Lapped Cuboid-based Perceptual Encryption for Motion JPEG Standard

Kosuke Shimizu*, Taizo Suzuki†, and Keisuke Kameyama†
* Department of Computer Science, University of Tsukuba, Japan
E-mail: shimizu@adapt.cs.tsukuba.ac.jp
† Faculty of Engineering, Information and Systems, University of Tsukuba, Japan
E-mail: {taizo, Keisuke.Kameyama}@cs.tsukuba.ac.jp

***Abstract*—This paper proposes cuboid-based perceptual encryption (Cd-PE) and a version of cube-based perceptual encryption (C-PE), named lapped cuboid-based perceptual encryption (LCd-PE), to enhance the security for Motion JPEG (MJPEG). Although C-PE provides a high level of security by dealing with several frames of the input video sequence simultaneously, keyless attackers may illegally decrypt the encrypted video sequence with conceivable attacks such as a cube-based jigsaw puzzle solver (CJPS) attack. LCd-PE subdivides the video sequence pre-encrypted with C-PE into small cuboids and further encrypts it so that attackers cannot conduct attacks such as CJPS. The experiments show that the compression performance of an encryption-then-compression (ETC) system with LCd-PE and MJPEG is almost equivalent to that of one using C-PE and yet achieves a higher level of security.**

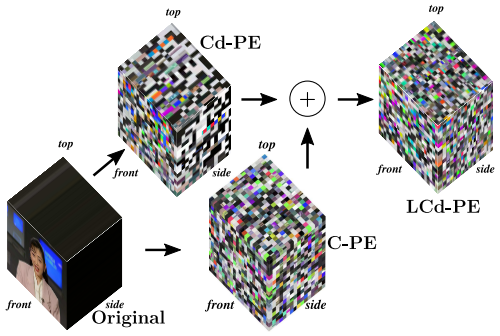

Fig. 1. LCd-PE consists of C-PE and Cd-PE.

I. Introduction

Multimedia communications on unsecured channels have been regarded as dangerous. Such communications include open systems such as social networking services (SNSs) allowing the anyone to see the uploaded contents except for login/logout and video on demand (VoD) and Pay TV controlling the quality of sent content. On the other hand, they do not include login/logout video on demand (VoD) and Pay TV systems controlling the quality of the sent contents. Open systems have different requirements for achieving both compression efficiency and real-time processing for applications using the saved content. For SNSs, a suitable security level is determined according to a trade-off with the compression ratio of the encrypted content, because communications are band-limited. For VoD and Pay TV, the suitable security level is determined according to a trade-off with real-time processing requirement in decoding, because their content should be sent without delay.

The conventional encryption schemes [1]–[3] have to be compatible with existing compression frameworks and able to decrypt video sequences in real-time. Most of them encrypt content on a per frame basis while hardly modifying the coefficients of the content being compressed. Because of that, keyless attackers may try to forcibly decrypt the encrypted coefficients with only one frame. In particular, [1] focused on the encryption-then-compression (ETC) system for the Motion JPEG (MJPEG) standard, which encrypts the input content before the international compression standard and can decrypt the content even if it is recompressed later. On the other hand, [2], [3] focused on encryption during compression for Advanced Video Coding (AVC) and High Efficiency Video Coding (HEVC), which are popular international compression standards. We previously presented an ETC system with a cube-based perceptual encryption (C-PE) for MJPEG in [4], because it is quite difficult to encrypt the input content before AVC or HEVC due to intra and inter predictions. Our system achieved a high level of security by processing the frames of the input video sequence simultaneously, but at the expense of a though real-time decryption ability. To be more specific, it regards the whole video sequence as a large cuboid, divides it into small cubes, and applies C-PE consisting of cube rotation, cube scrambling, cube negative-positive (nega-posi) reversal, and cube color component shuffling to the small cubes. C-PE provides a high level of security by dealing with several frames of the input video sequence simultaneously. However, keyless attackers may attempt to decrypt the encrypted video sequence illegally with conceivable attacks such as a cube-based jigsaw puzzle solver (CJPS) attack.

