Finely Tunable Bitcuboid-Based Encryption With Exception-Free Signed Binarization for JPEG Standard

Kosuke Shimizu and Taizo Suzuki, *Senior Member, IEEE*

***Abstract*— We propose a finely tunable JPEG format-compliant perceptual encryption (FE) with two novel strategies: (i) bitcuboid-based encryption (BE) and (ii) exception-free signed binarization (ESB). BE is an intra- and inter-bitplane encryption technique that provides finely tunable perceptual degradation by encrypting constrained subspaces ('bitcubes') of a cuboid-shaped bit set ('bitcuboid'). ESB is a binarization technique that redecimalizes encrypted binary sequences into signed decimal coefficients without any exception-handling by shifting the negative binary sequences one-by-one. BE with ESB (BEESB) is applied to the quantized discrete cosine transform (QDCT) domain in JPEG compression. The results of our first experiment show that the BE attains fine tunability, which means scalability of the perceptual degradation level with a single encryption method, by encrypting bitcubes of various types and sizes combinatorially. The results of our second experiment show that the BEESB suppresses the bitrate overheads and that BEESB with one of the most secure options suppresses approximately 0.80-187.58 % more of the bitrate overheads in terms of Bjøntegaard delta (BD)-rate compared with conventional methods except for some ones. The results of our third experiment show that BEESB has high resilience against attacks.**

***Index Terms*— Bitcuboid-Based encryption, exception-free signed binarization, format-compliant encryption, JPEG, tunable encryption, tunability.**

I. Introduction

PERCEPTUAL encryption (PE) is an encryption technique that visually protects image and video content. Many format-compliant PEs (FEs)[1] produce encrypted-encoded content in the same format as the original for image and video coding standards such as the JPEG and H.26x series [1], [2]. The existing decoders can display encrypted content with imperceptible changes in visual quality without any modification or any decryption key, unlike format-noncompliant encryptions such as the pure Advanced Encryption Standard (AES) [3][2] and the compressed sensing-based approach [5]. FEs have extensive applications in fields such as social networking services (SNSs), including Twitter and Facebook, and subscription-aware broadcasting services (SBSs), including video on demand (VOD) and stock photo. In such applications, the desired level of perceptual degradation provided with the FEs are different, and they are classified into a range from transparent encryption [6] to confidentiality encryption [7]. To meet various security demands, any FE method should be able to tune the perceptual degradation level, and thus many tunable encryptions have been proposed in the past two decade. The tunable FEs proposed for AVC and HEVC [8]–[10] handle the specific coded elements (e.g., intra prediction modes and motion vector differences) among the multiple video frames using the parameter that specifies how many number of them is encrypted. However, the video stream-specific syntax elements cannot be exploited to the tunable encryption for JPEG file interchange format (JFIF) syntax domain, despite the JPEG is a de-facto image coding standard. Although many JPEG FEs have been presented [11]–[24], they have not paid attention to 'tunability,' which means the scalability of perceptual degradation level achieved with a single FE method in this study. They tend to cherish that distortions generated with decoding and decrypting after encrypting and encoding are completely the same as ones generated without encrypting and decrypting and that the bitrate overheads of the encrypted-encoded content are small or zero compared with the normally encoded ones.

In the past decade, the JPEG FEs in the pre- and post-compression domain have been the focus of attention because these FEs hardly or moderately affect the coding efficiency [11]–[18]. The FEs that work before JPEG (FBJs) [11]–[13] apply four encryption modules (scrambling, rotation/inversion, nega-posi transformation, and component shuffling) to the original image or multiple video frames (or three encryption modules except for component shuffling to a grayscale image [14]) to achieve high confidentiality and good or moderate compression friendliness.

Manuscript received March 25, 2021; revised July 15, 2021; accepted August 27, 2021. Date of publication September 16, 2021; date of current version October 25, 2021. This work was supported by Grant-in-Aid for JSPS Fellows, Grant Number 20J14599. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lejla Batina. *(Corresponding author: Kosuke Shimizu.)*

Kosuke Shimizu is with the Department of Computer Science, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan (e-mail: kshimizu@wmp.cs.tsukuba.ac.jp).

Taizo Suzuki is with the Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan (e-mail: taizo@cs.tsukuba.ac.jp).

Digital Object Identifier 10.1109/TIFS.2021.3113510

[1] Although the cryptosystems are classified into symmetric and asymmetric ones, this study focuses on the symmetric ones, which are more suitable for "image" encryption, not the asymmetric ones commonly used for "key" encryption, because the latter requires relatively high computational complexity while involving the risks to the man-in-the-middle attacks to be solved by the signature solutions.

[2] AES can be part of a format-compliant encryption as presented in [4].

TABLE I
ADVANTAGES OF JPEG FES: (I) TUNABILITY, (II) BITRATE OVERHEAD, AND (III) ATTACK RESILIENCE

methods	(I)	(II)	(III)
FBJs [11]–[14]	no	**low or moderate**	**high**
FAJs [15]–[18]	no	**low**	low or high
RANDZZ (FWJ) [19]	no	high	**high**
IBS (FWJ) [20]	potentially certain	high	**high**
FIBS (FWJ) [21]	potentially certain	**low**	**high**
DCACS (FWJ) [22]	certain	**low**	low
RSF (FWJ) [24]	less	**low**	low
Proposed BEESB (FWJ)	**fine**	**low or moderate**	**high**

The FEs that work after JPEG (FAJs) [15]–[18] apply XOR operations or permutations to specific syntax elements of the compressed bitstream to achieve different types of perceptual degradation, e.g., luma-only degradation and DC-and-AC degradation. However, the FBJs and FAJs have no or less tunability.

The older JPEG FEs had been designed within the codec [19]–[24]. The FE that works within JPEG (FWJ) using randomized zig-zag scan (RANDZZ) [19][3] is resilient to the replacement attack, which overwrites the encrypted portions with a constant value (e.g., 0) to disclose the other unencrypted portions at the decoder stage, at the expense of significant bitrate overheads. Also, the FWJ embedding random sign-flips (RSF) [23], [24][4] suppresses the bitrate overheads, at the expense of resilience against the replacement attack. However, the RANDZZ and RSF have no or less tunability. On the other hand, some FWJs [20]–[22] have the potential to achieve some level of tunability. An FWJ using intra-bitplane shuffling (IBS) [20] achieves a certain level of tunability for resilience against the replacement attack at the expense of bitrate overheads. Moreover, an FWJ using full inter-block shuffling (FIBS) [21] and one using inter-block DC bitplane shuffling and intra/inter-block AC coefficient shuffling (DCACS) [22] achieve a certain level of tunability and suppress the bitrate overheads efficiently. However, they are still vulnerable against the replacement attacks. On the other hand, the FWJs are resilient to the sketch attacks [16], which reveal the global outline information of images, because they protect the blockwise detailed local texture information. For the same reason, the FWJs are commonly resilient to the jigsaw puzzle solver attack [25], which connects the unencrypted texture information. So far, we introduced the conventional FWJs, but they have not received much attention recently because they affect coding efficiency and are vulnerable to attacks as described above. Nevertheless, carefully encrypting the partially selected bits in the quantized discrete cosine transform (QDCT) domain in accordance with the old methods, we found that our FWJ can attain the fine level of tunability for JPEG while preserving the coding efficiency and attack resilience as described below.

[3]Although this method was originally for MPEG, we can easily apply it to JPEG.

[4]Although the newer approach uses alternative transforms, it corresponds to randomly flip the signs of transform coefficients, as stated in [23].

We propose a finely tunable FWJ, which contributes to meeting more security demands on image and video content than the existing methods while suppressing the bitrate overheads and sustaining the resilience against attacks, as shown in Table I. The proposed FWJ consists of the following novel strategies: (i) bitcuboid-based encryption (BE) and (ii) exception-free signed binarization (ESB). BE is an IBS-inspired encryption technique with the cuboid-shaped bit set ('bitcuboid') of the QDCT coefficient block. It consists of intra- and inter-bitplane encryptions in binary spaces, which are constrained by subdividing the bitcuboid into several smaller subsets ('bitcubes'). ESB is a binarization technique that redecimalizes a binarized sequence losslessly without any exception-handling to suppress the bitrate overheads. Experiments with encryption in JPEG show that the proposed BE with ESB (BEESB) achieves finer tunability in terms of the sizes of the bitcubes and their encryption percentages, more efficient bitrate overhead suppression of the encrypted-encoded content compared with conventional methods, and high resilience against attacks.

We presented a constrained BE in our preliminary study [26]: the size of the encrypted bitcube was constrained by using nonoverlapping $2 \times 2 \times 2$ (2^3) subsets. Moreover, the previous study used a fixed-length signed binarization (FSB), whose exception-handling increases the bitrate overheads of the encrypted-encoded content. This study unconstrains the size of the bitcube to achieve finer tunability and proposes ESB to further suppress the bitrate overheads.

The remainder of this paper is organized as follows: Section II describes JPEG and FSB. Section III describes BEESB. Section IV discusses the experiments aimed at assessing the fine tunability, compression efficiency, and resilience to attacks of our method compared with the conventional FWJs. Section V summarizes this paper.

II. REVIEW AND DEFINITIONS

A. JPEG

The JPEG encoder converts a large (noncompressed) original image into a compact size of compressed bitstream. The encoding procedure is as follows:

a) Apply an RGB-to-YCbCr color transform to an input image and then downsample the Cb and Cr planes often to quarters of the original spatial sizes.

b) Apply a DCT to each 8×8 nonoverlapping block of the Y, Cb, and Cr planes and quantize them. Each resulting QDCT coefficient block consists of one DC coefficient and 63 AC coefficients.[5]
c) Apply differential pulse-code modulation (DPCM) and run-length encoding (RLE) to the DC and AC coefficients, further compact them with Huffman coding, and packetize them byte-wisely.

