An Encryption-then-Compression System for JPEG XR Standard

Kenta Kurihara
Tokyo Metropolitan University
6–6, Asahigaoka, Hino, Tokyo, Japan 191–0065
Email: kurihara-kenta@ed.tmu.ac.jp
Telephone: 81–42–585–8454

Osamu Watanabe
Takushoku University
815–1, Tatemachi, Hachiouji, Tokyo, Japan 193–0985
Email: owatanab@es.takushoku-u.ac.jp

Hitoshi Kiya
Tokyo Metropolitan University
6–6, Asahigaoka, Hino, Tokyo, Japan 191–0065
Email: kiya@tmu.ac.jp
Telephone: 81–42–585–8454

***Abstract*—In many multimedia applications, image encryption has to be conducted prior to image compression. This paper proposes an Encryption-then-Compression system using a JPEG XR friendly perceptual encryption method, which enables to be conducted prior to JPEG XR compression. The proposed encryption method can provide approximately the same compression performance as that of JPEG XR compression without any encryption. It is also shown that the proposed system consists of four block-based encryption steps, and provides a reasonably high level of security. Most of conventional perceptual encryption methods have not been designed for international compression standards, but for the first time this paper focuses on applying the JPEG XR standard, which supports lossy and lossless coding for various kinds of images including high dynamic range images.**

***Index Terms*—Content protection and watermarking, Audio technology, Video coding and processing**

I. Introduction

With the wide/rapid spread of distributed systems for information processing, such as cloud computing and social networks, not only transmission but also processing is done on the public Internet, and thus contents are transmitted over an insecure bandwidth-constrained communication channel [1]. In the meantime, a lot of studies on secure, efficient and flexible communications have been reported. For securing multimedia data, full encryption with a state-of-the-art cipher (like RSA, AES, etc.) is the most secure option. However, many multimedia applications have been seeking a trade-off in security to enable other requirements, e.g., low processing demands, retaining bitstream compliance, and signal processing in the encrypted domain. Because of this situation, a lot of perceptual encryptions have been studied as one of schemes for achieving the trade-off [2].

In this paper, we focus on an Encryption-then-Compression (ETC) system [3], although the traditional way of securely transmit images is to use a Compression-then-Encryption (CTE) system. Recently, the JPEG committee has started to standardize a new work item, referred to as JPEG Privacy, in which secure transmission between network servers in cloud computing and social networks is supposed as one of the technical requirements.

978–1–4673–9044–6/16/$31.00

Most of the conventional works for ETC systems have no compatibility with the international standards, e.g., JPEG, JPEG XR, JPEG 2000, etc. [3]–[6]. Also, a number of perceptual encryption schemes have been studied for the international standards, but they do not correspond to ETC systems [7]–[10], except for the articles regarding JPEG and JPEG 2000 [11]–[13]. Because of such situations, a new ETC system is proposed under the assumption of the JPEG XR standard for the first time. JPEG XR allows lossy and lossless coding for still images and videos. It supports not only images with 8 bits, but also images with over 8 bits and floating point representation. Thus, it can support various kinds of images including high dynamic range (HDR) images. Therefore, the proposed scheme is widely available for many kinds of images, which can not be supported by JPEG and JPEG 2000 standards.

In the proposed system, an image is first divided into non-overlap blocks as well as in the JPEG XR standard, and then each block is encrypted by four block-based steps, which enable to control a visibility condition including color alteration and a level of security. Moreover, the proposed algorithm has approximately the almost same compression performance as that of the JPEG XR compression without any encryption.

II. Preparation

A. ETC system

In this paper, we focus on image ETC systems as illustrated in Fig. 1, in which a content owner Alice wants to securely and efficiently transmit an image I to a recipient Bob, via an untrusted channel provider Charlie. In particular, the use of the JPEG XR standard is supposed as a compression method.

B. JPEG XR standard

The JPEG XR is an image coding standard from the JPEG committee [14]. It allows lossy and lossless coding for still images and videos. It supports not only fixed point representation but also floating point representation. Thus, it can support various kinds of images including HDR images for a new generation of digital cameras.

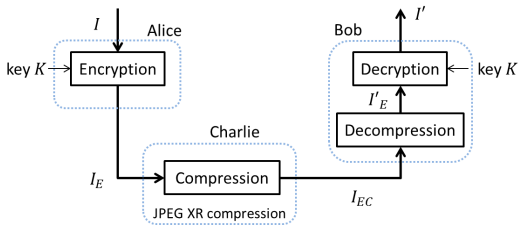


Fig. 1. Encryption-then-Compression system

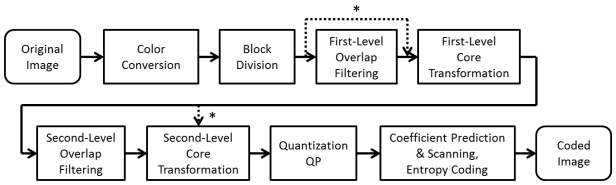


Fig. 2. JPEG XR coding (*: There are three modes)

The block diagram of JPEG XR encoding is illustrated in Fig. 2. The encoding consists of the following basic steps:

1) Performing a color conversion to YUV space.
2) Dividing an image into non-overlapped consecutive 16×16 blocks, called *macro block*, and then each macro block into consecutive 4×4 blocks, called *block* (see Fig. 3).
3) Applying two basic operators i.e., optional overlap filtering to the blocks and core transform, where the operators are hierarchically executed twice shown in Fig. 3.
4) Applying a coefficient quantization approach controlled by quantization parameters (QPs).
5) Executing adaptive coefficient scanning to convert the two-dimensional array transform coefficients within a block into a one-dimensional vector to be encoded. Finally, the coefficients are entropy encoded.

In step 3), one temporal DC coefficient and 15 HP coefficients are obtained for each block by the 1st-level core transform, and 16 temporal DC coefficients are gathered from each macro block as shown in Fig. 3. The 2nd-level core transform is then applied to them. As a result, one DC coefficient, 15 LP coefficients and 15×16 HP coefficients are calculated for each macro block. The two-level transform is referred to as a lapped biorthogonal transform (LBT). Therefore the transform coefficients are often called LBT coefficients, which consist of DC, LP and HP ones.

The overlap filtering is optionally used to reduce blocking artifacts. JPEG XR has three overlapping-modes. When mode 0 is chosen, no overlap filtering is performed. Otherwise, only the 1st-level overlap filtering is performed for mode 1, and both filtering operations are done for mode 2.

III. Proposed method

A. Block-based encryption with four steps

We investigate a block-based encryption scheme, in which an image with $M \times N$ pixels is divided into non-overlapped

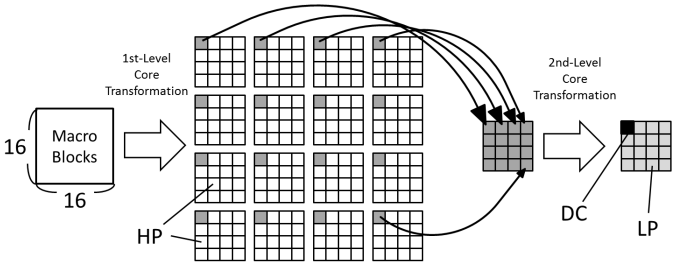
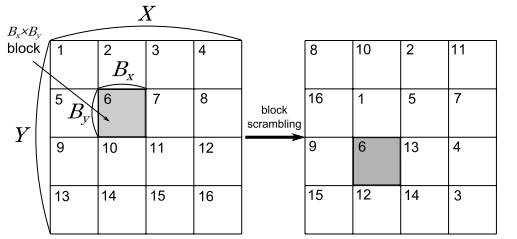


Fig. 3. Lapped biorthogonal transform



(a) Block division (b) Block scrambling

Fig. 4. Block division and block scrambling

consecutive blocks with $B_x \times B_y$ pixels as shown in Fig. 4. The proposed system consists of four block-based steps as illustrated in Fig. 5, so that it has a high relationship with the JPEG XR standard.