This paper proposes cuboid-based perceptual encryption (Cd-PE) and a version of conventional C-PE, named lapped cuboid-based perceptual encryption (LCd-PE), to enhance the security for MJPEG (Fig. 1). Fig. 1 shows that LCd-PE works by combining complete encryption with C-PE and partial encryption with Cd-PE. When LCd-PE is applied, the cubes of the same size created by C-PE are subdivided into cuboids of various sizes. The LCd-PE subdivides the video sequence pre-encrypted with C-PE into small cuboids and further encrypts

it so that attackers cannot conduct attacks such as CJPS. Our experiments show that the compression performance of an ETC system with LCd-PE and MJPEG is almost equivalent to that of the C-PE case and yet achieves a higher level of security.

II. Cube-based Perceptual Encryption

A. Method Details

C-PE consists of cube rotation, cube scrambling, cube nega-posi reversal, and cube color component shuffling (Fig. 2 (a)) [4]. It is conducted after dividing the whole input video sequence, regarded as a large cuboid, into small "cubes".

Cube rotation rotates smaller cube through four random angles: $0^\circ, 90^\circ, 180^\circ$, and 270° in three directions: depthwise, vertically, and horizontally (Fig. 3 (a)). When the cube rotation is conducted depthwise and vertically, a block in the observed frame is exchanged with a block in another frame. In particular, when the depthwise and vertical rotation angles are 90° or 270°, the spatiotemporal sides of the frames appear in the observed frame. Therefore, C-PE precludes keyless decryption of only a single frame. However, if the cube rotation is applied to MJPEG, the spatiotemporal sides that are usually not processed with MJPEG appear in the encrypted frames so that the compression efficiency is affected depending on the input video sequence. To decrypt the cube-rotated video sequence without any decryption key, the attacker must pick the encrypted blocks from other frames or spatiotemporal sides.

Cube scrambling permutes a pair of two randomly chosen cubes (Fig. 4 (a)). Exchanging cubes moves the blocks in the original frames to other frames. To decrypt the cube-scrambled video sequence without any decryption key, the attacker must pick the encrypted blocks from the other frames, like in the cube rotation case.

Cube nega-posi reversal randomly inverts the colors in the small cubes. Let ${}^\forall c \in \mathcal{C}_i = \{(\mathcal{R}_i, \mathcal{G}_i, \mathcal{B}_i)\}$ and $\mathcal{C}'_i$ be the pixel of the ith cube consisting of RGB components $\mathcal{R}_i, \mathcal{G}_i$, and $\mathcal{B}_i$ and the ith encrypted cube, respectively. The inverted colors in the ith cube are calculated as

$$\mathcal{C}'_i = \begin{cases} \bigcup_{\mathcal{C}_i} 255 - c & (\varepsilon_2(i) = 0) \\ \bigcup_{\mathcal{C}_i} c & (\varepsilon_2(i) = 1) \end{cases}, \tag{1}$$

where $\varepsilon_m(n)$ means an m-ary random number of the nth cuboid.

Cube color component shuffling permutes the order of the color components in each cube randomly. The shuffled order of the color components in the ith cube is calculated as

$$\mathcal{C}'_i = \begin{cases} \{(\mathcal{R}_i, \mathcal{G}_i, \mathcal{B}_i)\} & (\varepsilon_6(i) = 0) \\ \{(\mathcal{R}_i, \mathcal{B}_i, \mathcal{G}_i)\} & (\varepsilon_6(i) = 1) \\ \{(\mathcal{G}_i, \mathcal{R}_i, \mathcal{B}_i)\} & (\varepsilon_6(i) = 2) \\ \{(\mathcal{G}_i, \mathcal{B}_i, \mathcal{R}_i)\} & (\varepsilon_6(i) = 3) \\ \{(\mathcal{B}_i, \mathcal{R}_i, \mathcal{G}_i)\} & (\varepsilon_6(i) = 4) \\ \{(\mathcal{B}_i, \mathcal{G}_i, \mathcal{R}_i)\} & (\varepsilon_6(i) = 5) \end{cases}. \tag{2}$$