On the other hand, the JPEG decoder reconstructs the image from the encoded bitstream by performing the inverse processing of the above JPEG encoding. Note that the reconstructed image is distorted by the downsampling and quantization.

B. Fixed-Length Signed Binarization

By partially encrypting the binarized space, the conventional FWJs [20], [22] can provide a certain level of tunability to the encrypted-decoded images. A frequently-used binarization technique for the conventional FWJs is D-bit ($D \in \mathbb{N}$) FSB of a signed decimal number c to an n-bit binary sequence $\mathbf{b}$, as

$$\mathbf{b} = \mathrm{sgn}(c)|(\mathrm{abs}(c))_{2,n-1}, \tag{1}$$

$$\mathrm{sgn}(x) = \begin{cases} 0 & \text{if } x \geq 0 \\ 1 & \text{otherwise} \end{cases}, \tag{2}$$

$$\mathrm{abs}(x) = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{otherwise} \end{cases}, \tag{3}$$

$$(x)_{2,\eta} = b_{\eta-1} \cdots b_0, \tag{4}$$

where $b_i \in \{0, 1\}$, which is used in the coding parts of JPEG2000 EBCOT [28], HEVC CABAC [29], etc; here, (4) is an well-known decimal-to-binary conversion that generates an η-bit fixed-length binary sequence. When a binary sequence is encrypted before any variable length coding, which encodes decimal numbers, such as Huffman coding, the encrypted binary sequence should be redecimalized for the encoding. However, when the encrypted binary sequence consists of one negative sign bit and $D-1$ all-zero absolute bits, as

$$\underbrace{1}_{\substack{\text{negative} \\ \text{sign bit}}} \; | \; \underbrace{0 \cdots 0}_{\substack{D-1 \\ \text{absolute bits}}}, \tag{5}$$

the encrypted coefficients cannot be decrypted in the decoder stage because the binary sequence in (5) is redecimalized as $-1 \times 0 = 0$; i.e., the negative sign bit in (5) is deleted with a normal redecimalization converting $\mathbf{b}$ to c, as

$$c = (-1)^{b_{n-1}} \left(\sum_{i=0}^{n-2} b_i \times 2^i \right). \tag{6}$$

One of the strategies for handling this exception is to exceptionally redecimalize the binary sequence in (5) into -2^{D-1}.

[5]In the case of the JPEG reference software *libjpeg* [27], the QDCT coefficients can be stored as 10-bit signed variables when the quality factor Q is less than 96.

TABLE II
FSB AND ESB BINARY SEQUENCES (A SIGN BIT | $D-1$ ABSOLUTE BITS) OF THE CORRESPONDING SIGNED DECIMAL NUMBERS IN THE CASE OF $D = 10$

decimal number	FSB binary sequence	ESB binary sequence
511	0|111111111	0|111111111
510	0|111111110	0|111111110
509	0|111111101	0|111111101
⋮	⋮	⋮
3	0|000000011	0|000000011
2	0|000000010	0|000000010
1	0|000000001	0|000000001
0	0|000000000	0|000000000
−1	1|000000001	1|000000000†
−2	1|000000010	1|000000001
−3	1|000000011	1|000000010
⋮	⋮	⋮
−510	1|111111110	1|111111101
−511	1|111111111	1|111111110
−512	1|000000000*	1|111111111

*Irregular sequence to be redecimalized to -2^{D-1}
†Moved irregular sequence to be redecimalized to -1

In the decryption stage, the exceptionally-redecimalized coefficient -2^{D-1} is exception-freely binarized to

$$\underbrace{1}_{\substack{\text{negative} \\ \text{sign bit}}} \; | \; \underbrace{\not{1}}_{\substack{\text{one highest} \\ \text{absolute bit} \\ \text{to be ignored}}} \; \underbrace{0 \cdots 0}_{\substack{D-1 \\ \text{absolute bits}}} \Rightarrow 1|0 \cdots 0, \tag{7}$$

by one-time sign-extraction and $(D-1)$-times absolute-extraction in (1) so that we obtain the same binary sequence as in the encryption stage and decrypt it completely.

However, we must note that the above exception has a negative effect on the JPEG compression: since it is clear that the redecimalized irregular value -2^{D-1} increases the Huffman code length and Huffman table size, it affects the bitrate overhead of the encrypted-encoded bitstream. Moreover, this exception easily occurs because many encrypted binary sequences have many 0 absolute bits because of the JPEG characteristic, which exploits many zero bits (sparsity) in the binarized QDCT coefficients. For example, when the least significant bits (LSBs) of the binary sequences $(1|000000001, 0|000000000)$ that have been binarized from their decimal versions $(-1, 0)$ are replaced with $(1|000000000, 0|000000001)$, they are redecimalized to $(-512, 1)$, as shown in the second column of Table II.

III. BITCUBOID-BASED ENCRYPTION WITH EXCEPTION-FREE SIGNED BINARIZATION

A. Strategy

JPEG's entropy coding attains a more efficient compression when the QDCT domain is sparser, whereas bitwise encryptions tend to make the domain denser without taking any measure. Among them, inter-coefficient bit-shuffling may generate nonzero coefficients, whose number is equivalent to the total number of nonzero bits (i.e., 1s) contained in the

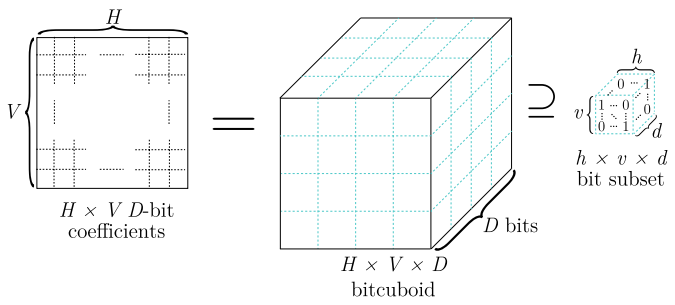

Fig. 1. Bitcuboid and its bit subset.

coefficient set. For instance, if the bits within an n-bit binary sequence containing m nonzero bits

$$\mathbf{b}_0 := b_{n-1} \cdots b_0 \text{ where } \sum_{i=0}^{n-1} b_i = m \tag{8}$$

are shuffled among $m+1$ sequences $\{\mathbf{b}_t\}_{t=0}^{m}$ ($\mathbf{b}_1, \cdots, \mathbf{b}_m = \mathbf{0}$), any m sequences of the $\{\mathbf{b}_t\}_{t=0}^{m}$ may contain nonzero bits and be converted to nonzero coefficients with redecimalization. However, as described in Section II-B, increasing the number of the nonzero coefficients affects the compression efficiency. The bitwise fine shuffling allows for fine tunability in the encrypted-decoded images, but to maintain the compression efficiency, the number of nonzero coefficients should be the same as before the encryption. On the other hand, if nonencrypted nonzero bits representing the original image features remain, these features may be extracted by removing some of the other encrypted bits. Therefore, to be able to resist attacks and maintain compression efficiency at the same time, this study randomly shuffles the nonzero bits in binary subspaces which are smaller subsets of the binary space.

B. Bitcuboid-Based Encryption

BE is a method that changes an $H \times V$ D-bit ($H, V, D \in \mathbb{N}$) decimal coefficient block into an $H \times V \times D$ cuboid-shaped bit set ('bitcuboid') and then randomly shuffles the bitwise relationships in each subdivided $h \times v \times d$ (≥ 2; $h \leq H, v \leq V, d \leq D$) bit set, as shown in Fig. 1. Although the bitcuboid and its subsets can be set to the above arbitrary sizes, this study sets $H = V = 8$ and $D = 10$ and $h = v = d$ ('bitcube') for simplicity and effective JPEG compression. Let 'm-cube' be the h^3 bitcube containing m nonzero bits as follows:

$$\mathbf{B}_m := b_{h^3-1} \cdots b_0 \text{ where } \sum_{i=0}^{h^3-1} b_i = m. \tag{9}$$

The encrypted m-cube $\mathbf{B}'_m$ is given by

$$\mathbf{B}'_m = \mathbf{B}_m \mathscr{P}_{h^3}, \tag{10}$$

where $\mathscr{P}_{h^3}$ is an $h^3 \times h^3$ random permutation matrix calculated by using a pseudo-random number generator (PRNG) [30].

Since the bitwise relationships can be changed according to the internal number of bits in each bitcube, BE provides fine tunability of the encrypted-decoded images. In addition, it does not affect the compression efficiency significantly. The encrypted m-cube $\mathbf{B}'_m$ also satisfies $\sum_{i=0}^{h^3-1} b_i = m$, as does the nonencrypted one $\mathbf{B}_m$ in (9). Since the QDCT domain is composed of quite sparse coefficients, even if the BE converts a zero coefficient into a nonzero one, the corresponding one tends to be converted into a zero coefficient; i.e., the zero coefficient ratio, which affects the compression efficiency, is hardly changed. The actual effectiveness is proved in Section IV-B.

C. Exception-Free Signed Binarization

BE requires a binarization and a redecimalization before and after the actual encryption/decryption, as follows:

- To bitwise encrypt/decrypt the QDCT coefficients, they must be binarized before the encryption/decryption.
- To easily input the encrypted/decrypted QDCT coefficients to the subsequent processing, they must be redecimalized after the encryption/decryption.

However, in Section II-B, we problematized that a simple BE with FSB (BEFSB) presented in [26] needs an exception-handling in the redecimalization process. To resolve the exception-handling problem, we propose ESB.

ESB binarizes a signed decimal number c into an n-bit binary sequence $\mathbf{b}$ as

$$\mathbf{b} = \mathrm{sgn}(c) | \left(\mathrm{abs}(c) - \mathrm{sgn}(c)\right)_{2,n-1}, \tag{11}$$

The subtraction of the binarized sign $\mathrm{sgn}(c)$ in (11) implies that negative binary sequences are shifted one-by-one. $\mathbf{b}$ is redecimalized to the original c, as

$$c = (-1)^{b_{n-1}} \left(\left(\sum_{i=0}^{n-2} b_i \times 2^i \right) + b_{n-1} \right). \tag{12}$$

By adding the binarized sign $b_{n-1} = \mathrm{sgn}(c)$ subtracted in (11) to the redecimalized absolute value $\sum_{i=0}^{n-2} b_i \times 2^i$ in (12), $\mathbf{b}$ can be redecimalized completely to a signed decimal number c without any exception-handling.

