Encrypted JPEG image retrieval using histograms of transformed coefficients

Peiya Li* and Zhenhui Situ†
* Jinan University, Guangzhou, China
E-mail: lpy0303@jnu.edu.cn
† The Hong Kong Polytechnic University, Hong Kong
E-mail: z.situ@connect.polyu.hk

***Abstract*—This work proposes an encrypted JPEG image retrieval mechanism based on the histograms of transformed coefficients. With this scheme, JPEG image is encrypted during its compression process by using other orthogonal transforms for blocks' transformation, rather than 8×8 DCT. Then the encrypted images are transferred to and stored in the cloud server. When receiving an encrypted query image from the authorized user, the server calculates the histograms of transformed coefficients located at different frequency positions. By computing the distance between the histograms of encrypted query image and database cipherimages, encrypted images with plaintext content similar to the query image are retured to the authorized user for decryption. Experiments are conducted to show that our scheme can provide effective cipherimage retrieval service, while ensure format compliance and compression friendly.**

I. Introduction

In recent years, thanks to the rapid development of digital services, a huge number of images have been generated and shared over various types of social network platforms, such as Facebook and Instagram. Commonly, image data contain rich information that can be explored for different purposes, for example, feature extraction and image retrieval. Searching intended types of images from a large image database has attracted increasing attention, and many related image retrieval schemes have been proposed [1]–[3]. However, searching one specific image from a huge image dataset needs a lot of storage space and expensive computational cost, hence individuals tend to store their images into the cloud server because of its on-demand access to ample storage and computation resources.

Nonetheless, outsourcing image data with private/sensitive information to the cloud is vulnerable to attack, the image owners usually encrypt all images before uploading them to the cloud server. Conventional image encryption algorithms impede further meaningful operations, and pose a threat to feasible retrieval over encrypted image dataset. As a consequence, developing retrieval techniques that are capable of offering privacy-preserving and effective retrieval is highly urgent.

So far, many cipher-image retrieval algorithms have been proposed. In [4], Lu *et al.* proposed three visual feature protection schemes with distance preserving property, which enabled the cloud server to conduct similarity comparison among cipherimages. With the method in [5], the retrieval speed was further accelerated using secure search indexes. In [6], Xia *et al.* applied secure kNN algorithm to protect image feature vectors and designed a watermark-based protocol to deter illegal distributions of retrieved images by the authorized query users. Although the above work solve the privacy issue, they perform image encryption and feature extraction/encryption separately, which incurs extra computational workload and inconvenience for the content owner and authorized users. Therefore, retrieval schemes with features being directly extracted from the cipherimage have been developed. Cheng *et al.* [7] proposed to encrypt JPEG images by permuting DCT coefficients and conduct cipherimage retrieval based on these coefficients' histogram invariance. This mechanism freed the users from feature extraction/encryption, however, the encryption technique caused file size increasing and histogram information leakage. Another encrypted JPEG image retrieval scheme was realized in [8] by encrypting the DC and AC coefficients of JPEG images using a stream cipher and scrambling encryption, respectively. The encryption operations did not modify the histogram of AC coefficients, hence this information was utilized by the server for retrieval. Obviously, information leakage problem also existed in this method. Moreover, this scheme could not ensure JPEG format compliance preservation. To solve the problem, Ref. [9] developed a Markov process based retrieval scheme. The coded data was encrypted by a stream cipher, and the Markov features could be directly extracted form the encrypted data. But, it was a supervised retrieval mechanism, which indicated that the image dataset used for training the retrieval model needed to be provided in advance. In [10], they exploited block-wise feature comparison for encrypted JPEG image retrieval. This scheme was an unsupervised mechanism, however, it disclosed feature of the plainimages, and the efficiency of block-wise feature comparison was unacceptable [11]. To overcome the message leakage problem, Liang *et al.* [12] proposed to change the huffman-code histograms of JPEG images through a stream cipher and permutation cipher, however, severe bit-stream size increment was incurred.