B. Advantage and Disadvantage

Unlike other frame-based encryptions [1]–[3], C-PE encrypts several frames simultaneously. In addition, the cube rotation produces spatiotemporal sides, which affect the compression efficiency depending on the input video sequence, in the observed frame. Thus, C-PE indeed precludes keyless decryption of only a single frame. However, the encrypted video sequence may be illegally decrypted by concatenating cubes of uniform sizes, i.e., by conducting the CJPS attack. In this paper, we aim to preclude the CJPS attack.

III. Lapped Cuboid-based Perceptual Encryption

A. Method Details

The Cd-PE consists of cuboid rotation, cuboid scrambling, cuboid nega-posi reversal, and cuboid color component shuffling (Fig. 2 (b)). It is conducted after dividing the whole input video sequence, regarded as a large cuboid, into small "cuboids" whose sizes are $\ell_V \times \ell_H \times \ell_D$ $({}^\forall \ell_V, \ell_H, \ell_D \in \mathbb{Z}_{>0})$, where ℓ_V, ℓ_H, and ℓ_D are vertical, horizontal, and depthwise lengths, respectively.

Cuboid rotation rotates a smaller cuboid through random angles in the horizontal, vertical, and depthwise directions (Fig. 3 (b)). Each angle through which to rotate a chosen cuboid in each direction is determined with random numbers generated from a pseudo random number generator (PRNG). The cuboid is rotated through ${}^\forall \theta_1, \theta_2 \in \{0, 180\}^\circ$ vertically and depthwise and through ${}^\forall \theta_3 \in \{90, 180, 270\}^\circ$ horizontally, in accordance with the random numbers. Unlike in the cube rotation, application of cuboid rotation to MJPEG hardly affects compression efficiency because the spatiotemporal sides of the frames do not appear in the observed frame due to angle constraints.

Cuboid scrambling permutes a pair of two randomly chosen cuboids (Fig. 4 (b)). The two cuboids chosen to be scrambled are selected with random numbers generated from the PRNG.

Cuboid nega-posi reversal randomly inverts the colors in each smaller cuboid. It is conducted with a binary random number generated from the PRNG. In accordance with this number, the inverted colors of the ith cuboid ${}^\forall c \in \mathfrak{C}_i = \{(\mathfrak{R}_i, \mathfrak{G}_i, \mathfrak{B}_i)\}$ are calculated as

$$\mathfrak{C}'_i = \begin{cases} \bigcup_{\mathfrak{C}_i} 255 - c & (\varepsilon_2(i) = 0) \\ \bigcup_{\mathfrak{C}_i} c & (\varepsilon_2(i) = 1) \end{cases}. \tag{3}$$

Cuboid color component shuffling permutes the order of color components in each cuboid randomly. The cuboid color component shuffling is conducted with a 6-ary random number generated from the PRNG. In accordance with this random number, each shuffled color components of the ith cuboid is calculated as

$$\mathfrak{C}'_i = \begin{cases} \{(\mathfrak{R}_i, \mathfrak{G}_i, \mathfrak{B}_i)\} & (\varepsilon_6(i) = 0) \\ \{(\mathfrak{R}_i, \mathfrak{B}_i, \mathfrak{G}_i)\} & (\varepsilon_6(i) = 1) \\ \{(\mathfrak{G}_i, \mathfrak{R}_i, \mathfrak{B}_i)\} & (\varepsilon_6(i) = 2) \\ \{(\mathfrak{G}_i, \mathfrak{B}_i, \mathfrak{R}_i)\} & (\varepsilon_6(i) = 3) \\ \{(\mathfrak{B}_i, \mathfrak{R}_i, \mathfrak{G}_i)\} & (\varepsilon_6(i) = 4) \\ \{(\mathfrak{B}_i, \mathfrak{G}_i, \mathfrak{R}_i)\} & (\varepsilon_6(i) = 5) \end{cases}. \tag{4}$$