Since ESB redecimalizes the binary sequence $1|0 \cdots 0$ to the signed negative -1 as shown in the third column of Table II, the redecimalized number is packetized as the shortest length of negative binary amplitude in the entropy coded segment (ECS) of the JFIF syntax structure [31]. Therefore, ESB can suppress the adverse effects on the compression efficiency better than the simple FSB does.

Remark: Since ESB is a simple binarization algorithm, it can also be easily applied to certain conventional FWJs, e.g., RANDZZ [19], IBS [20], and DCACS [22].

D. Encryption and Decryption Algorithms

We apply BEESB to JPEG QDCT domain after initializing the 19936-bit state vector of Mersenne twister (MT) with 256-bit SHA-2 digest as shown in Fig. 2. In the QDCT domain, we apply the ESB binarization, core encryption/decryption (permutation) of the detected m-cubes using (10) and MT random numbers, and ESB redecimalization for each 8×8 coefficient block. The encryption and decryption are exactly the same because the bits encrypted with the permutation can be decrypted with the same permutation based on the same encryption key.

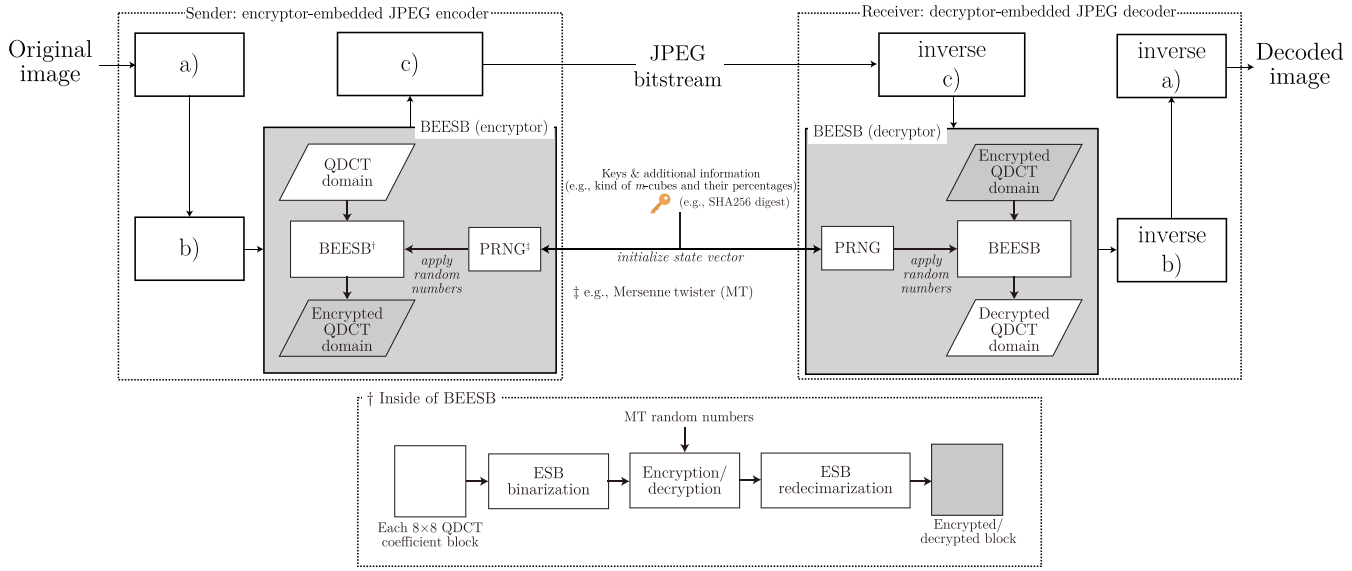

Fig. 2. Bitcuboid-based encryption and decryption embedded in the JPEG encoding and decoding; here, a), b), and c) correspond to the ones of section II-A.

Fig. 3. Perceptual degradation induced by different types of 2^3 m-cubes in BEESB (JPEG $Q = 70$): (a) particular areas of nonencrypted *ucid00059*, (b) $\ell_1^{[2]} = 100$, (c) $\ell_2^{[2]} = 100$, and (d) $\ell_3^{[2]} = 100$.

TABLE III

BD METRICS OF BEESBS AND THE CONVENTIONAL METHODS IN THE CASE OF UCID DATASET [32]: (A) BEESB ($\ell^{[2]}_{1,\cdots,7} = 100$), (B) BEESB ($\ell^{[3]}_{1,\cdots,26} = 100$), (C) BEESB ($\ell^{[4]}_{1,\cdots,63} = 100$), (D) RANDZZ [19], (E) IBS [20] ($\ell_e = 64$), (F) FIBS [21] ($\ell_e = 64$), (G) DCACS [22] ($\ell_{DC} = 7, \ell_{AC} = 5$), (H) RSF [24], AND (I) GBE (FBJ) [14]

Methods	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
mean BD-PSNR [dB]	−0.90	−1.96	−1.87	−4.23	−4.81	−0.93	−0.31	**−0.21**	−1.79
mean BD-rate [%]	19.19	44.18	42.80	106.44	129.22	19.99	6.22	**4.25**	41.42

Fig. 4. Perceptual degradation induced by all 2^3 m-cube types in the full BEFSB and BEESB (JPEG $Q = 70$ and $\ell^{[2]}_{1,\cdots,7} = 100$): (a) BEFSB and (b) BEESB.

E. Fine Tunability

BEESB achieves fine tunability by encrypting only specific kinds of bitcubes, as follows:

- Encrypt arbitrary types of m-cubes ($m \leq h^3 - 1$) combinatorially.
- Encrypt $\ell_m (\in \mathbb{Z}_{[0\ 100]})$ % of all m-cubes contained in all bitcuboids.

Encrypting the l types of m-cubes with the above two tunings allows for

$$\mathcal{L} = \prod_{i=1}^{l} \max \ell_{m_i} = 100^l \quad (m_i \in \mathbb{Z}_{[0\ h^3]},\ l \leq h^3 - 1) \tag{13}$$

ways of tunability. Even m-cubes that are contained in only a few bitcuboids provide some tunability.

Remarks: There are many other self-evident ways of achieving tunability, for example, different selections of the ℓ_m per bitcuboid, random construction of the bitcubes, alteration of the way to encrypt bitcubes, and so on.

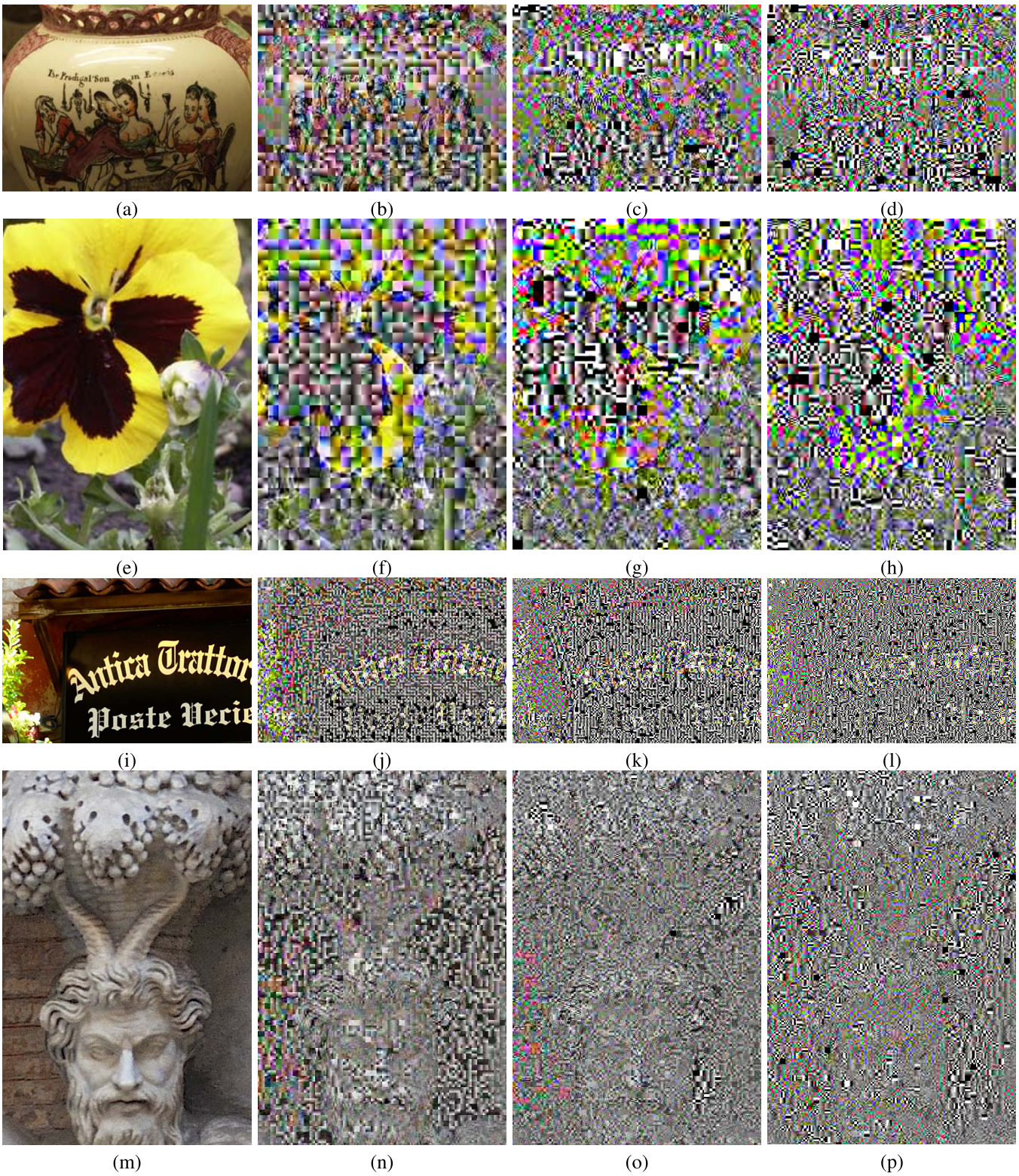

Fig. 5. Perceptual degradation induced by m-cubes of different sizes in BEESB (JPEG $Q = 70$): (a-d) particular areas of *ucid00059*, (e-h) particular areas of *ucid00069*, (i-l) particular areas of *r00444b95t*, (m-p) particular areas of *r00b8d4a2t*, (a,e,i,m) nonencrypted, (b,f,j,n) $\ell^{[2]}_{1,\cdots,7} = 100$, (c,g,k,o) $\ell^{[3]}_{1,\cdots,26} = 100$, and (d,h,l,p) $\ell^{[4]}_{1,\cdots,63} = 100$.

