacy of these documents from the service provider and potential hackers. Another example is the biometric database where it is desirable to encrypt the biometric records in order to prevent any unauthorized access and identity theft, but it is also critical that after encryption we can still quickly match a query to similar records in the database.

Prior work on information retrieval in the encrypted domain focused on text documents. Song et al. [1], Brinman et al. [2], and Boneh et al. [3] explored Boolean search to identify whether or not a query term is present in an encrypted text document. Recent work by Swaminathan et al. [4] 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 search techniques can be applied to keyword based search of multimedia data. 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, if can be done, provides more flexibilities, whereby sample images, audios or videos are presented as query and documents with similar audio-visual content in the database are identified.

An emerging area of related work to secure multimedia retrieval is secure signal processing, which aims at performing normal signal processing tasks but keeping the signals being processed secret. Erkin et al. [5] 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 [6]. Cryptographic primitives themselves 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. [7] 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 properly formulating the query message and response message during multiple rounds of communications between the user and the server, the server is made oblivious to the actual search path and thus unaware of the query content.

Compared with the work of Shashank et al. [7], this paper focuses on content-based multimedia retrieval over encrypted databases, where both the query and database documents are encrypted and their privacy is protected. 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. We demonstrate the proposed techniques using images 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 two secure indexing schemes built upon visual words representation [8] of images. The first scheme makes use of inverted indexes of visual words and the second scheme exploits randomized hash functions. Both indexing schemes achieve efficient image retrieval and are scalable for large databases. We jointly exploit cryptography, image processing, and information retrieval techniques to ensure that the encrypted search indexes can preserve the search capability. Our experiments on an encrypted color image database show that data confidentiality can be preserved while retaining good retrieval performance. 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 applications in secure online services for multimedia management.

The paper is organized as follows: Section 2 describes the system model of secure retrieval scenario and the properties of a desirable indexing scheme. Section 3 presents the two proposed secure indexing schemes, namely, secure inverted index and secure min-Hash schemes. Section 4 summarizes experimental results on retrieval over a color image database and provides security analysis of the two schemes under different attack models. Conclusions are drawn in Section 5.

2. SYSTEM MODEL

As discussed in the introduction, 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 directly by treating images as ordinary data, or using image specific techniques such as selective and format-compliant encryption [9], [10], [11] to enable post-processing such as transcoding on encrypted images. Built upon the established cryptographic encryption tools, it is computationally difficult for the server to decrypt image files in order to learn the database content.

Encryption keeps data content safe from the server but also makes it difficult for the server to build searchable indexes. In secure retrieval scenario, the search indexes need to be generated and properly encrypted by software tools on the user side using a secret key and then transferred to the server. A system model for this scenario is shown by the left and the center dash-dotted blocks in Figure 1.

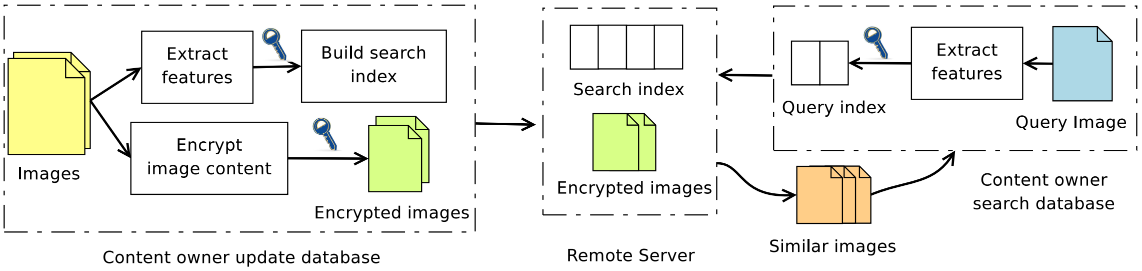

Figure 1: System model for secure image retrieval

A simple method to perform content-based search is to store feature vectors of each image in the database directly, and compare the feature vector of the query image and that of each image in the database to determine which images in the database are similar to the query. This naive search scheme is both inefficient and insecure for two reasons. Firstly, visual features, such as color histograms and salient points, are usually high dimensional vectors, so comparing every pair of such vectors is computationally prohibitive for a large database. Secondly, image features in cleartext may reveal information about image content. For example, a color histogram with large values for the blue components would indicate the likely presence of sky or ocean, and SIFT descriptors [12] may reveal information about some objects in the image.

A desirable indexing scheme for secure image retrieval, in addition to being efficient and scalable, should retain the similarity between image pairs and be properly secured using a secret key. Therefore, without knowing the key, it should be difficult to search the database or infer information about the database content. On the other hand, the content owner who knows the secret key can generate a properly encrypted query index from the query image using the secret key. The server then compares the query index with the stored indexes and returns the encrypted files of the most similar images to the user for decryption and viewing. The encrypted query index also helps protect the privacy of the query image. The retrieval stage is shown by the center and the right dash-dotted blocks in Figure 1.

An efficient way of representing images and potentially enabling fast and scalable search is by the visual words representation [8]. Feature vectors are first extracted and hierarchically clustered into a vocabulary tree, and each image is then indexed based on this vocabulary tree and represented as a bag of visual words. This bag of visual words describes how many times the representative feature vectors in the vocabulary tree occur in the image of question, which is analogous to term frequencies in text retrieval and thus allows for extending the state-of-the-art text search techniques to images. Experiments on object recognition in the recent literature [8,13] show that visual words based representation can be scaled to large databases. In the next section, we present two secure indexing schemes based on visual words representation, with the goal of preserving both retrieval efficiency and data confidentiality.

3. SECURE INDEXING SCHEMES

In this section, we consider two representative indexing schemes, namely, inverted index and min-Hash, and develop techniques to secure these indexes and use them for secure image search and retrieval.

3.1 Secure Inverted Index

Inverted Index for Visual Words: Inverted index [14] 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, and 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, inverted index can be extended to image documents and facilitates search and retrieval over large image databases.

As discussed in Section 2, in order to protect the privacy of query image and minimize the database information leaked to the server during search, the inverted indexes need to be generated and protected on the user side before being transferred to the server. The service provider usually has a large collection of training images and the resources to perform hierarchical clustering and create the vocabulary tree. Each node in the vocabulary tree denotes a representative feature vector, and each leaf node represents a visual word. A content owner stores a copy of the vocabulary tree. To build search index, the content owner extracts the visual features from each image, assigns each feature to the closest visual words in the vocabulary tree, and updates the inverted indexes for those visual words. This procedure of index generation is illustrated in Figure 2.

Consider a total of N visual words and N_i images containing the i^{th} visual word. Figure 3 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. 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.

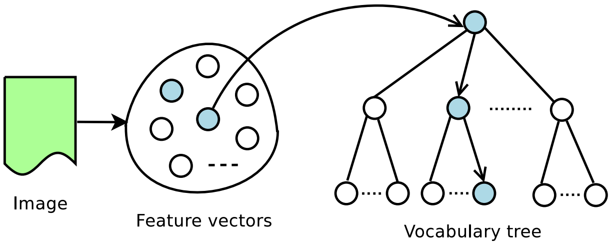

Figure 2: Generation of inverted index by the content owner

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

Figure 3: Content of inverted index

Inverted Index Encryption: Since the vocabulary tree is created by and thus known to the service provider, proper encryption of the inverted indexes generated by the content owner is needed. Otherwise, the server can look up the visual words present in each image from the word IDs, and thus infer the image content. 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 enough, because the server can still use visual word frequencies to identify the words that appear more frequently. An example is given in Figure 4, showing the distribution of word frequencies for local color histograms extracted from a 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) [15] to alleviate this problem. OPE has the property that for two values x and y, if $x < y$, then after encryption $\mathcal{E}(\cdot)$, the order relation is preserved so that $\mathcal{E}(x) < \mathcal{E}(y)$. By applying OPE on the word frequencies, we can make the distribution of encrypted frequency values close to a uniform distribution 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 this interval is proportional to the number of times that the value w occurs in all inverted indexes, and intervals of different values are non-overlapping and order preserving. These intervals form a partition of a larger predefined range, and we use [0,7800] in our experiments. Figure 5 shows the distribution of word frequencies after OPE, which is close to uniform.