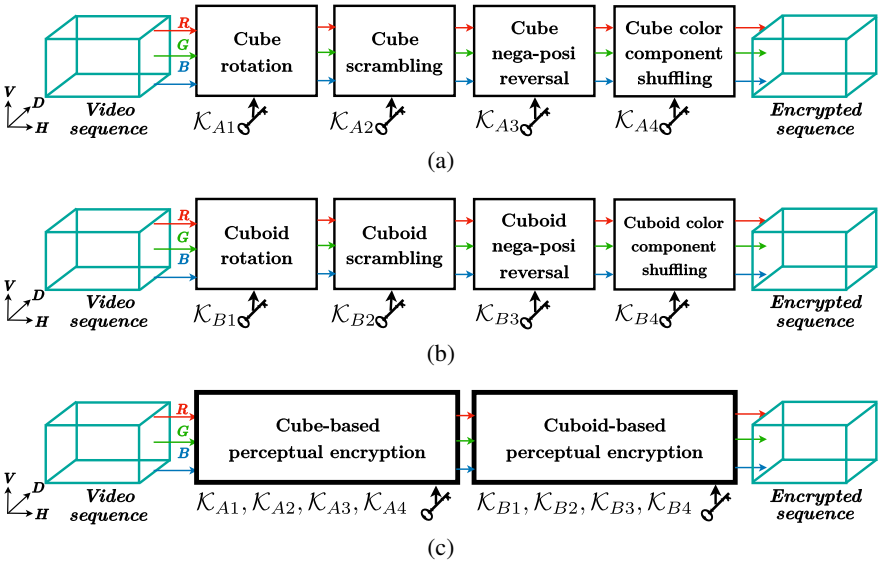

Fig. 2. Procedures of three perceptual encryptions: (top-to-bottom) conventional C-PE, Cd-PE, and LCd-PE.

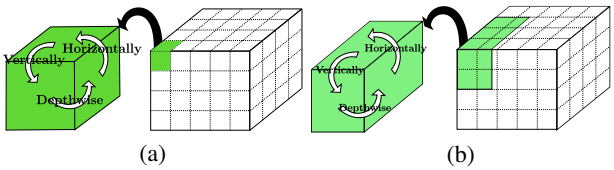

Fig. 3. Rotation methods: (a) cube rotation and (b) cuboid rotation.

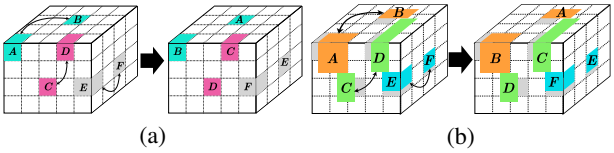

Fig. 4. Scrambling methods: (a) cube scrambling and (b) cuboid scrambling.

Here, we should note that it is desirable that there are as many sizes of cuboid as possible in the encrypted video sequence. Although one way to achieve this with only Cd-PE is to determine the sizes and coordinates of the cuboids for each Cd-PE operation, it is inefficient because the encryptor does not know the appropriate sizes or coordinates. Therefore, we decided to create a simpler encryption applying "partial" Cd-PE–the overall method is called LCd-PE–to a video sequence pre-encrypted with C-PE (Fig. 2 (c)). To achieve partial Cd-PE, we generate a binary random number from the PRNG, and when the number is 1, we apply each corresponding Cd-PE operation to a video sequence already pre-encrypted with C-PE. By iterating these operations, the video sequence is further divided and partially encrypted partially. Consequently, when LCd-PE is applied, the same sized cubes created by C-PE are subdivided into cuboids of various sizes.

B. Security

The conceivable attacks include the CJPS attack, which attempts to match cubes, and the algorithmic brute force (ABF) attack that tries all algorithmic candidates. [5] has analyzed the block-based jigsaw puzzle solver (BJPS) attack in the block-based perceptual encryption (B-PE) and proved that choosing an appropriate block size and B-PE method complicates the task presented to BJPS. Generally, when attackers match square blocks, they select the pair of sides that have minimal differences. However, since matching sides does not also match the intra-block textures when the block boundaries have distortions, BJPS must also take the intra-block variance into account. Whereas a CJPS attack against C-PE has not been realized yet, a suitable C-PE with an appropriate choice of cube size can sufficiently complicate the task of BJPS, since it involves matching six sides (of a cube surface) rather than just four (of a block side).