F. Resilience to Attacks

There are several ciphertext-only attacks (COAs), such as replacement attack, brute-force attack, sketch attack, and jigsaw puzzle solver attack, that can recover the encrypted information without using any keys. We will theoretically consider the resilience of our method against these COAs in this subsection and experimentally show its actual attack resilience and statistical analyses in Section IV-C.

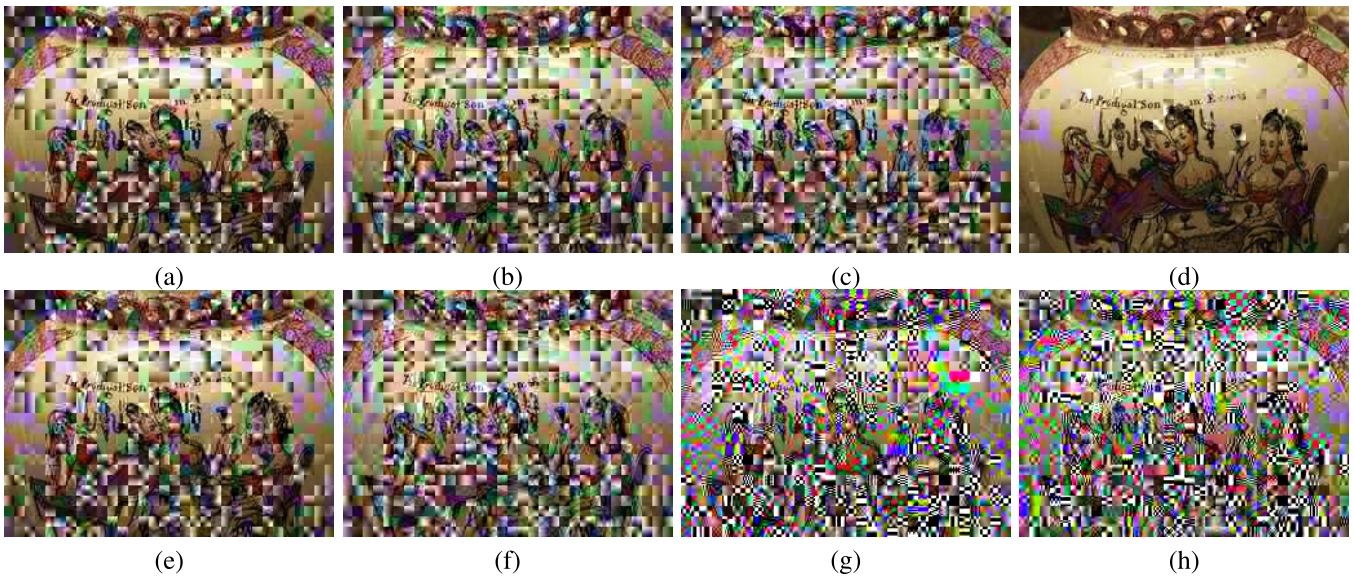

Fig. 6. Perceptual degradation induced by m-cubes of different sizes and their probabilities in BEESB (JPEG $Q = 70$): (a) $\ell_1^{[2]} = 50$, (b) $\ell_1^{[2]} = 70$, (c) $\ell_1^{[2]} = 90$, (d) $\ell_2^{[2]} = 50$, (e) $\ell_{1,2,3}^{[2]} = 50$, (f) $\ell_1^{[2]} = 70, \ell_2^{[2]} = 90, \ell_3^{[2]} = 100$, (g) $\ell_{1,2,3}^{[2]} = 50$ & $\ell_{1,2,3}^{[3]} = 50$ & $\ell_{1,2,3}^{[4]} = 50$, and (h) $\ell_{1,2,3}^{[2]} = 80$ & $\ell_{1,2,3}^{[3]} = 60$ & $\ell_{1,2,3}^{[4]} = 40$.

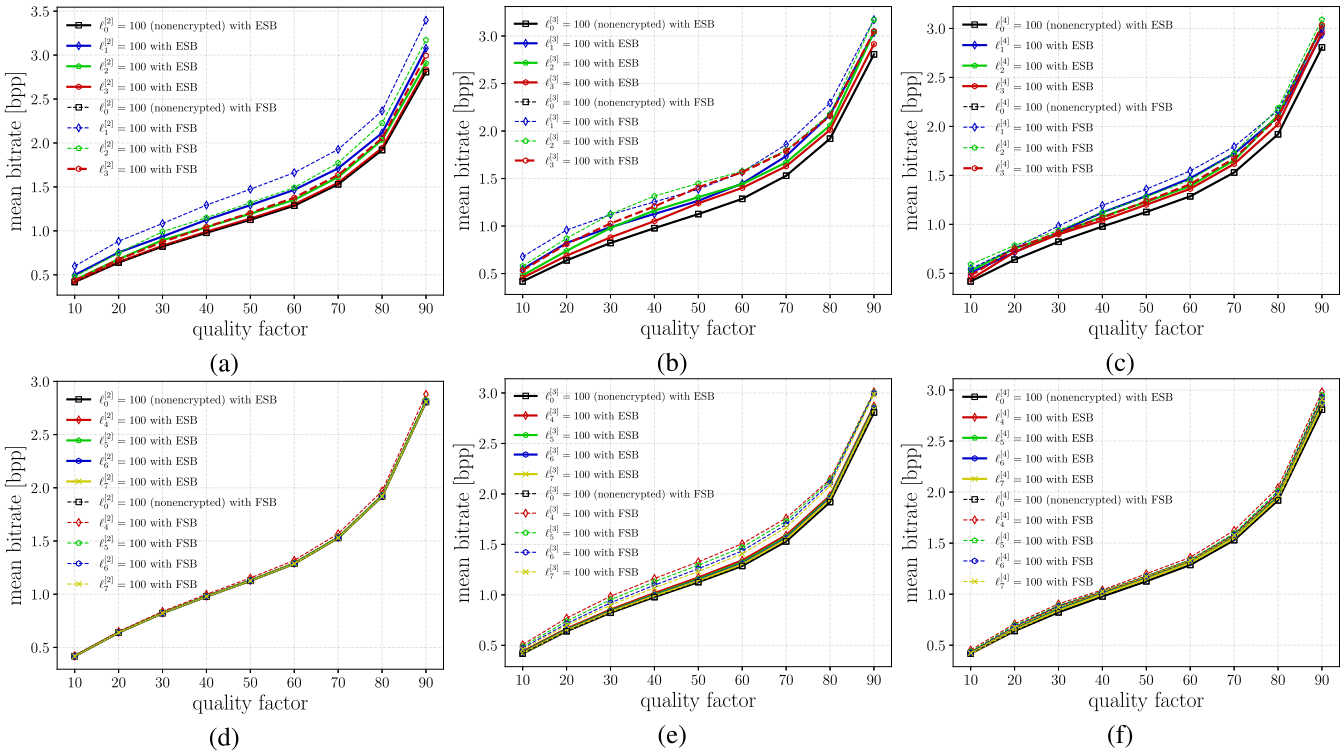

Fig. 7. Bitrate overhead analyses with the BEESB or BEFSB: (a) $\ell_0^{[2]}, \cdots$, or $\ell_3^{[2]} = 100$, (b) $\ell_0^{[3]}, \cdots$, or $\ell_3^{[3]} = 100$, (c) $\ell_0^{[4]}, \cdots$, or $\ell_3^{[4]} = 100$, (d) $\ell_4^{[2]}, \cdots$, or $\ell_7^{[2]} = 100$, (e) $\ell_4^{[3]}, \cdots$, or $\ell_7^{[3]} = 100$, and (f) $\ell_4^{[4]}, \cdots$, or $\ell_7^{[4]} = 100$.

First, let us guess its resilience against the replacement attack, which overwrites the encrypted portions with a constant value (e.g., 0) in order to disclose the other unencrypted portions at the decoder stage. BEESB encrypts the m-cubes containing many original image features and the replacement attack overwrites the encrypted m-cubes. As a result, since the overwritten ones are the characteristic ones, BEESB can protect the original image features from the replacement attack. At lower compression, it is clear that the BEESB more resiliently protects the original image features from the replacement attack than does the higher compression, thanks to the ability to encrypt more number of characteristic m-cubes.

Next, let us guess the resilience against the brute-force attack. BEESB has more resilience against this attack as the number of encrypted m-cubes increases. Since it takes 2^{3h} attacks to determine the correct internal bit arrangement of

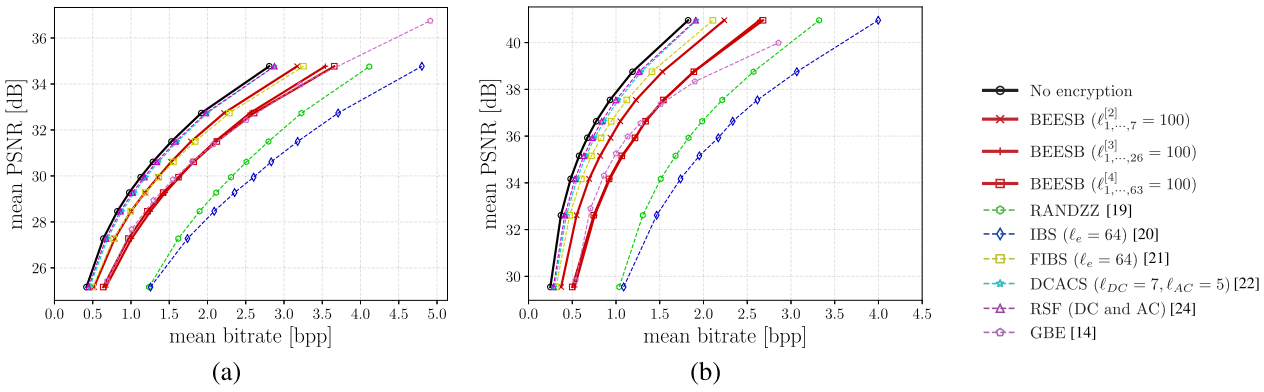

Fig. 8. R-D curves of JPEG compression with BEESB and the conventional methods: (a) UCID dataset [32] and (b) RAISE dataset [33].

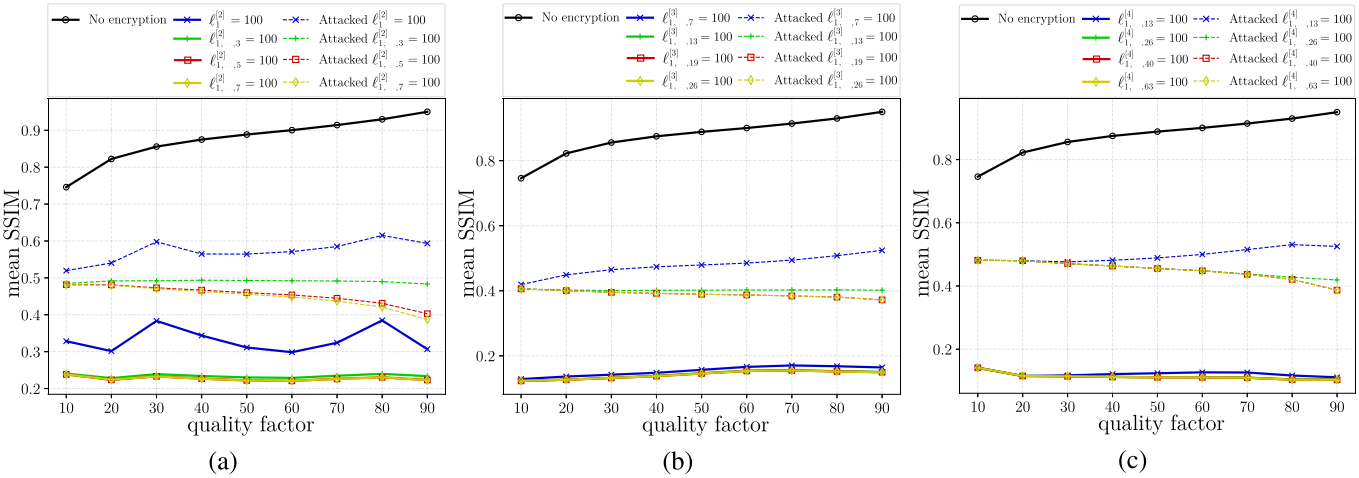

Fig. 9. Perceptual degradation and recovery for BEESBs with the various types and sizes of m-cubes: (a) 2^3 bitcubes, (b) 3^3 bitcubes, and (c) 4^3 bitcubes.

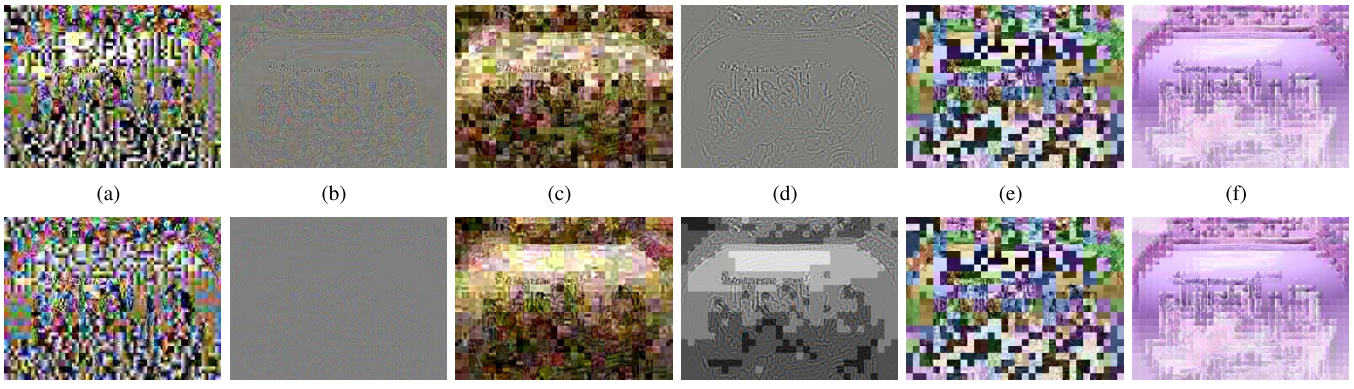

Fig. 10. Images encrypted with BEESB ($\ell^{[2]}_{1,\cdots,7} = 100$), DCACS ($\ell_{DC} = 7, \ell_{AC} = 5$), and RSF (DC and AC) and ones produced by the replacement attack: (a) BEESB ($Q = 50$), (b) attacked (a), (c) DCACS [22] ($Q = 50$), (d) attacked (c), (e) RSF [24] ($Q = 50$), (f) attacked (e), (g) BEESB ($Q = 90$), (h) attacked (g), (i) DCACS [22] ($Q = 90$), (j) attacked (i), (k) RSF [24] ($Q = 90$), and (l) attacked (k).

each bitcube packed with h^3 bits, it takes $(2^{3h})^N = 2^{3hN}$ attacks to do so on N encrypted m-cubes. Suppose that the SHA-256 hash digest [34] requiring 2^{256} attacks is the secure criterion; then, the resilience is ensured when $3hN \geq 256$. In addition, since the encrypted bit sequences are difficult to distinguish from the nonencrypted ones in the QDCT domain, the correct QDCT coefficients cannot be found from only the following hints:

- The differences between the DC coefficient and the AC coefficients in each block are explicitly large.
- The DC coefficient in each block usually cannot be negative without any offsetting.

The above information rather ensures that the original QDCT coefficients cannot be found. Consequently, it is very difficult for an attacker to find even a plausible QDCT coefficient set. Nonetheless, since the encrypted QDCT coefficients may

TABLE IV

BD METRICS OF BEESBS AND THE CONVENTIONAL METHODS IN THE CASE OF RAISE DATASET [33]: (A) BEESB ($\ell^{[2]}_{1,\cdots,7} = 100$), (B) BEESB ($\ell^{[3]}_{1,\cdots,26} = 100$), (C) BEESB ($\ell^{[4]}_{1,\cdots,63} = 100$), (D) RANDZZ [19], (E) IBS [20] ($\ell_e = 64$), (F) FIBS [21] ($\ell_e = 64$), (G) DCACS [22] ($\ell_{DC} = 7, \ell_{AC} = 5$), (H) RSF [24], AND (I) GBE (FBJ) [14]

Methods	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
mean BD-PSNR [dB]	−1.87	−3.43	−3.39	−6.44	−7.41	−1.18	−0.69	**−0.50**	−3.02
mean BD-rate [%]	39.04	81.78	79.98	183.90	226.62	23.07	13.03	**9.34**	79.91

TABLE V

MEANINGS OF THE TUNABILITY PARAMETERS USED IN CONVENTIONAL FWJS

tunability parameters	meanings
$\ell_e\ (\in \mathbb{Z}_{[1\ 64]})$	number of encrypted coefficients from the DC one in zig-zag order of each block
$\ell_{DC}\ (\in \mathbb{Z}_{[1\ 7]})$	number of encrypted DC bitplanes from the LSB one
$\ell_{AC}\ (\in \mathbb{Z}_{[1\ 5]})$	number of encrypted AC coefficients from the lowest one (except for the DC one) in zig-zag order of each block

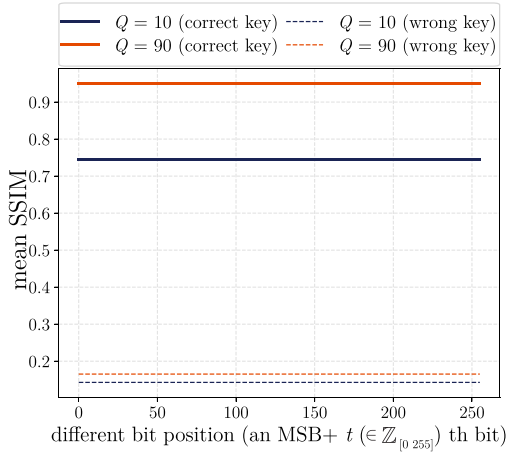

Fig. 11. Key sensitivity analysis when each bit of encryption key is different.

unfortunately be recovered with an incorrect decryption key, we should analyze the sensitivity when the wrong keys are exploited. We do so in Section IV-C.

Finally, let us guess the resilience against the sketch and jigsaw puzzle solver attacks. Note that the BEESB encrypts the blockwise detailed local textures. Although the sketch attack may reveal the outline formed by 8×8 encrypted blocks, it cannot recover each encrypted block clearly, i.e., it means that each detailed texture per block is still protected locally. Also, it is clear that the jigsaw puzzle solver attack cannot correctly connect the encrypted blocks. Therefore, this work will not mention the sketch and jigsaw puzzle solver attacks anymore.

On the other hand, since the histograms of the encrypted-decoded images may help that the attackers infer the original information, we should analyze the histograms through the experiments. We will show the analyses in Section IV-C.

TABLE VI

SSIM INDICATORS SUMMARIZED IN THE LITERATURE [35]

SSIM	Category
0.98+	indistinguishable from the original
0.95	barely watchable
0.9	really ugly
0.8	cannot still produce
0.7	fail to generate

IV. COMPARATIVE EXPERIMENTS

We show three experimental results to demonstrate the superiority of BEESB to the conventional methods. In Section IV-A, we analyze the fine tunability afforded by different types and sizes of m-cubes with various $\ell^{[h]}_m$s, which are arbitrary parameters with which to encrypt ℓ_m % of the m-cubes subdivided into h^3 bitcubes. In Section IV-B, we evaluated the actual compression efficiency of the bitstreams encrypted with BEESB in comparison with the conventional methods [14], [19]–[22]. In Section IV-C, we evaluated its resilience against the replacement attack and brute-force attack by overwriting the encrypted portions and exploiting incorrect keys and analyzed the histograms, which were encrypted with BEESBs. The experiments were based on the 100 full-color images from UnCompressed Image Dataset (UCID) [32], 20 full-color images from RAISE [33] dataset, and mid-standardized JPEG encoder/decoder software *libjpeg* [27]; hereafter, results of experiments were based on the UCID dataset unless otherwise noted. In the case of BEESBs and the conventional FWJs [19]–[22], the results were obtained by commonly iterating the following procedure:

1) JPEG-encode an input image while internally encrypting/nonencrypting the QDCT coefficients using the FWJs with the quality factor $Q = 10, 20, \cdots, 90$.
2) Measure the bitrates of the encrypted-encoded and normally encoded bitstreams obtained in 1).
3) JPEG-decode them while internally decrypting/nondecrypting the QDCT coefficients.
4) Measure the PSNRs and SSIMs [35] of the decrypted encrypted-decoded, nondecrypted encrypted-decoded, and normally decoded images obtained in 3).
5) Iterate from 1) to 4) for all input images to calculate the mean results.

Unlike the FWJs, the recent conventional FBJ [14] was applied to the pixel domain, i.e., outside the JPEG coder.

A. Fine Tunability

First, we confirmed the perceptual degradation induced by BEESB. In this experiment, we set an $8 \times 8 \times 8$ bitcuboid,

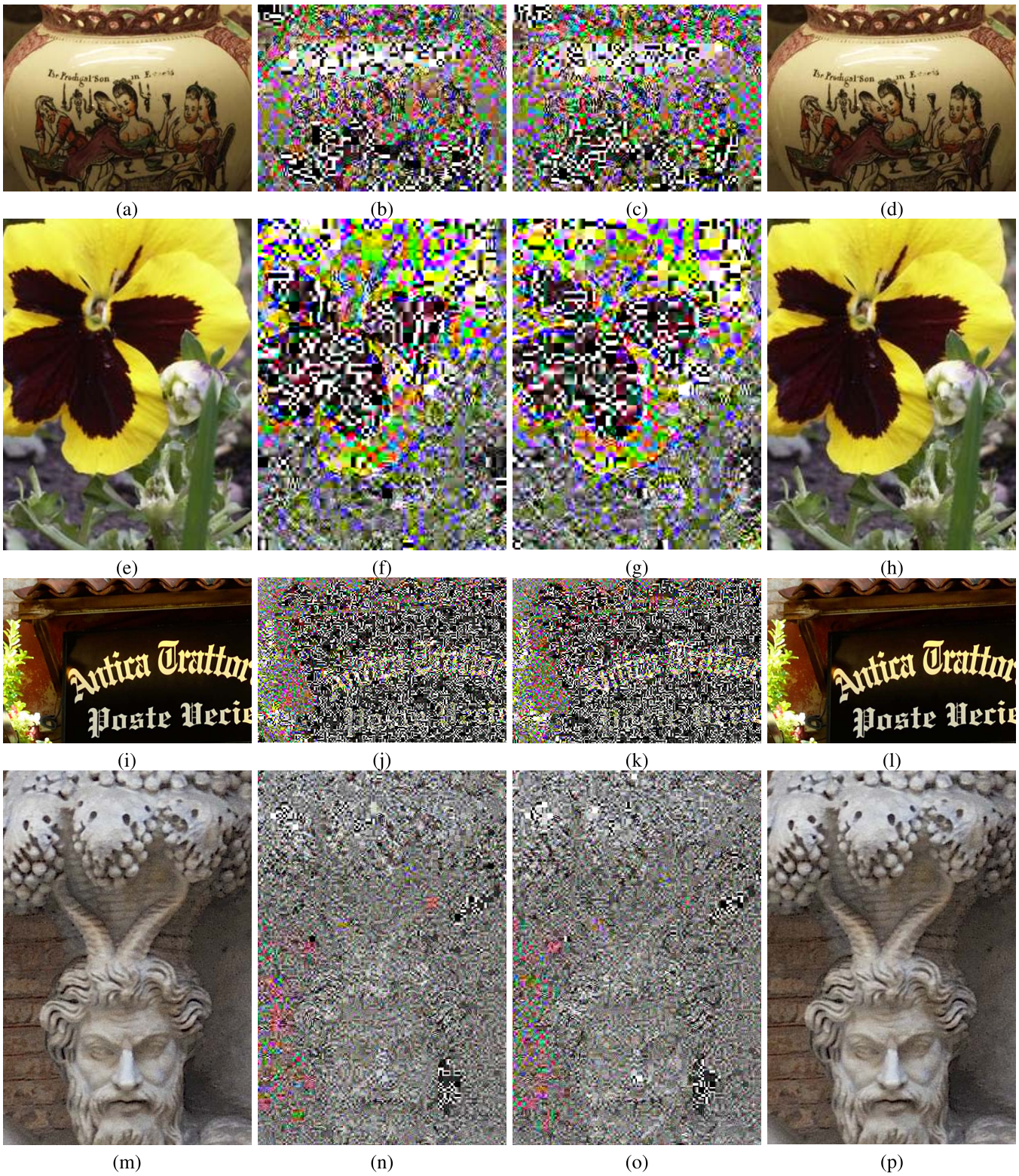

Fig. 12. Resilience against attacks with the wrong keys (JPEG $Q = 70$ and $\ell^{[3]}_{1,\cdots,26} = 100$): (a-d) particular areas of *ucid00059*, (e-h) particular areas of *ucid00069*, (i-l) particular areas of *r00444b95t*, (m-p) particular areas of *r00b8d4a2t*, (a,e,i,m) nonencrypted, (b,f,j,n) decrypted with the wrong key $\mathcal{F}_1$ completely different from the correct key $\mathcal{F}_0$, (c,g,k,o) decrypted with an wrong key $\mathcal{F}_3$ whose LSB is different from $\mathcal{F}_0$, and (d,h,l,p) decrypted with $\mathcal{F}_0$.

which was shaped from an 8×8 10-bit QDCT coefficient block without the most significant bit (MSB) (sign bit)- and LSB-planes and subdivided it into the nonoverlapping 64 2^3 bitcubes. This setting allowed for encryption of seven types of m-cubes ($m = 1, \cdots, 7$), where the internal bitwise relationships of the 0- and 8-cubes cannot be modified because they are just bitcubes filled with only 0s or only 1s. Different levels of perceptual degradation were provided to the encrypted-decoded image by changing the m-cubes to be encrypted as shown in Fig. 3. Since the 1-, 2-, and 3-cubes caused

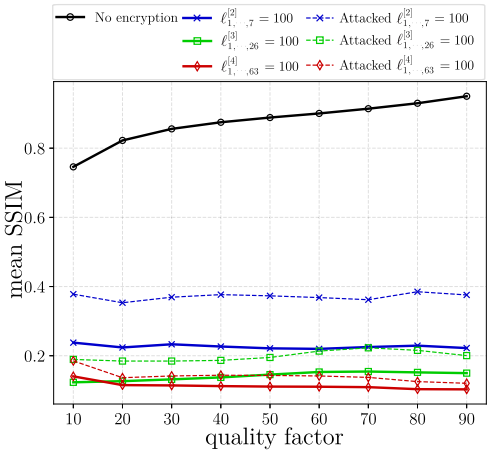

Fig. 13. Resilience against attacks with random numbers very close to the original ones used for the encryption.

more perceptual degradation while the other ones did so in negligible amounts, we omitted the case of 4- to 7-cubes here. As shown in Fig. 4, their combinations induced the largest perceptual degradation. BEFSB presented in [26] drastically changed both the colors and the textures due to the irregular values used by the exception-handling in FSB, whereas the BEESB did not do so as drastically because it does not use irregular values. The unpleasant degradation caused by BEFSB may risk attacks exploiting the explicit magnitude difference of the irregular value from the other QDCT coefficients, whereas the appropriate degradation by BEESB does not pose any risks.

Next, we evaluated the effect of different bitcube sizes. One can see that the BEESB reinforces the perceptual degradation level by varying the bitcube size, as shown in Fig. 5. The full encryption in the case of 2^3 bitcubes slightly preserved the outline of the image, because BEESB did not modify the bitwise relationships of the LSB plane. The full encryption in the case of 3^3 bitcubes encrypted the bits even in the LSB plane and provided relationally stronger visual confidentiality. The full encryption in the case of 4^3 bitcubes did not encrypt the LSB plane as in the case of 2^3 bitcubes, but provided a significantly strong perceptual degradation, because the BEESB varied the bitwise relationships in the larger constrained subspaces more than the one with 2^3 bitcubes.