The procedure of performing the proposed image encryption is given as follows:

Step1: Divide each color component of a color image $I = \{I_R, I_G, I_B\}$ into $B_x \times B_y$ blocks respectively (see Fig. 4(a)). The i-th block image is defined as $I(i) = \{I_R(i), I_G(i), I_B(i)\}$ where $i = 1, 2, \cdots$ is a block number.

Step2: Permute randomly the divided blocks using a random integer generated by a secret key K_1 (see Fig. 4(b)), where K_1 is commonly used for all color components.

Step3: Rotate and invert randomly each $B_x \times B_y$ block (see Fig. 6) using a random integer generated by a key K_2, where K_3 is commonly used for all color components as well.

Step4: Apply the negative-positive transformation to each $B_x \times B_y$ block using a random integer generated by a key K_3, where K_3 is commonly used for all color components.

Step5: Shuffle three color components in each $B_x \times B_y$ block using a random integer generated by a key K_4.

Step6: Generate the encrypted image by integrating the transformed block images.

1) Block scrambling: An original image with $M \times N$ pixels is divided into blocks $B(i)(i = 1, 2, \cdots, L)$ with $B_x \times B_y$ pixels, where L is the number of divided blocks and computed by

$$L = \lfloor \frac{M}{B_x} \rfloor \times \lfloor \frac{N}{B_y} \rfloor \quad (1)$$

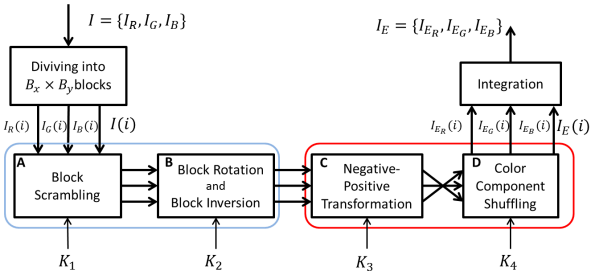
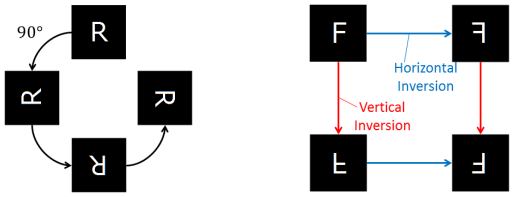
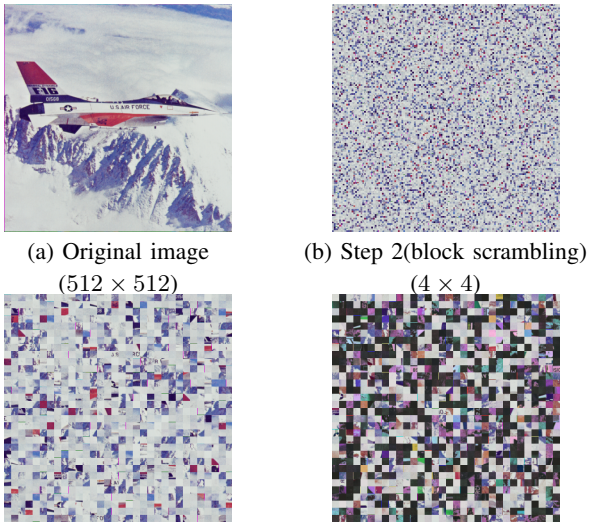


Fig. 5. Four block-based steps for encryption



(a) Block rotation (b) Block inversion

Fig. 6. Block rotation and inversion



(a) Original image (512×512) (b) Step 2(block scrambling) (4×4)

(c) Step 2 (block scrambling) (16×16) (d) Step 6 (proposed) (16×16)

Fig. 7. Encrypted images ($B_x \times B_y$)

where $\lfloor \cdot \rfloor$ is the function that rounds down to the nearest integer. The divided blocks are randomly permuted (see Fig. 4(b)).

Figures 7(b) and 7(c) show permuted images of the image in Fig. 7(a) with the block sizes 4×4 and 16×16 respectively.