For LCd-PE, cube-rotated blocks exposing the spatiotemporal direction are more subdivided by Cd-PE. The subdivided widths are specified with the cuboid sizes, and they are various, as aforementioned. An attacker supposes the size of cubes and the number of bundled cuboids and then conducts the CJPS attack on the bundled cuboids. Since such an attack cannot be assured to finish, if the fully attacked video sequence is not correctly recovered, the CJPS attack is iterated again. Therefore, LCd-PE completely precludes the CJPS attack.

In addition, the encryptor of C-PE [4] precludes the ABF attack in real-time. The previous study states that the ABF attack against a cube-rotated and cube-scrambled sequence cannot be concluded in real-time. Since this is an insight into the case of C-PE with a constant cube size, it means the difficulty faced by ABF is further increased that when LCd-PE is used. For example, the cube rotation now has to be

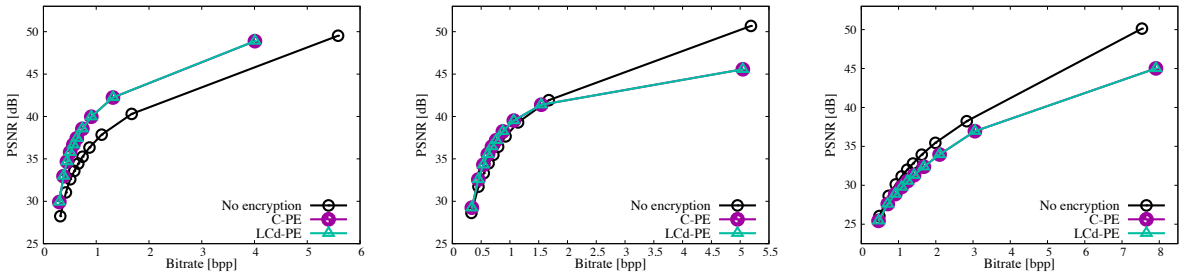

Fig. 5. R-D curve (average of 256 frames): (a) *Akiyo*, (b) *Bowing*, and (c) *Coastguard*.

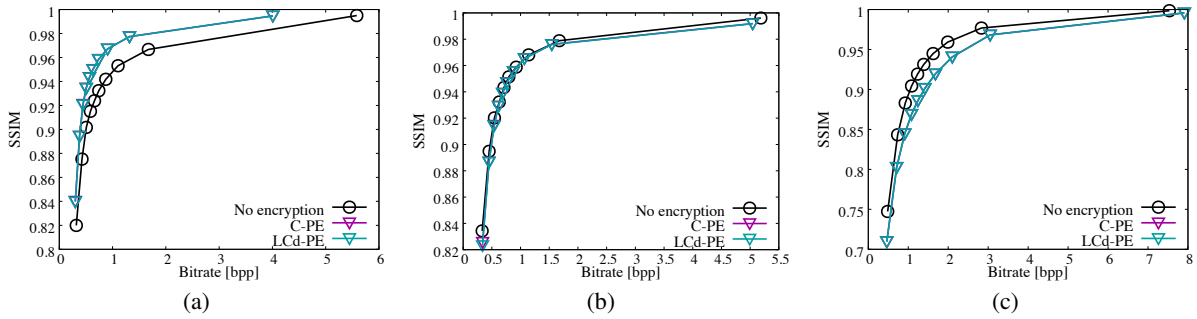

Fig. 6. SSIM performance (average of 256 frames): (a) *Akiyo*, (b) *Bowing*, and (c) *Coastguard*.

conducted on a cube constructed from randomly sized cuboids. The cube cannot have the correct thickness and cannot be used to reconstruct the original correct cube. Even if the incorrectly constructed cubes are inversely rotated, they cannot match any of the other ones. Therefore, we can see that LCd-PE is robust against both the CJPS attack and ABF attack.