Finally, we evaluated the effect of various settings in combination. By varying the sizes of the m-cubes, overlapping the bitcube subdivisions, and/or setting different encryption probabilities, BEESB provided excellently fine tunability, as shown in Fig. 6. In the case of 2^3 bitcubes, encryption with only the 1- or 2-cubes provided limited tunability (Fig. 6(a-d)), but encryption with the different probabilities of the 1-, 2-, and 3-cubes attained finer tunability (Fig. 6(e, f)). Furthermore, encrypting m-cubes of various bitcube sizes and overlapping them yielded very fine tunability (Fig. 6(g, h)).

B. Compression Efficiency

First, we compared the bitrate overheads of the encrypted-encoded bitstreams generated by using BEESB and BEFSB. Regardless of the bitcube sizes, the BEESB compressed the bitstreams more efficiently than the BEFSB did (Fig. 7). In the cases of 1-, 2-, and 3-cubes, although the BEFSB suppressed the bitrate overheads, BEESB did so even more, with bitrate overheads approximately half as large (Fig. 7(a-c)). In the cases of 4-, 5-, 6-, and 7-cubes, although BEFSB still affected the bitrate overheads to some extent, BEESB no longer did so, as shown in Fig. 7(d-f).

We also compared the bitrate overheads of BEESBs and the conventional FWJs by drawing their rate-distortion (R-D) curves and calculating their Bjøntegaard delta (BD) metrics [36]. We selected RANDZZ [19], IBS [20], FIBS [21], DCACS [22], and RSF [24] as conventional FWJs and grayscale block-based encryption (GBE) [14] as a recent conventional FBJ and used the proposed ESB for binarization in the conventional FWJs except for FIBS and RSF. One can see that BEESBs outperformed RANDZZ and IBS as shown in Fig. 8 and Tables III and IV; here, Table V shows the meanings of the tunability parameters ℓ_e, ℓ_{DC}, and ℓ_{AC}. In particular, the case of 2^3 bitcubes outperformed GBE in

TABLE VII
ENCRYPTION SPACES [BITS] OF BEESB
(EACH OF $\ell^{[2]}_{1,\cdots,7}$, $\ell^{[3]}_{1,\cdots,26}$, AND $\ell^{[4]}_{1,\cdots,63} = 100$)

Image	JPEG Q	BEESB		
		$\ell^{[2]}_{1,\cdots,7}$	$\ell^{[3]}_{1,\cdots,26}$	$\ell^{[4]}_{1,\cdots,63}$
ucid00001	50	96744	285120	427904
	70	125376	386208	520192
	90	246344	545670	702720
ucid00010	50	119744	340740	468032
	70	162520	436968	577216
	90	315600	624861	810368
ucid00020	50	125584	365283	440448
	70	175592	448335	591360
	90	337688	674271	868288
ucid00030	50	128632	339795	507328
	70	169880	439344	612416
	90	322304	620325	812544
ucid00040	50	77920	246348	356416
	70	101456	299538	453888
	90	167824	471852	601664
ucid00050	50	93752	305532	383808
	70	118512	360153	462272
	90	209056	527715	728000
ucid00060	50	82808	288225	370176
	70	109480	350730	599360
	90	200000	525906	702848
ucid00070	50	92848	302724	376320
	70	132840	374436	514624
	90	255200	559656	738176
ucid00080	50	134480	331236	474752
	70	175848	342171	574272
	90	342376	588708	797888
ucid00090	50	96872	293409	434176
	70	127112	392418	526400
	90	233920	559521	710144
Mean	50	105154	315836	407042
	70	143589	392176	522859
	90	271216	580330	758704

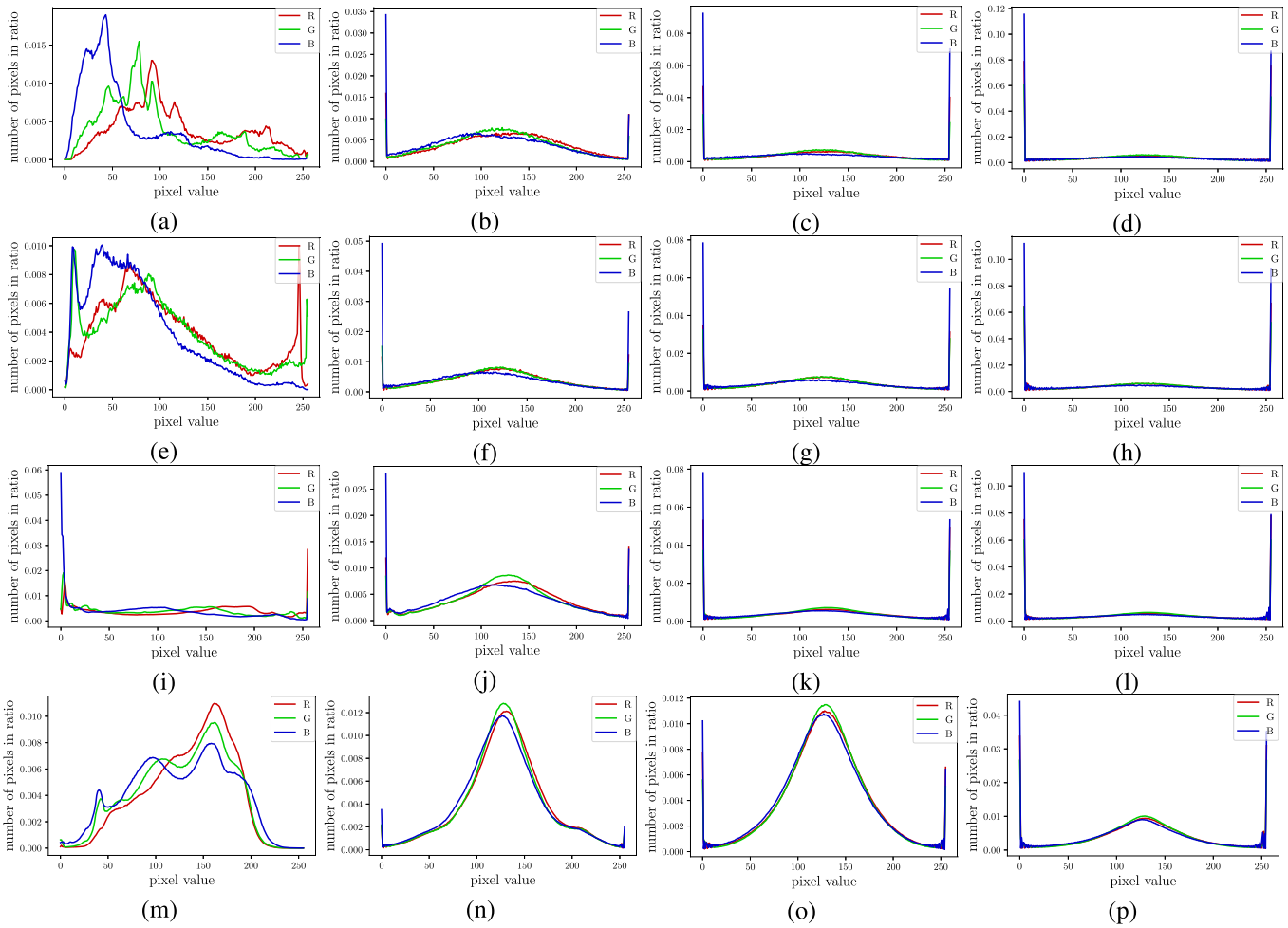

Fig. 14. Histogram analyses of BEESB-encrypted-decoded images (JPEG $Q = 70$): (a-d) *ucid00059*, (e-h) *ucid00069*, (i-l) *r00444b95t*, (m-p) *r00b8d4a2t*, (a,e,i,m) nonencrypted, (b,f,j,n) $\ell^{[2]}_{1,\cdots,7} = 100$, (c,g,k,o) $\ell^{[3]}_{1,\cdots,26} = 100$, and (d,h,l,p) $\ell^{[4]}_{1,\cdots,63} = 100$.

both image datasets and FIBS in UCID dataset. Note that DCACS and RSF, which outperformed BEESBs, are not subjectively resilient against the replacement attacks as shown in Section IV-C and their less tunability cannot meet the goal of this study: the fine tunability. Also, although FIBS tended to outperform BEESBs especially for high resolution images such as RAISE dataset, the FIBS cannot achieve the fine tunability like DCACS and RSF.

C. Resilience Against Attacks

First, we compared the attack resilience against the replacement attack in terms of the mean SSIMs [35] between the encrypted-decoded attacked/nonattacked images and the original ones. In the case of 2^3 bitcubes, the resilience of the BEESB with three types of m-cubes was preserved even in lower compression, whereas the resilience of the one with seven types of the m-cubes was further reinforced (Fig. 9(a)) because of the increased number of characteristic m-cubes. In the case of 3^3 bitcubes, the resilience of the BEESBs with up to seven types of m-cubes was low, but the resilience increased rapidly with 13 or more types of m-cubes (Fig. 9(b)). In the case of 4^3 bitcubes, the resilience of the BEESBs with more than 26 types of m-cubes were high enough (Fig. 9(c)). In addition, Fig. 10 shows images encrypted with BEESB ($\ell^{[2]}_{1,\cdots,7} = 100$), DCACS ($\ell_{DC} = 7, \ell_{AC} = 5$), and RSF as well as attacked ones in the case of $Q = 50$ and 90. It is clear that DCACS and RSF, which achieved the highest bitrate overhead suppression in Section IV-B, sacrificed the subjective resilience. If a reasonable level of security is to be expected, as in the SBSs, since the distorted content, whose SSIMs are smaller than 0.9, are subjectively regarded as 'really ugly' from Table VI, the distorted content, whose SSIMs are smaller than 0.5 as shown in Fig. 9, are ugly enough; i.e., not enough content can be recovered even if they are attacked. Furthermore, $\ell^{[2]}_{1,\cdots,7}$, $\ell^{[3]}_{1,\cdots,26}$, and $\ell^{[4]}_{1,\cdots,63}$, which are stronger at lower compression, are recommended for the higher security when a strict level of security is required, as in the SNSs.

We also analyzed the resilience against the brute-force attack. As described in Section III-F, this study assumed the SHA-256 hash digest [34] as a long enough encryption key. Since the encrypted content may unfortunately be decrypted into the plausible content with the wrong key, we analyzed the key sensitivity when each bit of the key is different from the correct one. From MSB to LSB, we confirmed that the wrong key cannot obtain the original image quality as shown in Fig. 11; even subjectively, several images decrypted with the wrong keys obviously do not approximate the correct decryption results as shown in Fig. 12. Moreover, we also

showed the resilience in an extremely-rare situation that an attacker generates and exploits random numbers very close to the original ones used for the encryption in Fig. 13. In this case, BEESBs achieved sufficient resilience even when only the LSBs of the random numbers were different from the original ones. As the proof of the resilience, we counted the bits in the encryption spaces of BEESBs. We confirmed that there were larger encryption space of the QDCT domain in each image when the quality factor became larger, as shown in Table VII. This is because the higher compression quality preserves much more nonzero bits that can be encrypted with BEESB. $\ell^{[4]}_{1,\ldots,63}$ had the most amount of encryptable bits, because the 4^3 bitcube contains more amount of zero/nonzero bits in itself than any of 2^3 and 3^3 one. Since the above encryption spaces were obviously larger than the 256-bit encryption key and the 19936-bit MT state vector, the key space related to the encryption space was not smaller than the original key space 2^{256} and the vector space 2^{19936}. Therefore, one can see that the BEESB using larger bitcube size and multiple blocks definitely guarantees the attack resilience.

Otherwise, we analyzed the histograms of images encrypted by BEESB (Fig. 14). Even when we chose any bitcube sides from $h = 2, 3, 4$, the pixel values encrypted by BEESB tended to come closer to min/max pixel values. Moreover, larger bitcube sizes suppressed the center upheaval. Therefore, we can see that the BEESBs with larger bitcube sizes conceal the larger amount of original color information, while preserving the reasonable structure information.

V. Conclusion

We proposed BEESB, a JPEG format-compliant perceptual encryption with bitcuboid-based encryption (BE) and exception-free signed binarization (ESB). BE is an encryption technique that provides fine tunablility by encrypting bitcubes obtained by constraining the bitcuboid. ESB is a binarization technique that redecimalizes encrypted binary sequences into signed decimal coefficients without any exception-handling by shifting the negative binary sequences one-by-one. We applied BEESB to the QDCT domain in JPEG compression. Experiments with encryption in JPEG showed that BEESB achieved finer tunability, more efficient bitrate overhead suppression of the encrypted-encoded content compared with the conventional methods, and high resilience against attacks.

Acknowledgment

The authors would like to thank Armstrong (from Human Global Communications Company, Ltd.) for many time checks, advises, and rewrites of this paper, and also would like to thank peer-reviewers for many recommendations and thoughtful comments regarding to this paper.

References

[1] *Information Technology—Digital Compression and Coding of Continuous-Tone Still Images—Requirements and Guidelines*, document ITU-T Recommendation T.81, 1993.

[2] *Series H: Audiovisual and Multimedia Systems Infrastructure of Audiovisual Services—Coding of Moving Video*, document ITU-T Recommendation H.265, 2013.

[3] *FIPS PUB 197: Advanced Encryption Standard (AES)*, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Nov. 2001.

[4] W. Puech and J. M. Rodrigues, "Crypto-compression of medical images by selective encryption of DCT," in *Proc. EUSIPCO*, Antalya, TR, USA, Sep. 2005, pp. 1–4.

[5] G. Ye, C. Pan, Y. Dong, K. Jiao, and X. Huang, "A novel multi-image visually meaningful encryption algorithm based on compressive sensing and Schur decomposition," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 2, pp. 1–14, Feb. 2021.

[6] H. Hofbauer, A. Uhl, and A. Unterweger, "Transparent encryption for HEVC using bit-stream-based selective coefficient sign encryption," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 1986–1990.

[7] D. Engel, T. Stütz, and A. Uhl, "A survey on JPEG2000 encryption," *Multimedia Syst.*, vol. 15, no. 4, pp. 243–270, Jan. 2009.

[8] G. Hong, C. Yuan, Y. Wang, and Y. Zhong, "A quality-controllable encryption for H.264/AVC video coding," in *Proc. Pacific-Rim Conf. Multimedia*, Hangzhou, China, Nov. 2006, pp. 510–517.

[9] Y. Wang, M. O'Neill, and F. Kurugollu, "A tunable encryption scheme and analysis of fast selective encryption for CAVLC and CABAC in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 9, pp. 1476–1479, Sep. 2013.

[10] F. Peng, X. Zhang, Z.-X. Lin, and M. Long, "A tunable selective encryption scheme for H.265/HEVC based on chroma IPM and coefficient scrambling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2765–2780, Aug. 2020.

[11] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for JPEG/motion JPEG standard," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. 98, no. 11, pp. 2238–2245, 2015.

[12] K. Shimizu, T. Suzuki, and K. Kameyama, "Cube-based encryption-then-compression system for video sequences," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. 101, no. 11, pp. 1815–1822, Nov. 2018.

[13] K. Shimizu, T. Suzuki, and K. Kameyama, "Lapped cuboid-based perceptual encryption for motion JPEG standard," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Honolulu, HI, USA, Nov. 2018, pp. 2022–2026.

[14] 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.

[15] A. Unterweger, K. V. Ryckegem, D. Engel, and A. Uhl, "Building a post-compression region-of-interest encryption framework for existing video surveillance systems," *Multimedia Syst.*, vol. 22, no. 5, pp. 617–639, Oct. 2016.

[16] K. Minemura, K. Wong, X. Qi, and K. Tanaka, "A scrambling framework for block transform compressed image," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 6709–6729, Mar. 2017.

[17] V. Itier, P. Puteaux, and W. Puech, "Recompression of JPEG crypto-compressed images without a key," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 646–660, Mar. 2020.

[18] J. Ting, K. Wong, and S. Ong, "Format-compliant perceptual encryption method for JPEG XT," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 4559–4563.

[19] L. Tang, "Methods for encrypting and decrypting MPEG video data efficiently," in *Proc. 4th ACM Int. Conf. Multimedia (MULTIMEDIA)*, Boston, MA, USA, Nov. 1996, pp. 219–229.

[20] 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, Jul. 2006.

[21] 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, Sep. 2007.

[22] 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. Int. Conf. Intell. Adv. Syst.*, Kuala Lumpur, Malaysia, Jun. 2010, pp. 1–6.

[23] B. Zeng, S.-K.-A. Yeung, S. Zhu, and M. Gabbouj, "Perceptual encryption of H.264 videos: Embedding sign-flips into the integer-based transforms," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 2, pp. 309–320, Feb. 2014.

[24] P. Li and K.-T. Lo, "Joint image compression and encryption based on order-8 alternating transforms," *J. Vis. Commun. Image Represent.*, vol. 44, pp. 61–71, Apr. 2017.

[25] D. Bridger, D. Danon, and A. Tal, "Solving jigsaw puzzles with eroded boundaries," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 3526–3535.

[26] K. Shimizu and T. Suzuki, "Flexibly-tunable bitcube-based perceptual encryption within JPEG compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 2702–2706.
[27] T. Richter. (2019). *ThorFDBG/libJPEG: A Complete Implementation of 10918-1 (JPEG) Comming From JPEG.org (the ISO Group) With Extensions for HDR Standardized as 18477 (JPEG XT)*. Accessed: Mar. 25, 2020. [Online]. Available: https://github.com/thorfdbg/libjpeg
[28] C.-J. Lian, K.-F. Chen, H.-H. Chen, and L.-G. Chen, "Analysis and architecture design of block-coding engine for EBCOT in JPEG 2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 3, pp. 219–230, Dec. 2003.
[29] V. Sze and M. Budagavi, "High throughput CABAC entropy coding in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1778–1791, Dec. 2012.
[30] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, pp. 3–30, Jan. 1998.
[31] H. T. Sencar and N. D. Memon, "Identification and recovery of JPEG files with missing fragments," in *Proc. 9th Annu. DFRWS Conf.*, Sep. 2009, pp. 88–98.
[32] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," *Proc. SPIE*, vol. 5307, pp. 472–480, Dec. 2004.
[33] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *Proc. 6th ACM Multimedia Syst. Conf.*, Mar. 2015, pp. 219–224.
[34] *FIPS PUB 180-4: Secure Hash Standard (SHS)*, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Aug. 2015.
[35] 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.
[36] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RDcurves*, document VCEG-M33, 2001.

Kosuke Shimizu received the B.E. degree in Computer Science Program of the Advanced Course from Tokyo Metropolitan College of Industrial Technology (TMCIT), Japan, in 2017, and the M.E. degree from the Department of Computer Science, University of Tsukuba, Japan, in 2019, where he is currently pursuing the Ph.D. degree. His current research interest is image and video processing.

Taizo Suzuki (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Japan, in 2004, 2006, and 2010, respectively. From 2006 to 2008, he was with Toppan Printing Company, Ltd., Japan. From 2008 to 2011, he was a Research Associate with the Global Center of Excellence (G-COE), Keio University. From 2010 to 2011, he was a Research Fellow with Japan Society for the Promotion of Science (JSPS) and a Visiting Scholar with the Video Processing Group, University of California at San Diego, La Jolla, CA, USA. From 2011 to 2012, he was an Assistant Professor with Nihon University, Japan. In 2012, he joined the University of Tsukuba, Japan, as an Assistant Professor, where he has been an Associate Professor since 2019. His current research interests include signal processing and filter banks/wavelets for image and video. From 2017 to 2021, he was an Associate Editor of the *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*.