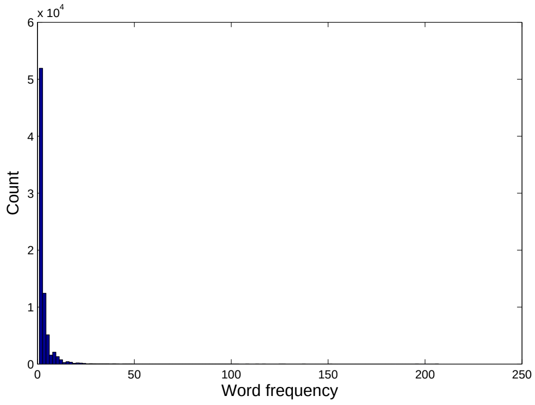

Figure 4: Histogram of word frequencies before OPE

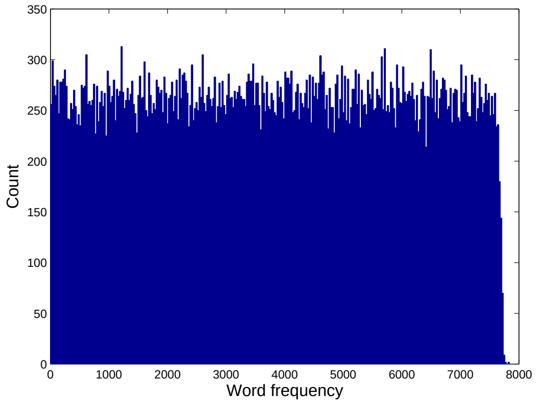

Figure 5: Histogram of word frequencies after OPE

Retrieval using Encrypted Index: After encryption, the visual words representations of the query image and database image are denoted by $\{\mathcal{E}(Q_1), \mathcal{E}(Q_2), \cdots, \mathcal{E}(Q_N)\}$ and $\{\mathcal{E}(D_1), \mathcal{E}(D_2), \cdots, \mathcal{E}(D_N)\}$, respectively, where $\mathcal{E}(\cdot)$ represents the order preserving encryption. To take into consideration the different amount of information carried by various visual words, 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) [16]. 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. Visual words that are common in many images carry little discriminative information, so they are assigned small weights in similarity comparison as their IDF is low. After encryption and weighting, we can represent the query image and database image as

$$Q_{OPE} = \{\tilde{Q}_1, \tilde{Q}_2, \cdots, \tilde{Q}_N\}, \text{ where } \tilde{Q}_i = \mathcal{E}(Q_i)\log\left(\frac{M}{N_i}\right), \tag{1}$$

$$D_{OPE} = \{\tilde{D}_1, \tilde{D}_2, \cdots, \tilde{D}_N\}, \text{ where } \tilde{D}_i = \mathcal{E}(D_i)\log\left(\frac{M}{N_i}\right). \tag{2}$$

The similarity of two images Q_{OPE} and D_{OPE} after OPE can be measured by the Jaccard similarity between $\{\mathcal{E}(Q_1), \mathcal{E}(Q_2), \cdots, \mathcal{E}(Q_N)\}$ and $\{\mathcal{E}(D_1), \mathcal{E}(D_2), \cdots, \mathcal{E}(D_N)\}$:

$$\text{Sim}(Q_{OPE}, D_{OPE}) = \frac{|Q_{OPE} \cap D_{OPE}|}{|Q_{OPE} \cup D_{OPE}|} = \frac{\sum_{i=1}^{N} \min(\tilde{Q}_i, \tilde{D}_i)}{\sum_{i=1}^{N} \max(\tilde{Q}_i, \tilde{D}_i)}. \tag{3}$$

The Jaccard similarity is a statistic measuring similarity between two sample sets and has been used for near duplicate detection of text and image documents [17, 18]. Because the order information used in the $\min(\cdot, \cdot)$ and $\max(\cdot, \cdot)$ functions is preserved by the order preserving encryption, the Jaccard similarity computed from the encrypted sets reflects the similarity of the sets' cleartext version. As such, similarity comparison can be done on the server in the encrypted domain and the encrypted files of the most similar images are returned to the user. These images are then decrypted on the user side and viewed or processed by the user.

One security limitation of using OPE on the inverted index is that the interval length of OPE is determined by the word frequency distribution 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 due to image addition, its OPE interval will have higher probability in the word frequency distribution than other intervals. Such change in the distribution will reveal the interval ranges used in OPE and make OPE less secure. As the storage size of the image indexes is typically much smaller than that of the images, this security problem with dynamic database applications 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.

3.2 Secure Min-Hash Algorithm

The algorithm of min-wise independent permutation, known as *min-Hash* [19], provides another efficient way to compare the Jaccard similarity between the visual words representations of two images. The min-Hash algorithm was originally developed to detect near duplicated copies of text documents [17]; extensions to near duplicate detection of images have been proposed recently by applying min-Hash to the visual words representation [18] [20]. Here, we focus on the security aspect of the min-Hash algorithm and examine its performance for secure ranking of image similarity.

The basic idea of the min-Hash algorithm is as follows: For any given set $\mathcal{A}$ (e.g. $\mathcal{A}$ can be 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 with the property that $\Pr[f(x) < f(y)] = \Pr[f(x) > f(y)] = 0.5$ for $x, y \in \mathcal{A}$ and $x \neq y$. The probability that two sets have the same min-Hash value is given by their Jaccard similarity defined in equation (3) above, i.e.

$$\Pr[m(\mathcal{A}_1, f) = m(\mathcal{A}_2, f)] = \text{Sim}(\mathcal{A}_1, \mathcal{A}_2) = \frac{|\mathcal{A}_1 \cap \mathcal{A}_2|}{|\mathcal{A}_1 \cup \mathcal{A}_2|}. \tag{4}$$

The set intersection in equation (4) counts the existence of common elements in $\mathcal{A}_1$ and $\mathcal{A}_2$. For visual words representation, we have both the existence and the frequency information. That is, for a given query image and target image, their visual words representations are:

$$Q_{MH} = \{\hat{Q}_1, \hat{Q}_2, \cdots, \hat{Q}_N\}, \text{ with } \hat{Q}_i = Q_i \log\left(\frac{M}{N_i}\right), \tag{5}$$

$$D_{MH} = \{\hat{D}_1, \hat{D}_2, \cdots, \hat{D}_N\}, \text{ with } \hat{D}_i = D_i \log\left(\frac{M}{N_i}\right), \tag{6}$$

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. A non-zero value in Q_{MH} and D_{MH} suggests the existence of the corresponding visual word and represents the number of occurrence scaled by the inverse document frequency. In order to extend Jaccard similarity to the sets Q_{MH} and D_{MH}, we follow the method by Chum et al. [20] and represent Q_{MH} and D_{MH} as the following sets:

$$\begin{aligned}\mathcal{A}(Q_{MH}) &= \{X_1^1, \cdots, X_1^{\hat{Q}_1}, X_2^1, \cdots, X_2^{\hat{Q}_2}, \cdots, X_N^1, \cdots, X_N^{\hat{Q}_N}\}, \qquad (7)\\ \mathcal{A}(D_{MH}) &= \{X_1^1, \cdots, X_1^{\hat{D}_1}, X_2^1, \cdots, X_2^{\hat{D}_2}, \cdots, X_N^1, \cdots, X_N^{\hat{D}_N}\}. \qquad (8)\end{aligned}$$

Here, X_i^j is a unique number indexed by i and j so that $X_i^j = X_s^l$ only if $i = s$ and $j = l$. The min-Hash values generated for $\mathcal{A}(Q_{MH})$ and $\mathcal{A}(D_{MH})$ satisfy probabilistically

$$\Pr[m(\mathcal{A}(Q_{MH}), f) = m(\mathcal{A}(D_{MH}), f)] = \mathrm{Sim}(Q_{MH}, D_{MH}) = \frac{|\mathcal{A}(Q_{MH}) \cap \mathcal{A}(D_{MH})|}{|\mathcal{A}(Q_{MH}) \cup \mathcal{A}(D_{MH})|} = \frac{\sum_{i=1}^N \min(\hat{Q}_i, \hat{D}_i)}{\sum_{i=1}^N \max(\hat{Q}_i, \hat{D}_i)}. \quad (9)$$

In order to obtain a reliable estimate of $\mathrm{Sim}(Q_{MH}, D_{MH})$, we use k independent randomized hash functions $f_1, f_2, \cdots, f_k$ 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

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

Thus, the maximum likelihood estimate for the similarity of two images $\mathrm{Sim}(Q_{MH}, D_{MH})$ 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 withhold the original word frequency information from the server. The hash function can be implemented via a cryptographically secure random number generator. A hash function $f_i(\cdot)$ maps each element in the set $\mathcal{A}(\cdot)$ to a random number between $(0, 1)$, and the min-Hash of $\mathcal{A}(\cdot)$ returns an element X_s^l for some s and l so that X_s^l is mapped to the smallest value under f_i. X_s^l is represented by a trapdoor function $g(s, l)$ uniquely determined by s and l so that it is easy to compute in one direction to obtain $g(s, l)$ given s and l, but it is computationally difficult to compute in the opposite direction, i.e. to determine s and l given $g(\cdot, \cdot)$ and $g(s, l)$.

During index generation, the content owner creates min-Hash sketch for every image using a secret key and store 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 and the server will compare the query sketch to the sketches of database images. Similarity between two images is computed as the percentage of identical values in their min-Hash sketches. In order to efficiently determine the most similar images, we can further organize similar sketches into the same slot of a hash table [21] and compare only to sketches with similarity higher than a certain threshold.

4. EXPERIMENTS AND ANALYSIS

4.1 Experimental Results

Experiment Setup: We perform search and retrieval experiments on an image database containing 1000 color images from the Corel dataset [22]. 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 [23] and image annotation [24]. Sample images from the database are shown in Figure 6.

The features used to generate the vocabulary tree in this experiment are localized color histograms in the color space of Hue, Saturation, and Value (HSV). 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,

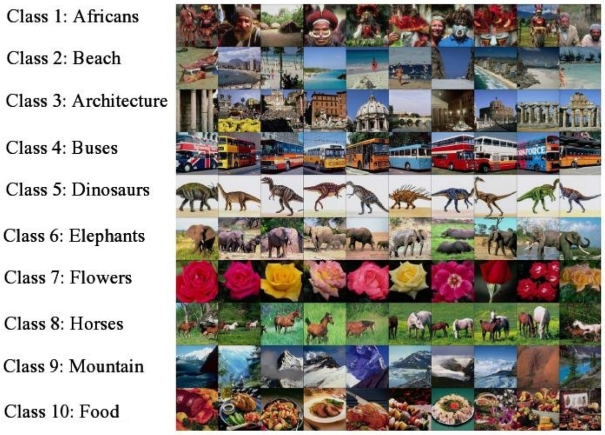

Figure 6: Selected content of the Corel dataset

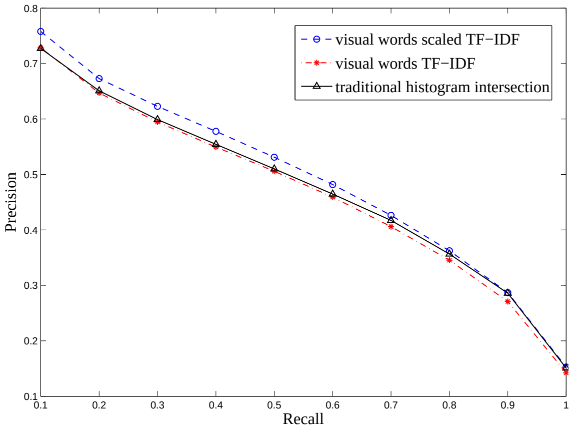

Figure 7: Baseline retrieval performance of visual words representation

4, and 4 levels, respectively, where finer quantization is allocated to hue as suggested by Jeong et al. [23]. 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 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 $10^3 = 1000$ visual words.

During search and retrieval, images in the database will be returned in a descending order of their similarity to the query, as computed using either the inverted index or the min-Hash sketches. 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.

Baseline Performance: To establish the baseline retrieval performance of the visual words representation, we first demonstrate retrieval using the inverted index without any encryption. We compare with the work by Jeong et al. [23], 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. Using the visual words representation and inverted index, we compare the set of localized color histograms between images using the Jaccard similarity. Taking every image in the database as query and evaluating the precision value at fixed recall values, we obtain the average precision-recall curve for both the histogram intersection method and the baseline inverted index method, as shown in Figure 7. We can see that histogram intersection and inverted index with term frequency-inverse document frequency (TF-IDF) weighting achieve very similar performance. Here, every element $\hat{Q}_i$ in the bag of words representation takes the form $\hat{Q}_i = Q_i \log\left(\frac{M}{N_i}\right)$, as shown in equations (5) and (6), where Q_i is the term frequency value and $\log\left(\frac{M}{N_i}\right)$ is the inverse document frequency weighting. 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,

$$\hat{Q}_i = \begin{cases} (1+\log(Q_i))\log(M/N_i), & \text{if } Q_i \neq 0, \\ 0, & \text{if } Q_i = 0, \end{cases} \tag{10}$$

and find that the inverted index using visual words representation outperforms the histogram intersection by around 3% in precision. The comparison in Figure 7 shows that visual words representation can be used for rank-ordered retrieval of color images, while its success for object recognition using SIFT [12] features have also been reported [8, 13].

Secure Retrieval Performance: 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 on the encrypted inverted indexes and compare in Figure 8 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 negative 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 take less than 1 second.

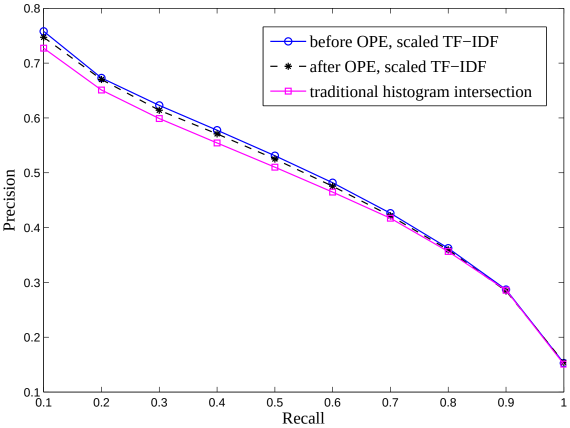

Figure 8: Retrieval performance of OPE

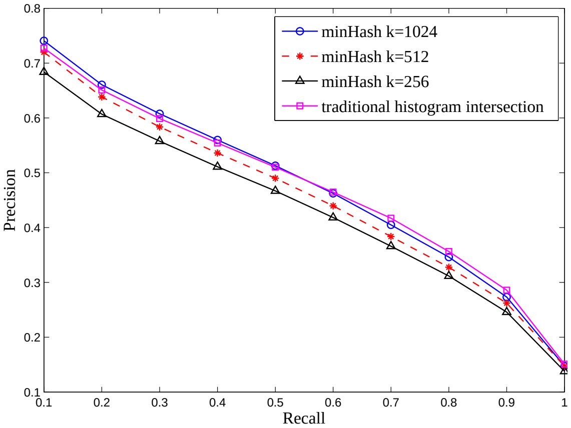

Figure 9: Retrieval performance of min-Hash

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 9.

From Figure 9, 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. With increasing length of the sketch, the estimate for image similarity using 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. During retrieval, we compare sketches of all the images in the database in order to obtain the precision-recall curve. In practice, only the most similar images are usually of interest, so hash tables can be constructed for those sketches to further improve efficiency of retrieval.

4.2 Security Analysis

As discussed in Section 2, in order to ensure data confidentiality, images are encrypted by cryptographic ciphers before storing on the server. Built on top of established cryptographic primitives, it can be assumed computationally difficult for the server to decrypt the images without knowing the secret key. Our analysis focuses on what information the server can learn from the encrypted search indexes (i.e. encrypted inverted index and min-Hash sketches) and the security of the two indexing schemes under different attack models.

Ciphertext Only Attack (COA): In this model, the server has access to the encrypted indexes but does not know anything more. As described in Section 3, the encrypted inverted index contains randomly permuted word IDs so that the server cannot tell which feature vectors are present in each image. The word frequency values are also encrypted to follow an approximately uniform distribution, so that it is difficult for the server to deduce the word IDs from the statistical information of the word frequencies. For indexing using min-Hash sketches, only similarity information is retained in the sketches while the exact values of word frequencies are obscured. Since each sketch contains outputs of cryptographically strong trapdoor functions, it is also computationally difficult for the server to infer the original word frequency information.

As our encrypted indexes allow for similarity comparison in order to enable rank-ordered retrieval, one possible attack from the server is to compare all database images and obtain their similarity information. The similarity information can also be learned by the server during retrieval because images similar to the query will eventually be returned to the user. However, given that both the query and database images are encrypted, the similarity information alone does not help the server to infer image content.

Another possible attack is for the server to create some synthetic images containing a few selected visual words and use these images to search the database. Without knowing the secret key used in building the search index, the server cannot generate a proper query index. For inverted index, without knowing the random permutation order, the word IDs of the query are randomly mapped to some word IDs of the database image and retrieved images are essentially a random selection from the database. For min-Hash sketches, the server needs to guess the key used in the randomized hash functions. Denote the correct key by K_1 and the guessed key by K_2. Assuming that the key space is sufficiently large and the key is randomly selected, we have $\Pr(K_1 = K_2)$ close to 0. For the same image Q, the sketch generated by the correct key and wrong key are denoted by $\{m_1^{(K_1)}, m_2^{(K_1)}, \cdots, m_k^{(K_1)}\}$ and $\{m_1^{(K_2)}, m_2^{(K_2)}, \cdots, m_k^{(K_2)}\}$, respectively, with $m_i^{(K_\alpha)} = g(\arg\min_{\hat{Q}_j \in Q} f_{K_\alpha}(\hat{Q}_j))$, where $g(\cdot)$ is the trapdoor function and $f_{K_\alpha}(\cdot)$ is the hash function with key K_α for $\alpha \in \{0, 1\}$. Under two different keys, $f_{K_1}(\hat{Q}_j)$ and $f_{K_2}(\hat{Q}_j)$ are independent, so we have

$$\Pr[\arg\min_{\hat{Q}_j \in Q} f_{K_1}(\hat{Q}_j) = \arg\min_{\hat{Q}_j \in Q} f_{K_2}(\hat{Q}_j)] = \frac{1}{N}, \tag{11}$$

where N is the total number of visual words in the database. Equation (11) holds true for any two images Q and D. This implies that the sketch generated by a wrong key will have approximately the same similarity values of k/N to all sketches in the database, so retrieved images will again be a random selection from the database.

To verify that retrieval using a query index generated by a wrong key returns images randomly selected from the database, we perform retrieval using every image in the database as the query but encrypt the query index with a key different from the one used in index generation. Precision-recall curves for the inverted index and min-Hash sketch approaches are shown in Figure 10 and Figure 11, respectively.

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 these two figures. Although the server can search the database using its own images, it cannot learn anything about the image content from the ranking of returned images, which are actually a random selection from the database.

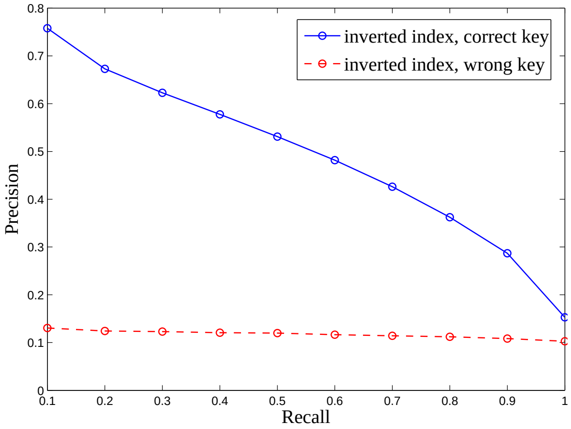

Figure 10: Inverted index retrieval using wrong key

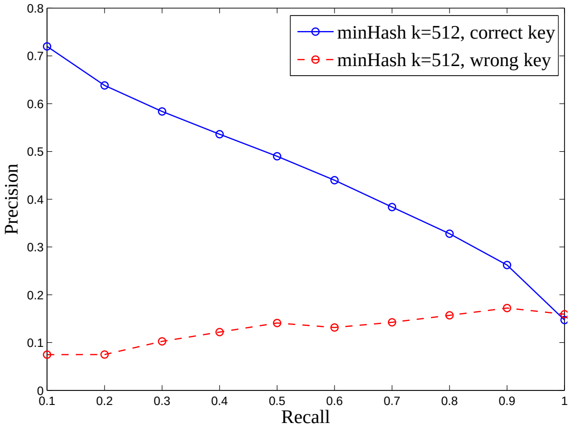

Figure 11: Min-Hash retrieval using wrong key

Known Plaintext Attack (KPA): In this attack model, the server knows some pairs of plaintext images and their encrypted indexes. For inverted index, this means that the server can generate the visual words representation for the known images and compare with the encrypted indexes. Using the order of word frequency values, the server can figure out part of the permutation order, and with several pairs of known images, the entire permutation order can be revealed. With the permutation order known, the server can easily obtain the interval ranges used in OPE and then search the database using specific images to learn more information about the database content. Therefore, it is important for the user to keep their database images secret from the server when inverted index is used.

