SECURE IMAGE RETRIEVAL THROUGH FEATURE PROTECTION

Wenjun Lu, Avinash L. Varna, Ashwin Swaminathan, and Min Wu

Department of Electrical and Computer Engineering, University of Maryland, College Park

ABSTRACT

This paper addresses the problem of image retrieval from an encrypted database, where data confidentiality is preserved both in the storage and retrieval process. The paper focuses on image feature protection techniques which enable similarity comparison among protected features. By utilizing both signal processing and cryptographic techniques, three schemes are investigated and compared, including bitplane randomization, random projection, and randomized unary encoding. Experimental results show that secure image retrieval can achieve comparable retrieval performance to conventional image retrieval techniques without revealing information about image content. This work enriches the area of secure information retrieval and can find applications in secure online services for images and videos.

Index Terms— Secure image retrieval, feature protection, relevance based search, content based image retrieval

1. INTRODUCTION

Information retrieval from encrypted databases is an important technological capability for privacy protection in multiparty information management. Representative application scenarios include online services of webmail such as Gmail, photo hosting such as Flickr, and financial management such as Mint.com, where users store their private information on some remote server and the server provides functionalities to the user, such as categorization, search, and data analysis. Currently, servers operate on plaintext data, making users' private information vulnerable to attacks by untrustworthy administrators and malicious intruders. To provide secure online services, technologies that protect users' privacy without sacrificing functionalities are desirable.

The growth of online photo services and the concerns of privacy protection make searching over encrypted images both attractive and necessary. A desirable feature for online photo services such as Google Picasa or Flickr would be the capability to encrypt and store private images, and later retrieve relevant images without revealing any information about the encrypted images to the server. Prior work on secure information retrieval was focused on text documents. Techniques for identifying the presence or absence of a keyword in an encrypted text document were proposed in [1, 2]. Recent work in [3] investigated secure rank-ordered search, where encrypted text documents are returned in the order of relevance to the query keyword.

Although secure text search techniques can be extended to image retrieval based on user assigned tags, extension to content based image retrieval (CBIR) is not straightforward. CBIR systems often rely on comparing the similarity among image features, such as color histograms, shape descriptors, or salient points, which are usually high dimensional vectors [4]. Comparing similarity among high dimensional vectors using cryptographic primitives is challenging. To the best of our knowledge, no existing techniques address secure feature comparison efficiently and effectively.

To build a secure CBIR system, both images and features should be protected. For a feature based retrieval system, images can be encrypted separately using cryptographic ciphers or image encryption algorithms. This paper focuses on the problem of image feature protection which allows the computation of similarity measures among encrypted features, so that secure CBIR can be achieved.

To our best knowledge, this work along with [5] are the first endeavors on content based image retrieval in an encrypted domain. We address the problem by jointly using signal processing and cryptographic techniques. Three feature protection schemes are explored and compared in terms of security, retrieval performance, and computational complexity. We show that retrieval performance comparable to conventional CBIR techniques can be achieved by the proposed feature protection schemes. These schemes can be used as building blocks to build efficient indexes, for search over large image databases. They can also be extended to secure video search by protecting features from the key frames.

2. FEATURE PROTECTION METHODOLOGY

Similarity of two images is typically measured by computing the distance between features extracted from the images [4]. For secure image retrieval, we seek to design techniques to encrypt image features, while approximately preserving 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 image 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, and c is a constant scaling factor. In the remainder of this section, we describe techniques to construct encryption functions that are approximately distance-preserving.

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

2.1. Bit-plane Randomization

The most significant bits (MSB) of an image capture important information about image appearance. The concept of processing bitplanes 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 distance are also likely to have similar patterns among their MSB bitplanes. This motivates us to investigate scrambling of feature values in such a way that the patterns in their MSB bitplanes are preserved.

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 $[b_{i1}, \cdots, b_{il}]$, where l is the total number of bitplanes. The jth bitplane of $\mathbf{f}$ is composed of the jth MSB of its components, denoted as $[b_{1j}, b_{2j}, \cdots, b_{nj}]$. The Hamming distance between two bitplanes is preserved when they are XORed with the same binary vector or when they are permuted using the same permutation pattern. We exploit this property to encrypt the top k bitplanes of the feature vectors while preserving their Hamming distances. The encryption of the jth bitplane of any feature vector is illustrated in Fig. 1. The bits comprising the bitplane are first XORed with a random bit sequence to hide the original number of 1's in each bitplane. The resulting bits are then randomly permuted to obtain the encrypted bitplane.

All the encrypted bitplanes form the encrypted feature vector $\mathcal{E}(\mathbf{f}) = [\tilde{f}_1, \cdots, \tilde{f}_n]$. The distance between two encrypted feature vectors $\mathcal{E}(\mathbf{f})$ and $\mathcal{E}(\mathbf{g})$ is computed as a weighted sum of the Hamming distance between their individual bitplanes:

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

Here $w(j)$s are the weights assigned to the bitplanes to reflect their unequal importance. $w(j)$ is chosen to be 2^{-j} in this paper. Since using the same permutation and XOR pattern on corresponding bitplanes of two feature vectors preserves their Hamming distance, we have

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

The distance $d_{\mathcal{E}}(\cdot, \cdot)$ between encrypted features upper bounds the original L_1 distance. The distortion between the distance distributions before and after the aforementioned encryption mainly comes from the fact that some feature vectors which have 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 we shall show in Section 3.1 that bitplane randomization leads to only a slight reduction in retrieval accuracy, as a trade-off for security.

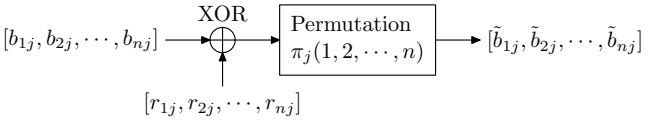

Fig. 1. Encryption of the jth bitplane

2.2. Random Projection

Random projection is based on the idea that close points in high dimensional space will remain close with high probability after projection onto a low dimensional space and has been used as a building block for developing efficient search techniques for large databases [6]. Random projection can be used to obfuscate the original values of the feature vectors while approximately preserving their distance.

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 then defined as $\mathcal{E}(\mathbf{f}) = \mathbf{R} \cdot \mathbf{f}$. Considering the L_1 distance of encrypted features, i.e. $d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g})) = \|\mathcal{E}(\mathbf{f}) - \mathcal{E}(\mathbf{g})\|_1$, the linearity of random projection makes the distribution of $d_{\mathcal{E}}(\cdot, \cdot)$ proportional to the original distribution in the Euclidean space, i.e. $d_{\mathcal{E}}(\mathbf{f}, \mathbf{g}) \approx c \cdot \|\mathbf{f} - \mathbf{g}\|_2$ with high probability for some scaling factor c [7]. L_1 distance between original feature vectors can be preserved by projecting $\sqrt{\mathbf{f}} = [\sqrt{f_1}, \cdots, \sqrt{f_n}]$ instead of $\mathbf{f}$. The distance distortion under both L_1 and L_2 distance metrics can be made arbitrarily small by increasing m. The projection dimension m controls the trade-off between retrieval performance and storage, as will be shown in Section 3.1.

2.3. Randomized Unary Encoding

Since non-integer valued features can be converted to integers after proper scaling and round off, we consider integer-valued feature vectors here and represent them in binary form through unary encoding. Unary encoding of a positive integer N is a binary string of N 1's followed by 0's. The unary encoding of a feature vector is a binary string formed by concatenating the unary representation of its components:

$$\mathcal{U}(f_i) = \underbrace{11\cdots11}_{f_i}\underbrace{00\cdots00}_{M-f_i}, \quad (3)$$

$$\mathcal{U}(\mathbf{f}) = [\mathcal{U}(f_1), \mathcal{U}(f_2), \cdots, \mathcal{U}(f_n)], \quad (4)$$

where M is the maximum possible value in all the feature vectors.

By performing XOR and randomly permuting the unary representation, we preserve the Hamming distance among $\mathcal{U}(\mathbf{f}), \forall \mathbf{f}$, which also equals the L_1 distance between original feature vectors. However, the disadvantage of using XOR and permutation alone is the storage increase from $O(n \log M)$ bits to $O(nM)$ bits. To reduce storage, we further apply random projection on $\mathcal{E}_1(\mathbf{f})$, which also helps enhance the security of the scheme, as will be shown in Section 3.2.

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 $\mathcal{E}(\mathbf{f}) = \mathcal{E}_2(\mathcal{E}_1(\mathbf{f})) \in \mathbb{R}^m$. Considering L_1 distance of encrypted features, we have the approximate distance preserving property

$$d_{\mathcal{E}}(\mathcal{E}(\mathbf{f}), \mathcal{E}(\mathbf{g})) \approx c \cdot \|\mathcal{E}_1(\mathbf{f}) - \mathcal{E}_1(\mathbf{g})\|_2 = c \cdot \|\mathbf{f} - \mathbf{g}\|_1. \quad (5)$$

The randomized unary encoding scheme effectively preserves the L_1 distance of original feature vectors with high probability and enhanced security.

3. PERFORMANCE ANALYSIS

3.1. Retrieval Accuracy

We evaluate the retrieval performance of image features encrypted by the three schemes on a subset of the Corel database containing 1000 images, obtained from [8]. 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. This database has been widely used as ground-truth for evaluating color image retrieval [4, 9]. We use global color histogram in the HSV space as image features. A 128-dimensional color histogram is generated from every image by quantizing the hue, saturation, and intensity channels into 8, 4, and 4 levels, respectively, where finer quantization is allocated to hue as suggested in [9].

We evaluate retrieval performance by the precision-recall curves. Every image in the database is used as a query and the average precision-recall curve for each protection scheme is obtained and shown in Fig. 2. As a reference, the results using plaintext color histograms from [9] are plotted as the top and the bottom curves. We can see that retrieval based on feature protection schemes achieve comparable performance to plaintext retrieval: better than plaintext retrieval based on L_2 distance and only slightly lower than plaintext retrieval based on L_1 distance, which is approximately preserved in the three schemes. By searching over encrypted features, we only need to retrieve about $1\% - 9\%$ more images to get the same number of relevant images as in plaintext search. Thus, secure retrieval can be achieved by trading off retrieval accuracy.

Among the three feature protection schemes introduced in Section 2, we can see the trade-off among retrieval performance, storage, and computational complexity. 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 (not shown in the figure). Bitplane randomization has time complexity $O(kn)$ which is lower than $O(mn)$ in random projection and $O(mnM)$ in randomized unary encoding. 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 is analyzed in the next section.

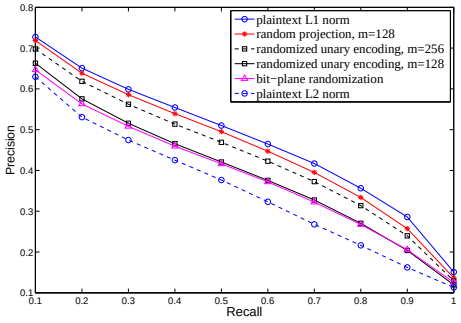

Fig. 2. Retrieval performance on the Corel database

3.2. Security Analysis

The random permutation, XOR pattern, and Gaussian random matrix used in above schemes are generated based on a user specified key K, which is kept secret from attackers. A cryptographically secure pseudo random number generator is used for generating keys and random numbers. Here we focus our security analysis on two attack models, namely, ciphertext only attack and known plaintext attack.