IV. Experiments

A. Experimental Conditions and Procedure

We used three test video sequences [6] with different moving/stopping area sizes, as shown in Table I and Fig. 7 (a, d, g). The cube size used in C-PE was set as $16 \times 16 \times 16$ in accordance with [4], and the cuboid sizes used in LCd-PE were set as shown in Table II. Since the $16k \times 16l$ ($^\forall k, l \in \mathbb{Z}_{>0}$) blocks in the observed frames do not affect the 8×8 processing blocks in MJPEG and MJPEG does not conduct inter-frame prediction, unlike MPEG2, H.264/AVC, and H.265/HEVC, we can consider that the encryption with $16k \times 16l \times m$ ($^\forall k, l, m \in \mathbb{Z}_{>0}$) cuboids hardly affects the MJPEG compression performance (in accordance with [1]). However, the spatiotemporal sides shown by cube rotation affect the compression performance, as described in [4]. We thus evaluated the MJPEG compression performance of video sequences encrypted with LCd-PE.

The common procedure was as follows.

1) Apply LCd-PE to the frames.
2) Compress the encrypted frames with *libjpeg-turbo* [7], whose compression qualities are $\mathcal{Q} := \{10, 20, \cdots, 100\}$.
3) Calculate the mean bitrates of the compressed frames.
4) Decompress the compressed frames.
5) Decrypt the decompressed frames.
6) Calculate the mean PSNRs and mean SSIMs between the input frames and the decrypted frames.

B. Experimental Results

The rate-distortion (R-D) curves and the SSIM performances are shown in Fig. 5 and Fig. 6. The compression efficiencies of LCd-PE (blue lines) were equivalent to those of C-PE only (purple lines) [4]. This is because the spatiotemporal sides were shown by cube rotation for achieving high security. LCd-PE had almost the same compression performance as C-PE despite it having a higher level of security.

The results of the whole encryption are shown in Fig. 7. The video sequence encrypted only with C-PE (Fig. 7(b, e, h)) is still susceptible to a CJPS attack because the cubes are observed on the spatiotemporal sides. On the other hand, the video sequence encrypted with LCd-PE (Fig. 7(c, f, i)) has cuboids of variable depth. Keyless attackers must concatenate the cuboids randomly and match them with each other during rotation. Therefore, we can see that LCd-PE indeed precludes the CJPS attack.

Fig. 8 compares the spatiotemporal (top) images. When only C-PE is applied, the cubes are of the same size (Fig. 8 (a)). When the cuboid rotation and cuboid scrambling are added, finer divisions are generated by randomly specifying the cuboid widths (Fig. 8 (b)). Moreover, when cuboid nega-posi reversal and cuboid color component shuffling are added, more