For min-Hash sketches, the information flow from the word frequency to the final sketch is one way, as shown below:

$$Q = \{\hat{Q}_1, \cdots, \hat{Q}_N\} \xrightarrow[generator]{random\ number} f_i(\hat{Q}_j) \rightarrow X_i = \arg\min_{\hat{Q}_j} f_i(\hat{Q}_j) \xrightarrow{trapdoor} g(X_i) = m_i(Q).$$

Due to the use of trapdoor functions, knowing plaintext images does not help the server to obtain useful information on the random hash function $f_i(\cdot)$. So in KPA model, the server can know which encrypted image files in the database may have similar content to those known images, but the server cannot search the database using any other images. The combination of several cryptographic stages helps improve the security of min-Hash indexing scheme. As a probabilistic method, min-Hash scheme requires longer sketches to achieve better performance. In order to achieve performance similar to that of the inverted index scheme, sketch of length 1024 is needed in our experiment. This would impose slightly more storage requirement than the inverted index scheme.

If the server can choose any image and obtain its encrypted sketches, which is the Chosen Plaintext Attack (CPA) model, the min-Hash scheme becomes insecure. This is because in the CPA model, the server can construct simple images with only one or two visual words and learn the behavior of hash functions from the returned sketch values. CPA can happen if the content owner's computer becomes compromised by hackers, and in this case even the secret key may be obtained by the hacker to decrypt all images in the database.

5. CONCLUSIONS

This paper makes the first endeavor on content-based retrieval over an encrypted multimedia database. Using image database as an example, we focus on building secure search indexes, which protect the privacy of image content from the server and preserve the capability of similarity comparison. Two secure indexing schemes, namely, secure inverted index and secure min-Hash sketches, are designed by jointly exploiting techniques from cryptography, image processing, and information retrieval. Both schemes can achieve good retrieval performance through encrypted indexes and serve as very good candidates for privacy-preserving multimedia retrieval. Future work will further improve the efficiency and security of search and retrieval, and enable more signal processing in the encrypted domain to achieve comprehensive secure data management.

REFERENCES

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

[2] 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.

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

[4] 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.

[5] 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.

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

[7] J. Shashank, P. Kowshik, K. Srinathan, and C. Jawahar, “Private content based image retrieval,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[8] D. Nistér and H. Stewénius, “Scalable recognition with a vocabulary tree,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[9] Y. Mao and M. Wu, “A joint signal processing and cryptographic approach to multimedia encryption,” *IEEE Trans. on Image Processing*, vol. 15, pp. 2061– 2075, 2006.

[10] M. Grangetto, E. Magli, and G. Olmo, “Multimedia selective encryption by means of randomized arithmetic coding,” *IEEE Trans. on Multimedia*, vol. 8, 2006.

[11] H. Kim, J. Wen, and J. D. Villasenor, “Secure arithmetic coding,” *IEEE Trans. on Signal Processing*, vol. 55, 2007.

[12] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60(2), pp. 91–110, 2004.

[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[14] J. Zobel and A. Moffat, “Inverted files versus signature files for text indexing,” *ACM Transactions on Database Systems*, vol. 23, pp. 453–490, 1998.

[15] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, “Order preserving encryption for numeric data,” in *Proc. of SIGMOD*, 2004.

[16] G. Salton and M. J. McGill, *Introduction to modern information retrieval.* McGraw-Hill, 1983.

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

[18] 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.

[19] 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.

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

[21] M. Slaney and M. Casey, “Locality-sensitive hashing for finding nearest neighbors,” *IEEE Signal Processing Magazine*, pp. 128–131, 2008.

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

[23] S. Jeong, C. Won, and R. Gray, “Image retrieval using color histograms generated by Gauss mixture vector quantization,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[24] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, “Supervised learning of semantic classes for image annotation and retrieval,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2007.