Ciphertext Only Attack (COA): This model assumes that the attacker, such as an untrustworthy server or malicious intruder, has access to encrypted features stored on the server but has no knowledge of the plaintext feature or the secret key. Under COA model, the attacker can compute distances between image features and infer which images in the database are similar. This information leakage is inevitable if we are to provide search capability. We show below that an attacker cannot gain any additional information about the database.

In COA model, guessing a secret key or the randomization pattern requires an exhaustive search over a prohibitively large space. For the 128-dimension color histogram, the number of possible permutations in the bitplane randomization scheme is around 10^{215}. An attacker may instead search the database with query features encrypted using a randomly selected key and analyze the retrieval results.

Denote by $\mathcal{E}_K(\mathbf{f})$ the feature $\mathbf{f}$ encrypted using the user's key K and by $\mathcal{E}_{K_a}(\mathbf{g})$ the feature encrypted using a wrong key K_a, where $\mathbf{g}$ is a feature known to the attacker. The distance between $\mathcal{E}_K(\mathbf{f})$ and $\mathcal{E}_{K_a}(\mathbf{g})$ in bitplane randomization scheme can be written as

$$d_{\mathcal{E}}(\mathcal{E}_K(\mathbf{f}), \mathcal{E}_{K_a}(\mathbf{g})) = \sum_{i=1}^{n} \sum_{j=1}^{l} |\tilde{b}_{ij}^{(\mathbf{f})} - \hat{b}_{ij}^{(\mathbf{g})}| \times 2^{-j}. \quad (6)$$

As $\tilde{b}_{ij}^{(\mathbf{f})}$ and $\hat{b}_{ij}^{(\mathbf{g})}$ are generated using two different keys, they are independent and $|\tilde{b}_{ij}^{(\mathbf{f})} - \hat{b}_{ij}^{(\mathbf{g})}|$ becomes a random variable equally likely to be 0 or 1 for all i, j. By the law of large numbers, $d_{\mathcal{E}}(\mathcal{E}_K(\mathbf{f}), \mathcal{E}_{K_a}(\mathbf{g}))$ is a constant for any $\mathbf{f}$ given $\mathbf{g}$ with high probability. Using similar arguments, it can be shown that the distance between two features encrypted using

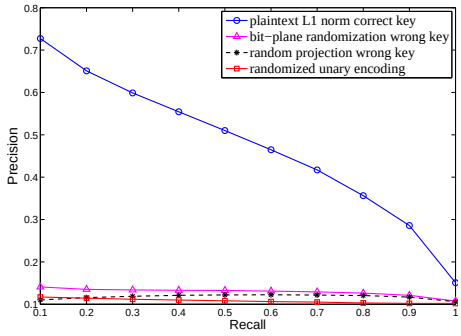

Fig. 3. Retrieval performance with wrong key.

random projection with different keys is a constant with high probability.

From the above analysis, we conclude that the feature encrypted using a wrong key has approximately equal distance to every encrypted feature in the database. Hence, retrieval using a wrong key is equivalent to randomly picking images from the database. Precision-recall curves for retrieval using a wrong key, shown in Fig. 3, verify this conclusion. The top curve is from plaintext search with a correct key and the remaining curves are for retrieval using an incorrect key, for each of the three schemes. We see that the curves of the three schemes using a wrong key have approximately a constant precision value of 0.1. Since there are 100 images for each category among the 1000 images, this verifies that retrieval over the encrypted database using a wrong key returns images randomly selected from the database.

Known Plaintext Attack (KPA): This model assumes that the attacker knows some pairs of plaintext features and corresponding encrypted features. We compare the security of the three schemes using the number of ciphertext-plaintext pairs required to accurately estimate the randomization.

In bitplane randomization, the XOR pattern can be obtained directly from one pair of ciphertext and plaintext once the permutation pattern is known. To evaluate the security of permutation under KPA model, we consider one bitplane $[b_{1j}, \cdots, b_{nj}]$ and its permuted version $[\tilde{b}_{1j}, \cdots, \tilde{b}_{nj}]$. It is clear that $n-1$ linearly independent binary vectors and their permuted versions will reveal the permutation pattern. Thus, the attacker requires $O(n)$ pairs of plaintext and ciphertext to break the scheme.

To evaluate the security of random projection under the KPA model with the pairs of $(\mathbf{f}_i, \mathcal{E}(\mathbf{f}_i)), i = 1, \cdots, k$, known by the attacker, we have $\mathcal{E}(\mathbf{F}) = \mathbf{R} \cdot \mathbf{F}$, where $\mathbf{F}, \mathcal{E}(\mathbf{F})$ have $\mathbf{f}_i, \mathcal{E}(\mathbf{f}_i)$ as their ith column, respectively. The encryption matrix $\mathbf{R}$ can then be easily obtained if $\mathbf{F}$ is invertible. Thus, the attacker requires $O(n)$ pairs of plaintext and ciphertext to break the random projection scheme.

In randomized unary encoding, feature encryption is done in two stages: $\mathbf{f} \rightarrow \mathcal{E}_1(\mathbf{f}) \rightarrow \mathcal{E}_2(\mathcal{E}_1(\mathbf{f})) = \mathcal{E}(\mathbf{f})$, where $\mathcal{E}_1$ denotes XOR and permutation and $\mathcal{E}_2$ denotes random projection. Deducing the encryption functions $\mathcal{E}_1(\cdot), \mathcal{E}_2(\cdot)$ requires knowledge of $(\mathbf{f}, \mathcal{E}_1(\mathbf{f}))$ and $(\mathcal{E}_1(\mathbf{f}), \mathcal{E}(\mathbf{f}))$. Plaintext $\mathbf{f}$ is decorrelated from ciphertext $\mathcal{E}(\mathbf{f})$ because $\mathcal{E}_1(\mathbf{f})$ is unknown. The security of randomized unary encoding under KPA model is essentially equal to the security of other two schemes under COA model. This implies that in applications which require higher security, we can use randomized unary encoding to provide enhanced security at the expense of increased computation.

4. CONCLUSIONS

This paper explores techniques which enable similarity comparison among encrypted image features, based on which secure content based image retrieval can be achieved. We show that the combination of signal processing and cryptographic techniques, such as random projection, unary encoding, and random permutation, helps us address the problem of secure image retrieval, which is otherwise difficult using traditional cryptography alone. The feature protection schemes explored in this paper exhibit retrieval performance comparable to the state-of-the-art techniques, and good trade-off can be achieved between security and computational complexity. These schemes can also be combined with efficient indexing techniques such as [6, 10] and scaled to large databases.

5. REFERENCES

[1] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches in Encrypted Data," *IEEE Symp. on Research in Security and Privacy*, pp. 44-55, 2000.

[2] D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano, "Public-key Encryption with Keyword Search," *Proc. of Eurocrypt*, pp. 506-522, 2004.

[3] 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," *Proc. of the ACM Workshop on Storage, Security, and Survivability*, pp. 7-12, Oct. 2007.

[4] R. Datta, D. Joshi, J. Li and J. Z. Wang, "Image Retrieval: Ideas, Influences, and Trends of the new age," *ACM Computing Surveys*, 2008.

[5] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling Search over Encrypted Multimedia Databases," to appear in *SPIE Media Forensics and Security XI*, Jan. 2009.

[6] A. Gionis, P. Indyk, and R. Motwani, "Similarity Search in High Dimensions via Hashing," *Proc. of the Int'l Conf. on Very Large Data Bases*, 1999.

[7] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality Sensitive Hashing Scheme based on p-stable Distributions," *Proc. of the ACM Symp. on Computational Geometry*, 2004.

[8] The image database used in this paper is available online at http://wang.ist.psu.edu/docs/related/

[9] 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, 2004.

[10] D. Nistér and H. Stewénius, "Scalable Recognition with a Vocabulary Tree," *CVPR*, 2006.