2) Block Rotation and Block Inversion: As shown in Fig. 6(a), the block rotation is the operation that rotates randomly each block either $0°$, $90°$, $180°$ or $270°$ where $B_x = B_y$ is supposed. Also, the block inversion is the operation that inverts each block horizontally and vertically as shown in Fig. 6(b). There are four patterns of block inversion. Note that each rotated or inversed block has the same DC component as the original block.

The images encrypted by the methods *1)* and *2)* have the same color and histogram as those of the original image. Next encryption steps can improve these issues.

3) Negative-Positive Transformation: The negative-positive transformation is the operation that reverses all of the pixel values in each $B_x \times B_y$ block by using a random number of either zero or one. In the i-th block, the transformed pixel value p' is computed by

$$p' = \begin{cases} p & (r(i) = 0) \\ 255 - p & (r(i) = 1) \end{cases} \tag{2}$$

where p is the pixel value of an original image with 8 bpp, and $r(i)$ is a random integer given for the i-th block.

4) Color Component Shuffling: The color component shuffling is the operation that permutes values among R, G and B components by using a random integer in each $B_x \times B_y$ block. Table I shows permutation of color components corresponding to the random integer. Figure 7(d) shows an encrypted image by using the above four steps, where its original image is Fig. 7(a). The color and histogram of the images encrypted in Step 4 and Step 5 are different from those of the original image.

TABLE I
PERMUTATION OF COLOR COMPONENTS FOR A RANDOM INTEGER

Random Integer	R	G	B
0	R	B	G
1	G	R	B
2	G	B	R
3	B	R	G
4	B	G	R

B. Block size

As shown in Fig. 3, the JPEG XR is a block-based coding standard. Its compression processes are performed for each 16×16 macro block and each 4×4 block included in a macro block. Therefore, an image is split into 16×16 blocks as MCU (Minimum Coded Unit). In the case that an encrypted image is compressed by using the JPEG XR, it is expected that the proposed scheme has high compression performance, when the block size $B_x \times B_y$ is chosen as 16×16 or its multiple integer.

IV. KEY SPACE ANALYSIS

There are several kinds of attack on an encryption, such as the brute-force attack, the differential attack, the statistical attack, and so on. In this work, we evaluate the safety of the proposed system with its key space, assuming that an attacker performs the brute-force attack. In the proposed scheme, the key space is determined by the number of divided blocks. If an original image with $M \times N$ pixels is divided into blocks with $B_x \times B_y$ pixels, the number of the blocks L is computed by Eq. (1)

In the block scrambling, the key space N_B, which is the number of permutation of L blocks, is given by

$$N_B = {}_L P_L = L!. \tag{3}$$

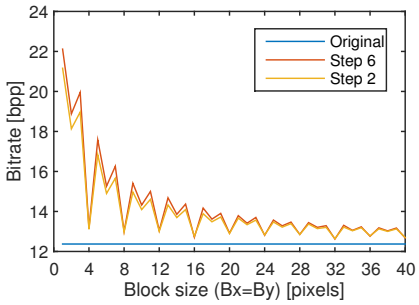


Fig. 8. Bitrate-Block size curves (Airplane)

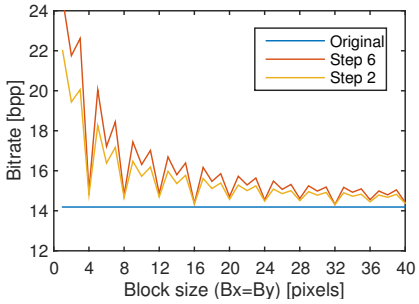


Fig. 9. Bitrate-Block size curves (Lena)

Similarly, when the key spaces of other encryption steps are given as

$$N_R = 8^L,\ N_N = 2^L,\ N_C = \left({}_3P_3 - 1\right)^L = 5^L \qquad (4)$$

where N_R, N_N and N_C are the key spaces of the encryption combining the block rotation and the block inversion, the negative-positive transformation and the color component shuffling respectively. Even N_N, which is the smallest key space in the above key spaces, is larger than 2^{256} when $L > 256$, i.e., the key space of the proposed scheme is larger than that of the 256-bit key, when the divided image has more than 256 blocks at least.

Consequently, the key space of encrypted images by using all the proposed encryption steps, N_A, is represented by

$$\begin{aligned} N_A &= N_B \cdot N_R \cdot N_N \cdot N_C \\ &= L! \cdot 8^L \cdot 2^L \cdot 5^L. \end{aligned} \qquad (5)$$

As a result, if an encrypted image has more than 28 blocks, the key space of the image is larger than that of the 256-bit key.

V. Simulation

We evaluate the effectiveness of the proposed encryption method by a number of simulations. Five images were reversibly or irreversibly compressed by the JPEG XR standard. Airplane, Lena, Mandrill, Milkdrop and Pepper (512×512, RGB24bpp) were used as the test images.

TABLE II
Compression results [bpp]

	Original	16×16	4×4
Mandrill	18.45	18.78	19.09
Lena	14.18	14.37	14.98
Airplane	12.36	12.71	13.16
Pepper	15.39	15.46	16.02
Milkdrop	12.18	12.28	13.16
Average	14.51	14.72	15.28
Ratio	1	1.015	1.053

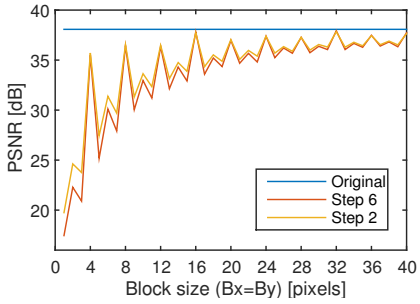


Fig. 10. PSNR-Block size curves (Airplane, 2.0 [bpp])

A. Lossless compression

Figures 8 and 9 show the relationship between bitrates and block sizes for reference images, Airplane and Lena, respectively, compressed by the JPEG XR reversibly. In the both figures, when 16×16 or 32×32 is chosen as a block size, which is the proposed size, the compression performance of encrypted images is almost same as that of the original image without any encryption. Also when 4×4 or its multiple integer is chosen, the encrypted image has a good compression performance due to two core transform operations shown in Fig. 3.

Table II shows the compression results for reference images and ones encrypted by the proposed scheme with $B_x \times B_y = 16 \times 16$ or $B_x \times B_y = 4 \times 4$. When the block size is equal to 16×16, which is the same size as macro blocks of the JPEG XR, the performance degradation is just one to two percent under the criteria of the number of bits per pixel (bpp), compared to the case without any encryption. On the other hand, when the block size is equal to 4×4, which is the same size as blocks, the compression performance becomes a little lower than the case with the size 16×16, but the key space increases due to a larger number of blocks. There is the trade-off relation between the compression performance and the security level, and thus a user should choose a suitable block size depending on the application scenario.

B. Lossy compression

We irreversibly compressed encrypted images, and then computed the PSNR values of the decrypted ones. Figures 10 and 11 show the relation between PSNR values and block sizes for the reference images, Airplane and Lena, respectively with 2.0 bpp. When 16×16 or 32×32 is chosen as a block size, the compression performance of encrypted images is almost

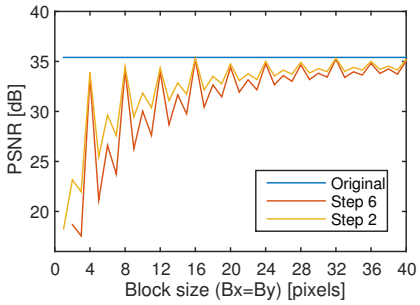


Fig. 11. PSNR-Block size curves (Lena, 2.0 [bpp])

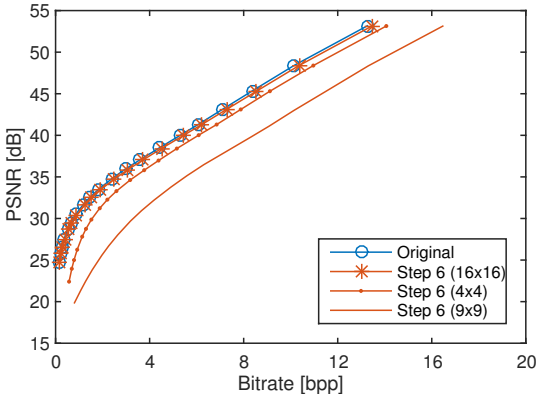


Fig. 12. RD curves of decrypted images (average)

same as that of the original image without any encryption, and when 4×4 or its multiple integer is chosen, the encrypted image has a good compression performance, as with lossless compression. Figure 12 shows Rate-Distortion (RD) curves of the test images, where each PSNR values is the average of the test images. The figure shows that the curve of images encrypted by the proposed scheme is close to that of the original images without any encryption, with any bitrates.

VI. Conclusion

This paper proposed an efficient ETC system for the JPEG XR standard. Four block-based encryption steps were used as the perceptual encryption schemes in the proposed system. We evaluated the safety of the proposed system with its large key space. Considering the JPEG XR standard, an appropriate block size was also proposed. The experimental results demonstrated that the proposed block size was suitable for the JPEG XR compression and the proposed system achieved both acceptable compression in cases of lossless and lossy compression and sufficient security for secure image communication while maintaining the compatibility with the JPEG XR standard. Using the proposed system, the user can control the safety and the compression performance by selecting a suitable block size.

References

[1] C. T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varadharajan, and C. C. J. Kuo, "Survey on securing data storage in the cloud," APSIPA Transactions on Signal and Information Processing, vol.3, e7, Jun. 2014.

[2] A. K. Jain, K.Nandakumar, and A. Nagar, "Biometric template security," EURASIP J. Adv. Signal Process, vol.2008, p.113, Jan. 2008.

[3] 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 Trans. Inf. Forens. Security, vol.9, no.1, pp.39-50, Jan. 2014.

[4] R. Hu, X. Li, and, B. Yang, "A new lossy compression scheme for encrypted gray-scale images," In Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.7436-7440, May 2014.

[5] X. Zhang, "Lossy compression and iterative reconstruction for encrypted image," IEEE Trans. Inf. Forens. Security, vol.6, no.1, pp.53-58, Mar. 2011.

[6] D. Klinc, C. Hazay, A. Jagmohan, H. Krawczyk, and T. Rabin, "On compression of data encrypted with block ciphers. Information Theory," IEEE Trans. Inf. Theory, 58(11), 6989-7001, Nov. 2012.

[7] M. I. Khan, V. Jeoti, and M. A. Khan, "Perceptual encryption of JPEG compressed images using DCT coefficients and splitting of DC coefficients into bitplanes," International Conference on Intelligent and Advanced Systems (ICIAS), pp. 1-6, Jun. 2010.

[8] D. Engel, T. Stutz, and A. Uhl, "Assessing JPEG2000 encryption with key-dependent wavelet packets, " EURASIP Journal on Information Security, 2012(1), 1-16, Apr. 2012.

[9] H. Kiya, S. Imaizumi, and O. Watanabe, "Partial-scrambling of image encoded using JPEG2000 without generating marker codes, " In Proceedings of the IEEE International Conference on Image Processing (ICIP). Volume III, pp.205-208, Sep. 2003.

[10] O. Watanabe, A. Nakazaki, and H. Kiya, "A fast image-scramble method using public-key encryption allowing backward compatibility with JPEG 2000," In Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 3435-3438, Oct. 2004.

[11] O. Watanabe, A. Uchida, T. Fukuhara, and H. Kiya, "An encryption-then-compression system for JPEG 2000 standard, " In Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.1226-1230, Apr. 2015.

[12] K. Kurihara, S. Shiota, and H. Kiya, "An Encryption-Then-Compression System for JPEG Standard," In Proceedings of Picture Coding Symposium 2015, pp.119-123, Jun. 2015.

[13] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An Encryption-Then-Compression System for JPEG Standard for JPEG/motion JPEG standard," IEICE Transactions on Fundamentals, vol.E98-A, no.11, pp.2238-2245, Nov. 2015.

[14] F. Dufaux, G. Sullivan, and T. Ebrahimi, "The JPEG XR image coding standard [standards in a NUTSHELL]," IEEE Signal Process. Mag., vol.26, no.6, pp.195-199, 204-204, Oct. 2009.