In this work, we follow the framework of Refs. [7]–[10], namely, feature extraction/encryption is unnecessary on the user side. An encryption technique using multiple new orthogonal transforms for blocks' transformation is developed. The whole encryption operation is embedded into JPEG's transformation stage, which is a must-step during compression.

Experimental results indicate that plain-images are protected well without leaking any information to the server, for example, the histogram of DCT coefficients. Moreover, the compression efficiency of JPEG and format compliance can be maintained under the encryption algorithm. After encryption, histograms of transformed coefficients located at different frequency positions are calculated for cipher JPEG retrieval, and the final retrieval performance is pretty good.

The remainder of the paper is organized as follows. Section II describes the proposed scheme containing image encryption and image retrieval. Experimental results are provided in Section III. Performance of our scheme are evaluated from three aspects: retrieval performance, security analysis, and compression efficiency. Section IV gives a conclusion and presents future research directions.

II. Proposed scheme

The proposed scheme involves three different types of entities: image owner, authorized user, and cloud server. Image owner encrypts images in JPEG format with the secret encryption key and stores the cipherimages into the cloud. An authorized user requests image retrieval service by uploading an encrypted JPEG query image to the cloud server. When receiving the encrypted query image, the server calculates the similarities between the query cipherimage and database cipherimages, without the knowledge of the encryption key. Different similarities are then sorted by the server, and the encrypted images closest to the query image in plaintext content are returned to the authorized user for decryption.

A. Image encryption

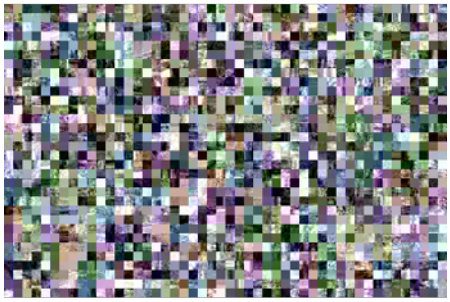
As JPEG is the majority of image format used by various social network platforms as well as digital cameras, retrieval on encrypted JPEG images is undoubtedly of great practical significance. Purpose of our scheme is to address the problem of image retrieval in encrypted domain, meanwhile preserve the format compliance and file size of JPEG images. In JPEG compression standard, three major stages are included: DCT transformation stage, quantization stage, and entropy coding stage. In our encryption method, we merely conduct encryption at the transformation stage by replacing original 8×8 DCT with the orthogonal transform set developed in [13], followed by 8×8 blocks' permutation. The other two stages of JPEG remain unchanged.

In [13], the orthogonal transform set is only tested on grayscale image, here we extended it for transforming blocks of color image. The pseudo-random key generator we select is BLAKE2, because of its fast speed and security. The block permutation algorithm adopted is the two key-driven cyclical shift developed in [14]. To improve the security, different keys are used for different components of a color image. The encryption algorithm is described as follows:

Encryption Algorithm

Step 1: Initialize the BLAKE2 key generator by the predefined secret keys;

(a)



(b)

Fig. 1: Encryption example: (a) original image, (b) encrypted image.

Step 2: For each color image, YCbCr components are used with 4:2:0 chroma sampling format;
Step 3: Divide each component into non-overlapping 8×8 blocks, and for each 8×8 block, do
Step 3.1: Get 63 bits from the random generator;
Step 3.2: The first 7 bits are used to select one transform from the transform set for *all* rows in the 1^{st} dimension;
Step 3.3: The following 56 bits are used to select one transform from the transform set for *each* column in the 2^{nd} dimension;
Step 3.4: Transform the 8×8 block using selected transforms;
Step 4: Permute all 8×8 transformed blocks of each component, then perform JPEG's quantization, and entropy coding procedures;
Step 5: Repeat *Step 3* and *Step 4* until all the three components Y, Cb, and Cr are processed, generate the encrypted bit-stream and transmit it to the cloud server;

For the decryption algorithm, we just follow JPEG's decoding process by using the corresponding secret keys. If keys are unavailable, the decoding procedure of JPEG can still be applied on the encrypted bit-stream, because of the format compliant property. In Fig. 1, we take an architecture image as example to show the encryption performance of our encryption algorithm.

B. Image retrieval

Upon receiving the encrypted bit-stream of the query image from the authorized user, the server first parses the encrypted JPEG bit-stream to extract the Huffman codes for transformed coefficients of each component. The extraction operation can be successfully accomplished, since the encryption technique does not destroy the format information of JPEG. Then the

server rearranges the entropy decoded coefficients into 8×8 blocks, and uses the quantization table extracted from the JPEG bit-stream header to de-quantize these blocks' coefficients.

In a 8×8 de-quantized block, there are 64 coefficients belonging to different frequencies, we pick coefficients on the same frequency position to form a group. Coefficients from different components are picked separately. For example, the DC coefficients of all 8×8 blocks are put together as one group, the first AC coefficients of all 8×8 blocks form the next group, and so on. Totally 64 coefficient groups will be created for each component. We denote the occurrence number of a value x in frequency position i as $H_i(x)$, $1 \leq i \leq 64$, and the interpolation process and normalization operation are implemented on all histograms. For each two given encrypted images $C1$ and $C2$, we compute the distance between their histograms of a certain component using the following formula:

$$d = \sum_{i=1}^{64} \sum_{x=-\infty}^{+\infty} \frac{\left|H_i^{C1}(x) - H_i^{C2}(x)\right|}{1 + H_i^{C1}(x) + H_i^{C2}(x)}. \quad (1)$$

Since there are three components, totally three distances can be computed, and they are represented as d_Y, d_{Cb}, and d_{Cr}. We can derive an integrated distance from the following equation:

$$D = w_1 \times d_Y + w_2 \times d_{Cb} + w_3 \times d_{Cr}. \quad (2)$$

where $w_j (j = 1, 2, 3)$ is a weight parameter that indicates the importance of each component, and the range of w_j is $[0, 1]$.

III. Experimental Results

In this section, the experiment simulations evaluate the proposed scheme from three aspects (retrieval performance, security analysis, and compression performance), and they are compared with the state-of-the-art algorithms. We use Corel image database [15] for test, which contains ten different image categories with 100 images of each category, totally 1000 images. The ten categories are: African, Beach, Architecture, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountain, and Food.

A. Retrieval performance

We use the precision-recall curve (P-R curve) to evaluate the proposed scheme's retrieval efficiency. The precision-recall curve is defined as follows:

$$Precision = \frac{N_P}{N_R}, Recall = \frac{N_P}{N_A}. \quad (3)$$

where N_P is the number of positive/correct images in all returned images, N_R is the number of all returned images, N_A is the number of positive/correct images in the whole image dataset.

We first encrypt all 1000 images under our encryption algorithm, and use every cipherimage as a query image submitted to the server, and we can get a number of retrieval results according to the distance obtained from (2). The precision-recall performance varies with respect to the retrieval results, and the average precision-recall performance can be calculated. Fig. 2 illustrates the average P-R curves when different combinations of the Y, Cb, and Cr components are adopted for similarity measure in our scheme. It can be seen that the retrieve performance of the integrated distance ($w_1 = w_2 = w_3 = 1$) is better than that of the single distances. In Fig. 3, we have compared the retrieval performance of our scheme with methods in [7] and [8], in which the $w_1 = w_2 = w_3 = 1$ case is considered. From the figure, we can observe that the P-R curve of our proposed scheme is much higher than that of [8], which indicates a better retrieval performance. Ref. [7] achieves the best retrieval efficiency, because the retrieval basis they use is the histograms of DCT coefficients, which remain same before and after encryption, and this obviously leaks information of plainimages.

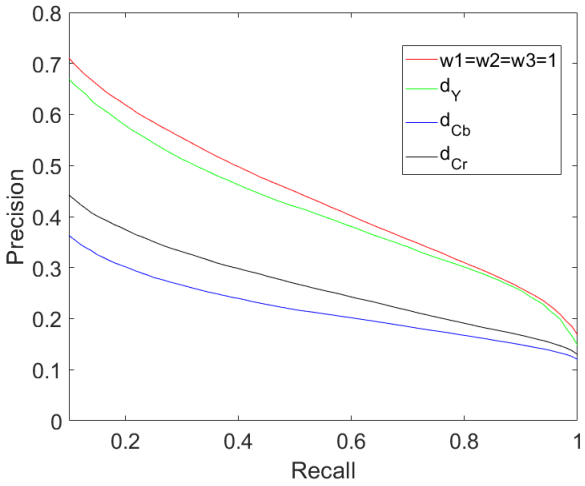


Fig. 2: Precision-recall performance of our scheme.

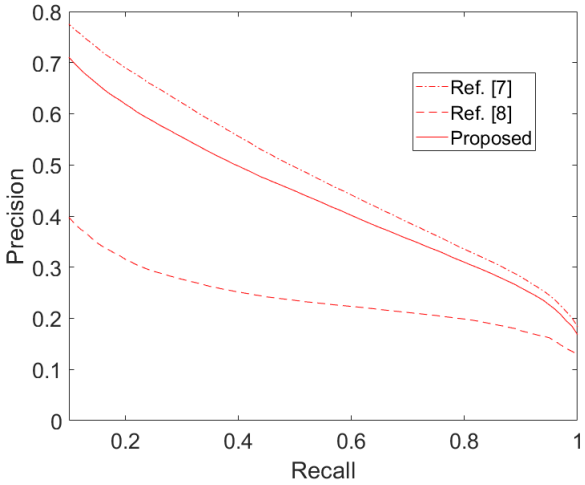


Fig. 3: Retrieval performance comparison.

B. Security analysis

1) Brute-force attack: This attack is a typical attacking method in ciphertext-only attack, in which attackers only have access to the encrypted data. To make this attack infeasible, key space of the cryptosystem should be large enough. In our scheme, we use three different keys for the Y, Cb, and Cr components of color images. These keys are 128-bit each, and are set as the input of BLAKE2 hash function to produce three different pseudo-random key-streams. Therefore, the key space of our scheme is $(2^{128})^3$, which is large enough to resist the brute-force attack. Additionally, if we adopt varying keys for different plainimages, the key space will be further enlarged.

2) Statistical model-based attack: The reason why we evaluate this attacking method is because many existing retrieval algorithms [7], [8], [10] on cipher JPEG images cannot withstand it. In this type of attack, attackers try to predict the plainimage without knowing the key through studying the predictable relationship of some data segments between plainimage and cipherimage. Generally, histograms of the original image and the encrypted image are one of the common ways to denote the degree of relationship. To make the statistical model-based attack unavailable, plainimages's histogram and cipherimage's histogram should be different. In [7], they realized encryption through coefficients shuffling operation, histograms of DCT coefficients are not changed, which means this scheme is vulnerable to statistical attack. For the encryption method proposed in [8], they encrypted DC coefficients using a stream cipher, thus histogram of DC coefficients changed. However, for AC coefficients encryption, only block permutation was conducted, and this operation did not alter the histogram of AC coefficients. In our scheme, we replace the original 8×8 DCT with new orthogonal transforms, which will alter the values of original DCT coefficients, together with the histograms of transformed coefficients.

C. Compression performance

Since our encryption and retrieve operations are conducted on JPEG images, guaranteeing that the compression efficiency of JPEG is not compromised is particularly important, and this will save the transmission bandwidth and storage space. We use bit per pixel (BPP) and peak signal-to-noise ratio (PSNR) value to evaluate the compression performance of our scheme. In Fig. 4, the plain architecture image shown in Fig. 1 are taken as example to show the BPP-PSNR curve of our scheme and Ref. [7], compression performance of Ref. [8] is not listed due to the format incompatible. From Fig. 4, it is clear that when the encryption keys are not available, the PSNR value of our scheme has larger drop compared with [7], which indicates a stronger protection ability. Moreover, when the keys are available to the decoder, the BPP-PSNR curve obtained by our method is much closer to JPEG, indicating a better compression efficiency than method in [7].

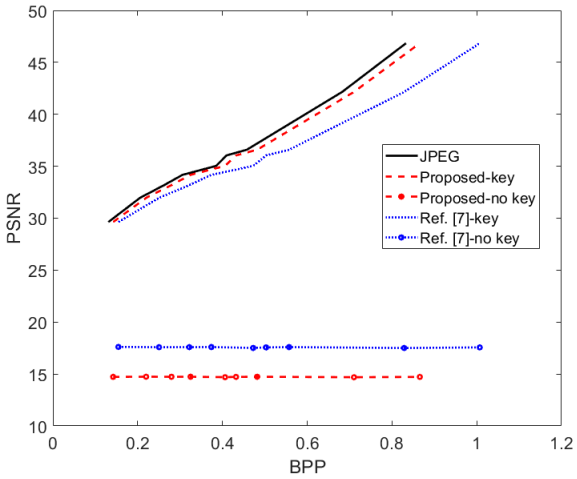


Fig. 4: Compression efficiency comparison.

IV. Conclusion

In this paper, a novel encrypted JPEG image retrieval scheme is developed by extracting the histogram of transformed coefficients as cipherimage's feature for similarity measure. These transformed coefficients are different from the DCT coefficients of JPEG, since we modify JPEG's transformation stage through applying new orthogonal transforms for blocks' transformation, rather than 8×8 DCT. This strategy does not produce additional computation, when compared with several existing algorithms [7]–[10]. Because their encryption operations are performed on elements produced from intermediate/final stages of JPEG, and this leads to an extra process during compression. The experimental results have shown that our scheme can achieve good retrieval accuracy and security, without sacrificing the compression performance of JPEG.

In our next work, we plan to exploit other kinds of features, rather than only the histogram of transformed coefficients, so as to improve the retrieval accuracy. The relationship between the security and retrieval performance will be investigated in more detail. Besides, other image formats will also be taken into account.

Acknowledgment

This work is supported by "the Fundamental Research Funds for the Central Universities", project no. 21619314.

References

[1] G. Schaefer, "Fast compressed domain jpeg image retrieval," in *2017 International Conference on Vision, Image and Signal Processing (ICVISP)*. IEEE, 2017, pp. 22–26.

[2] Y. Wang, M. Shi, S. You, and C. Xu, "Dct inspired feature transform for image retrieval and reconstruction," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4406–4420, 2016.

[3] P. Poursistani, H. Nezamabadi-pour, R. A. Moghadam, and M. Saeed, "Image indexing and retrieval in jpeg compressed domain based on vector quantization," *Mathematical and Computer Modelling*, vol. 57, no. 5-6, pp. 1005–1017, 2013.

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

[5] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," in *Media Forensics and Security*, vol. 7254. International Society for Optics and Photonics, 2009, p. 725418.

[6] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[7] X. Zhang and H. Cheng, "Histogram-based retrieval for encrypted jpeg images," in *2014 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*. IEEE, 2014, pp. 446–449.

[8] H. Cheng, X. Zhang, and J. Yu, "Ac-coefficient histogram-based retrieval for encrypted jpeg images," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13 791–13 803, 2016.

[9] H. Cheng, X. Zhang, J. Yu, and F. Li, "Markov process-based retrieval for encrypted jpeg images," *EURASIP Journal on Information Security*, vol. 2016, no. 1, p. 1, 2016.

[10] H. Cheng, X. Zhang, J. Yu, and Y. Zhang, "Encrypted jpeg image retrieval using block-wise feature comparison," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 111–117, 2016.

[11] H. Liang, X. Zhang, H. Cheng, and Q. Wei, "Secure and efficient image retrieval over encrypted cloud data," *Security and Communication Networks*, vol. 2018, 2018.

[12] H. Liang, X. Zhang, and H. Cheng, "Huffman-code based retrieval for encrypted jpeg images," *Journal of Visual Communication and Image Representation*, vol. 61, pp. 149–156, 2019.

[13] P. Li and K.-T. Lo, "A content-adaptive joint image compression and encryption scheme," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 1960–1972, Aug. 2018.

[14] J. Zhou, X. Liu, O. C. Au, and Y. Y. Tang, "Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation," *IEEE transactions on information forensics and security*, vol. 9, no. 1, pp. 39–50, 2013.

[15] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 9, pp. 947–963, 2001.