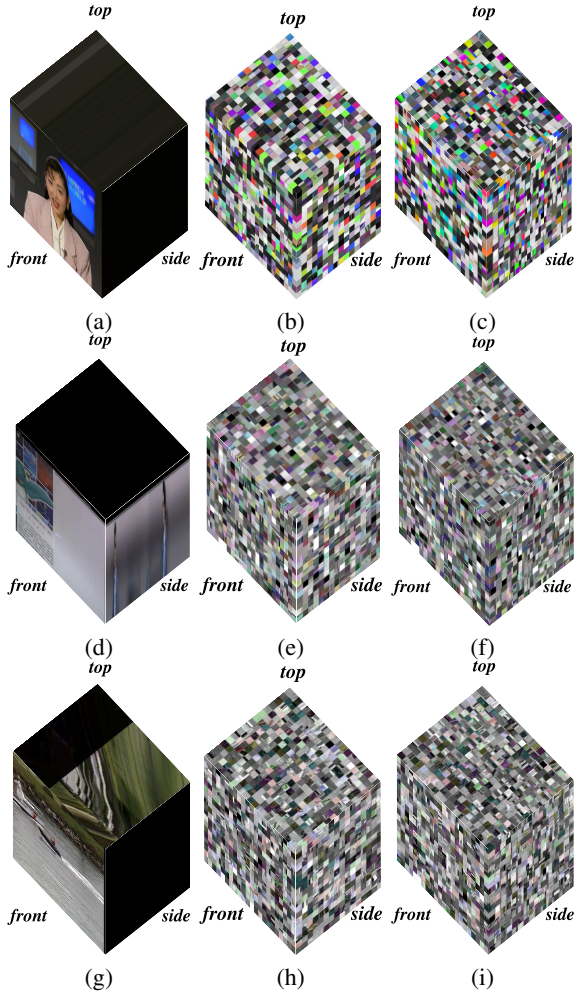

Fig. 7. Whole encrypted results: (left-to-right) original *Akiyo*, C-PE of *Akiyo*, LCd-PE of *Akiyo*, original *Bowing*, C-PE of *Bowing*, LCd-PE of *Bowing*, original *Coastguard*, C-PE of *Coastguard*, and LCd-PE of *Coastguard*.

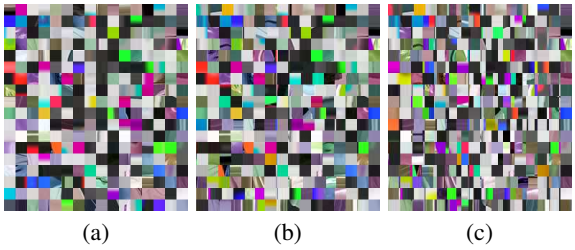

Fig. 8. Comparison of spatiotemporal images in *Akiyo*: (left-to-right) C-PE, C-PE+cuboid rotation+cuboid scrambling, and C-PE+cuboid rotation+cuboid scrambling+cuboid nega-posi reversal+cuboid color component shuffling.

random divisions are generated (Fig. 8 (c)). Therefore, we can see that the cuboid sizes become variable. The following features are also found:

- When the specified cuboid size is smaller, the Cd-PE method more finely subdivides the cubes: LCd-PE for MJPEG provides the best subdivision with $16 \times 16 \times 1$ cuboids and worst subdivision with $16 \times 16 \times$ the *number of frames*.
- The specified cuboid sizes for each Cd-PE method should not be multiples of each other.

TABLE I
TEST VIDEO SEQUENCES.

Input video sequence	*Akiyo*	*Bowing*	*Coastguard*
Moving area	small	medium	large
Stopping area	large	medium	small
Size & color depth	$288 \times 352 \times 256$,	8-bit RGB	

TABLE II
CUBOID SIZES USED IN LCD-PE.

Methods	$V \times H \times D$
Cuboid rotation	$16 \times 16 \times 65$
Cuboid scrambling	$16 \times 16 \times 113$
Cuboid nega-posi reversal	$16 \times 16 \times 17$
Cuboid color component shuffling	$16 \times 16 \times 57$

V. CONCLUSION

This paper proposed cuboid-based perceptual encryption (Cd-PE) and a version of conventional cube-based perceptual encryption (C-PE), named lapped cuboid-based perceptual encryption (LCd-PE), to enhance the security for Motion JPEG (MJPEG). LCd-PE subdivides the video sequence pre-encrypted with C-PE into small cuboids and further encrypts it so that attackers can not conduct conceivable attacks, such as a cube-based jigsaw puzzle solver (CJPS). The experiments showed that the compression performance of an encryption-then-compression (ETC) system with LCd-PE and MJPEG is almost equivalent to that of the C-PE case and yet achieves a the higher level of security.

ACKNOWLEDGMENT

This work was supported by a JSPS Grant-in-Aid for Young Scientists (B), Grant Number 16K18100.

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] B. Boyadjis, C. Bergeron, B. P. Popescu, and F. Dufaux, "Extended selective encryption of H.264/AVC (CABAC) and HEVC encoded video streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, pp. 892–906, Apr. 2017.

[3] B. Zeng, A. