FLEXIBLY-TUNABLE BITCUBE-BASED PERCEPTUAL ENCRYPTION WITHIN JPEG COMPRESSION

Kosuke Shimizu[†] *and Taizo Suzuki*[‡]

[†]Department of Computer Science, University of Tsukuba, Japan
[‡]Faculty of Engineering, Information and Systems, University of Tsukuba, Japan
Email: [†]kshimizu@wmp.cs.tsukuba.ac.jp, [‡]taizo@cs.tsukuba.ac.jp

ABSTRACT

We propose a perceptual encryption within JPEG compression (EWJ). Although some of the conventional EWJs have the 'tunability,' which is a property of how many perceptual degradation levels can be provided with the single encryption technique, it is insufficient because either strong level of security or low bitrate overhead is regarded as important. The proposed EWJ detects the specific subspaces of the bits in the quantized discrete cosine transform (QDCT) coefficient blocks ('bitcubes'), such that the non-zero bits crowd, and then the bits are permuted within each bitcube, i.e., not only within each bitplane like the conventional EWJs but also between adjacent bitplanes. The experiments show that the bitcube-based EWJ actually provides more flexible tunability than the conventional EWJs, while compromising the relation between the bitrate overhead and the attack robustness.

***Index Terms*—** bitplane, flexible tunability, JPEG, perceptual encryption, QDCT domain.

1. INTRODUCTION

Many perceptual encryptions (PEs) that visually protect multimedia contents have been presented [1–8]. Since the PEs can preview the contents in the visually protected state, they can be included in the applications of the subscription-aware broadcasting services (SBSs), e.g., video on demand (VOD) and stock photo, and of the social networking services (SNSs), e.g., Twitter and Facebook. The PEs for those applications should be able to tune the perceptual degradation levels to achieve the various security demands by the providers and customers. This study focuses on the tunable encryption. Hereafter, let 'tunability' be a property of how many perceptual degradation levels can be provided with the single encryption technique.

Most of the PEs are used in conjunction with the international image and video compression standards. Especially, since JPEG [9] is the most popular image compression standard and also inherited as the base layer part of the emerging alternative standard, JPEG XT [10], many PEs for JPEG were proposed [1–8]. The PEs for JPEG are roughly classified in terms of three types of techniques before, after, and within the compression (EBJ, EAJ, and EWJ). The PEs before the JPEG compression (EBJs) [1, 2] are friendly with JPEG, but cannot tune the perceptual degradation levels in each of their encryption modules, i.e., no tunability. As same as the case of EBJs, the PEs after the JPEG compression (EAJs) [3, 4] are also friendly with JPEG, but has no tunability. On the other hand, some PEs within the JPEG compression (EWJs) achieved the tunability. For example, the EWJ by Li and Lo [5] embedded random sign-flips with the alternative transforms for achieving the low bitrate overheads, but it has low tunability. Intra-bitplane shuffling (IBS) proposed by Mao and Wu [6] has the potential to provide the EWJ with flexible tunability, but it significantly increases the bitrate overheads for achieving higher security. In addition, the EWJs by Li and Yuan [7] and by Khan et al. [8] provide a certain level of tunability of the perceptual degradation levels in the encrypted-decoded images thanks to the encryption of several low AC coefficients in each quantized discrete cosine transform (QDCT) coefficient block. However, all of [6–8] are vulnerable against the replacement attack [5], which overwrites the encrypted portion as all zeros to unmask the un-encrypted portion when low compression, i.e., the tunability is insufficient.

We propose a novel EWJ providing more flexible tunability than the conventional EWJs. The proposed EWJ detects the specific subspaces of the bits in the QDCT coefficient blocks ('bitcubes'), such that the non-zero bits crowd, and then the bits are randomly permuted within each bitcube, i.e., not only within a bitplane like the conventional EWJs but also between adjacent bitplanes. The experiments show that the bitcube-based EWJ actually provides flexible tunability than the conventional EWJs, while compromising the relation between the bitrate overhead and the attack robustness.

2. BUILDING BLOCKS AND PRIOR REVIEWS

2.1. JPEG Compression

JPEG is a lossy compression technique consisting of the following five modules:

1. Change the image color space from RGB to YCbCr.
2. Downsample the chroma coefficients Cb and Cr.
3. Apply DCT to each of 8×8 blocks divided in the luma coefficients Y and the chroma coefficients Cb and Cr.
4. Quantize the DCTed coefficients.
5. Encode the coefficients using Huffman coding after differential coding of DC coefficients and run-length coding of AC coefficients in the QDCT domain.

In the step 5, the QDCT coefficients are efficiently compressed because of the sparsity obtained by the steps 1-4.

2.2. Bitplane Decomposition

When c_{ij} and $b_{ij}^{(t)}$ are an (i, j)th decimal QDCT coefficient and the tth ($t \in \mathbb{N}_{[1\ D]}$, $D^1 \in \mathbb{N}$) bit of binary representation from the

[1]It means the number of bits enough to represent the QDCT coefficient range. In the JPEG reference software *libjpeg-turbo* [11], it becomes 10 when the quantization factor Q is less than 96.

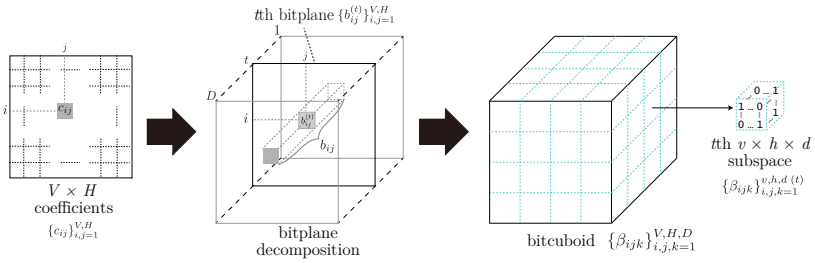


Fig. 1. Bitcuboid and subspace decomposition.

least significant bit (LSB), let $\{c_{ij}\}_{i,j=1}^{V,H}$ and $\left\{\{b_{ij}^{(t)}\}_{i,j=1}^{V,H}\right\}_{t=1}^{D}$ be the $V \times H$ ($V, H \in \mathbb{N}$) coefficients and the D bitplanes composed of a sign-bit as most significant bit (MSB) and $D-1$ bits of absolute part, respectively, i.e., $\{b_{ij}^{(t)}\}_{i,j=1}^{V,H}$ is the tth bitplane. Even if the bitplanes $\left\{\{b_{ij}^{(t)}\}_{i,j=1}^{V,H}\right\}_{t=1}^{D}$ are encrypted, the encrypted binary representation $\widetilde{b}_{ij}$ can be re-decimalized to the encrypted decimal QDCT coefficient $\widetilde{c}_{ij}$ as

$$\widetilde{c}_{ij} = -1^{\widetilde{b}_{ij}^{(D)}} \sum_{t=1}^{D-1} 2^{t-1} \widetilde{b}_{ij}^{(t)}. \tag{1}$$

As an exception, when the encrypted binary representation $\widetilde{b}_{ij}$ is formed as negative zeros, i.e., $-0\cdots0$, we redefine the re-decimalizaed value as

$$\widetilde{c}_{ij} = -2^{D-1}, \tag{2}$$

because the negative zeros are re-decimalized to zero if without (2).

2.3. Conventional Tunable EWJs

This subsection introduces three conventional EWJs [6–8] with the tunability based on QDCT domain.

We first introduce the EWJ derived from applying the IBS by Mao and Wu [6] to each 8×8 QDCT coefficient block in this study. The IBS decomposes the several ($\ell_e \in \mathbb{Z}_{[1\ 64]}$) QDCT coefficients detected in the zig-zag order in each QDCT coefficient block into the bitplanes and randomly permutes the bits within each bitplane. The resulting EWJ has the potential for providing flexible tunability by increasing ℓ_e, setting different conditions in each bitplane, and/or so on. However, it may be vulnerable against the replacement attack, which overwrites the encrypted portion as all zeros to unmask the unencrypted portion when low compression, because fewer coefficients should be encrypted for suppression of the bitrate overheads.

We second introduce the EWJ by Li and Yuan [7]. If the tunability is required, several (ℓ_e) QDCT coefficients are detected in the zig-zag order in each QDCT coefficient block and the coefficients in the same positions among the blocks are randomly permuted. It can suppress the bitrate overheads of the encrypted-encoded bitstreams efficiently unlike one by Mao and Wu, even when using more QDCT coefficients, but may be also vulnerable against the replacement attack.

We third introduce the EWJ by Khan et al. [8]. It decomposes all DC coefficients into the bitplanes and randomly permutes the bits within each of the several ($\ell_{DC} \in \mathbb{N}_{[1\ 7]}$) bitplanes from the LSB plane. In addition, the several ($\ell_{AC} \in \mathbb{N}_{[1\ 5]}$) AC coefficients detected in the zig-zag order in each QDCT coefficient block are randomly permuted among blocks and within each block: the AC coefficients in the same position are permuted among the blocks and the AC coefficients in the different position are permuted within each block. It can suppress the bitrate overheads like one by Li and Yuan, but is also vulnerable against the replacement attack, because the encrypted coefficients are only up to six in 64 coefficients.

From the above, the compromises between the security and the bitrate overheads should be considered. The bitrate overhead problem of the conventional EWJ by Mao and Wu is because of significantly different number of the non-zero coefficients after encryption compared with the original one. Also, the security problem of the conventional EWJs by Li and Yuan and by Khan et al. is because of that the number of non-zero bits to be encrypted per each block is not enough.

3. BITCUBE-BASED EWJ

We propose a novel EWJ, which encrypts the non-zero bits without changing the number of the zero bits, i.e., the sparsity, in each block much from the original ones, by extending the IBS by Mao and Wu [6].

3.1. Definitions of Bitcuboid and Bitcube

This subsection introduces a more general representation of bitplane and its subspace representation. The cuboid containing bits ('bitcuboid') is constructed by bundling the bitplanes from the MSB plane to the LSB plane as shown in Fig. 1. The set of bitplanes $\left\{\{b_{ij}^{(t)}\}_{i,j=1}^{V,H}\right\}_{t=1}^{D}$ are redefined as the $V \times H \times D$ bitcuboid $\{\beta_{ijk}\}_{i,j,k=1}^{V,H,D}$ as

$$\{\beta_{ijk}\}_{i,j,k=1}^{V,H,D} = \left\{\{b_{ij}^{(t)}\}_{i,j=1}^{V,H}\right\}_{t=1}^{D}. \tag{3}$$

Constructing the bitcuboid, the encryptor can move the bits in the bitcuboid not only within a bitplane but also between adjacent bitplanes.

However, the encryptor should move the bits in the smaller subspaces of the bitcuboid because the encrypted non-zero coefficients cause the bitrate overheads when the sparsity is lost. Therefore, the bitcuboid is divided to several $v \times h \times d$ ($\geq 2;\ v \in \mathbb{N}_{[1\ V]}, h \in \mathbb{N}_{[1\ H]}, d \in \mathbb{N}_{[1\ D]}$) subspaces. When the $\{\beta_{ijk}\}_{i,j,k=1}^{V,H,D}$ is divided to the L subspaces not overlapping mutually, the subspaces are denoted as

$$\left\{\{\beta_{ijk}\}_{i,j,k=1}^{v,h,d\ (t)}\right\}_{t=1}^{L} = \{\beta_{ijk}\}_{i,j,k=1}^{V,H,D}, \tag{4}$$

where $L = \lfloor V/v \rfloor \times \lfloor H/h \rfloor \times \lfloor D/d \rfloor$. The $\{\beta_{ijk}\}_{i,j,k=1}^{v,h,d\ (t)}$ is the tth subspace of the $\{\beta_{ijk}\}_{i,j,k=1}^{V,H,D}$ and contains the $v \times h \times d$ bits

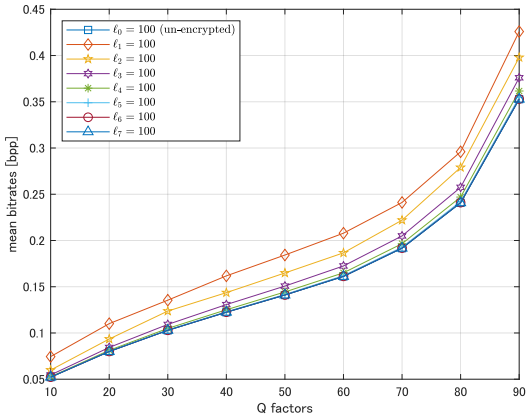


Fig. 2. Comparison of bitrate overheads induced with the various m-cubes.

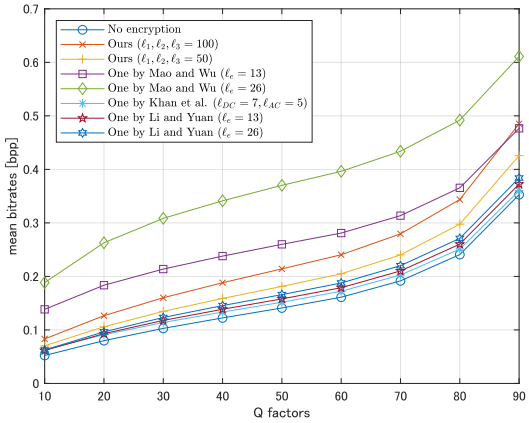


Fig. 3. Comparison of bitrate overhead suppressions with the EWJs.

that pass through the d bit sequences of the corresponding $v \times h$ coefficients in the $\left\{\{b_{ij}^{(t)}\}_{i,j=1}^{V,H}\right\}_{t=1}^{D}$. For the JPEG application, a bitcuboid $\{\beta_{ijk}\}_{i,j,k=1}^{8,8,D}$ is constructed from the set of bitplanes $\left\{\{b_{ij}^{(t)}\}_{i,j=1}^{8,8}\right\}_{t=1}^{D}$ except for the MSB plane (sign-bit plane) and LSB plane, because each 8×8 QDCT block has 8×8 coefficients in D-bit depth. Also, when $v = h = d$, the subspace is termed as a 'bitcube.'

3.2. Encryption Algorithm

There are several possible ways to encrypt the subspaces, but this study introduces a scrambling approach randomly permuting the bits within each bitcube. The encrypted subspace $\{\widetilde{a}_i\}_{i=1}^{P}$ ($P = v \times h \times d$) is obtained by randomly permuting the bits in the original subspace $\{a_i\}_{i=1}^{P}$ as

$$\{\widetilde{a}_i\}_{i=1}^{P} = \{\odot\,(a_i)\}_{i=1}^{P},$$
$$\odot : x \to y\ (x \neq y; x, y \in \{a_i\}_{i=1}^{P}) \tag{5}$$

with the random numbers calculated from the pseudo random number generator initialized with the seeds extracted from the encryption key long enough. The number of the non-zero bits (ones) of the subspace does not change even if permuted. Moreover, by selecting the

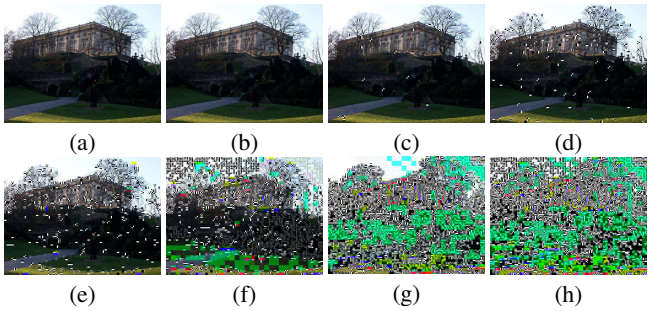


Fig. 4. Comparison of the perceptual degradations produced by the various types of m-cubes (JPEG $Q = 70$): (a) all 7-cubes ($\ell_7 = 100$), (b) all 6-cubes ($\ell_6 = 100$), (c) all 5-cubes ($\ell_5 = 100$), (d) all 4-cubes ($\ell_4 = 100$), (e) all 3-cubes ($\ell_3 = 100$), (f) all 2-cubes ($\ell_2 = 100$), (g) all 1-cubes ($\ell_1 = 100$), and (h) all $\{1, 2, 3\}$-cubes ($\ell_1, \ell_2, \ell_3 = 100$).

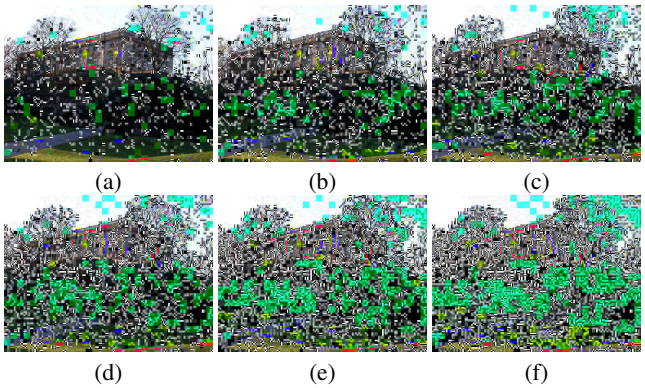


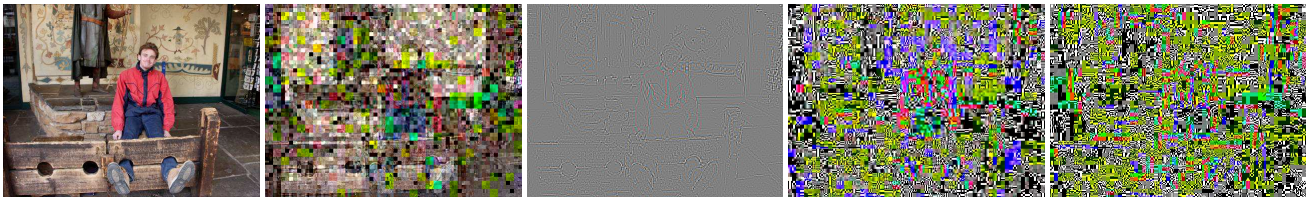
Fig. 5. Comparison of various perceptual degradation levels with the various percentages for encrypting 1-cubes (JPEG Q = 70): (a) 20% ($\ell_1 = 20$), (b) 40% ($\ell_1 = 40$), (c) 50% ($\ell_1 = 50$), (d) 60 % ($\ell_1 = 60$), (e) 80% ($\ell_1 = 80$), and (f) 100% ($\ell_1 = 100$).

specific subspaces to be encrypted carefully, we can achieve the flexible tunability in each bitcuboid. The subspaces containing the very few number of non-zero bits must be sparse to preserve the sparsity of the QDCT domain, i.e., the coding performance.

Hereafter, this study considers the subspace as the bitcube for simplicity and let the bitcube whose number of ones is m ($\in \mathbb{Z}_{[0\ P]}$) be 'm-cube.' We can take the advantages of the percentages that how many the m-cubes are encrypted to achieve the flexible tunability. The ratio of encrypting m-cubes for all bitcubes of bitcuboids ($\ell_m \in \mathbb{Z}_{[0\ 100]}$) contributes to the tunability. When we select the T ($\leq P$) types of m-cubes efficient for the encryption, 100^T ways are enable to achieve the tunability of the encrypted-decoded images.

3.3. Security Analysis

For the security analysis, we first measure the number of the brute-force attacks, under the condition that the attacker tries to reconstruct the all encrypted bitcubes in a bitcuboid (block) without understanding that which m-cube was encrypted and the encryption key. The number of the brute-force attacks to reconstruct each bitcube in a block is 2^P and the number of the m-cubes contained in a bitcuboid is l. Consequently, the number of the brute-force attacks



(a) (b) (c) (d) (e)

Fig. 6. Attacked-decoded images after the replacement attacks (JPEG $Q = 70$): (left-to-right) original one, encrypted one by Khan et al. ($\ell_{DC} = 7, \ell_{AC} = 5$), attacked encrypted one by Khan et al., encrypted one by ours ($\ell_1, \ell_2, \ell_3 = 100$), and attacked encrypted one by ours.

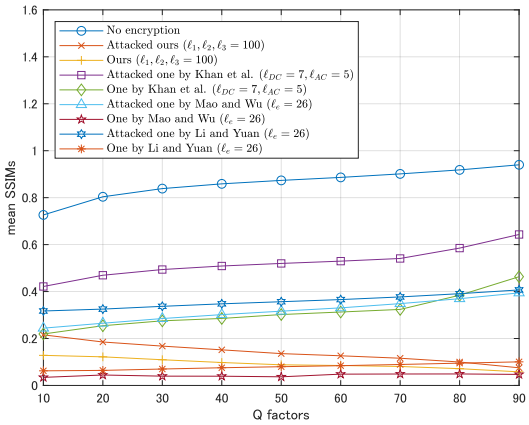


Fig. 7. Comparison of mean SSIMs between the encrypted-decode images and the original images.

to all bitcubes in a block is

$$(2^P)^l = 2^{Pl}. \tag{6}$$

For example, when the number in the case of $2 \times 2 \times 2$ bitcubes is $(2^8)^l = 2^{8l}$ and l is greater than 32, the number of the brute-force attacks against one bitcuboid is greater than the number of the brute-force attacks against the SHA-256 hash digest [12, 13], which is regarded as a security criterion. Even if l is less than or equal to 32, encrypting the multiple blocks can reinforce the robustness of the encrypted region.

We second discuss the robustness against the replacement attack [5]. When the bitcubes of the 1-cubes to the $(P-1)$-cubes ($\{1, \cdots, P-1\}$-cubes) are encrypted, the remain un-encrypted bitcubes in a bitcuboid are 0-cubes and P-cubes. If all of the $\{1, \cdots, P-1\}$-cubes are overwritten to the 0-cubes and the P-cubes are hardly existing, the bits representing the original features of the block are also hardly existing. Therefore, the replacement attacks cannot sufficiently recover the original features of each encrypted block and we can consider that the bitcube-based EWJ is robust against the replacement attack.

4. EXPERIMENTS

We compared the effectiveness of our bitcube-based EWJ with the conventional EWJs as following procedures:

1. Encode the test image with the *libjpeg-turbo* [11] while encrypting the QDCT coefficients using the EWJ (JPEG quality factor $Q = 10, 20, \cdots, 90$)
2. Measure the bitrates of the encrypted and not encrypted bitstreams.
3. Decode them.
4. Measure the SSIMs [14] of the decoded images.
5. Iterate from 1. to 4. for all test images and calculate the mean results.

We set the bitcube size to $v = h = d = 2$ and used the 100 full-color images of the Un-Compressed Image Dataset (UCID) database [15] as the test images. One can see that the bitcubes with $\ell_1 = 100$ caused the most bitrate overheads compared with the other bitcubes with $\ell_2 = \ell_3 = \cdots = \ell_7 = 100$ as shown in Fig. 2. Also, although the bitcube-based EWJ could not suppress more than ones by Li and Yuan and by Khan et al., it could suppress more of the bitrate overheads than one by Mao and Wu [6] as shown in Fig. 3.

We also compared the perceptual degradation levels of our encrypted-decoded images at the various m-cubes. The bitcubes with $\ell_1 = 100$ induced the strongest perceptual degradation into the encrypted-decoded image and the combination of the bitcubes with $\ell_1, \ell_2, \ell_3 = 100$ as shown in Fig. 4 (h) induced much stronger degradation than the other patterns as shown in Fig. 4 (a-g). Besides, according to the increase of the percentage (the value of ℓ_m), the bitcube-based EWJ could flexibly tune the perceptual degradation levels as shown in Fig. 5.

The images, whose all 1-cubes, all 2-cubes, and all 3-cubes were encrypted by the bitcube-based EWJ, are obviously robust against the replacement attack, because the encrypted image by Khan et al. was recovered to the rough textures that can be previewed but one by ours was not recovered as shown in Fig. 6. We also confirmed the objective robustness with the mean SSIMs between the encrypted-decoded images and the original images as shown in Fig. 7. It it clear that the attacked encrypted-decoded images by ours were not recovered the visual qualities with the replacement attack unlike three conventional EWJs.

5. CONCLUSION

We proposed a novel EWJ providing flexible tunability. The proposed bitcube-based EWJ was placed into the JPEG QDCT domain and randomly permuted the bits of the bitcubes in each of the QDCT coefficient blocks. The experiments showed that the bitcube-based EWJ actually provided more flexible tunability than the conventional EWJs, while compromising the relation between the bitrate overhead and the security.

6. REFERENCES

[1] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for JPEG/Motion JPEG standard," *IEICE Trans. Fundamentals*, vol. E98-A, no. 11, pp. 2238–2245, Nov. 2015.

[2] T. Chuman, W. Sirichotedumrong, and H. Kiya, "Encryption then-compression systems using grayscale-based image encryption for JPEG images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1515–1525, Nov. 2019.

[3] V. Itier, P. Puteaux, and W. Puech, "Recompression of JPEG crypto-compressed images without a key," *IEEE Trans. Circuits Syst. Video Technol.*, Jan. 2019, Early Access.

[4] J. Ting, K. Wong, and S. Ong, "Format-compliant perceptual encryption method for JPEG XT," in *Proc. of ICIP'19*, Taipei, Taiwan, Sept. 2018, pp. 4559–4563.

[5] P. Li and K.-T. Lo, "Joint image compression and encryption based on order-8 alternating transforms," *J. Vis. Commun. Image*, vol. R, no. 44, pp. 61–71, Apr. 2017.

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

[7] W. Li and Y. Yuan, "A leak and its remedy in JPEG image encryption," *Int. J. Comput. Math*, vol. 84, no. 9, pp. 1367–1378, Sept. 2007.

[8] 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," in *Proc. of ICIAS'10*, Kuala Lumpur, Malaysia, June 2010, pp. 1–6.

[9] "Overview of JPEG," https://jpeg.org/jpeg.

[10] "Overview of JPEG XT," https://jpeg.org/jpegxt.

[11] "JPEG software," https://jpeg.org/jpeg/software.html.

[12] K. Dmitry, R. Christian, and S. Alexandra, "Bicliques for preimages: Attacks on Skein-512 and the SHA-2 family," in *Proc. FSE'19*, Washington, DC, Mar. 2012, pp. 244–263.

[13] National Institute of Standards and Technology, *FIPS PUB 180-4: Secure Hash Standard (SHS)*, pub-NIST, Aug. 2015.

[14] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[15] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," in *Proc. SPIE 5307, Storage and Retrieval Methods and Applications for Multimedia*, San Jose, CA, Jan. 2004, pp. 472–480.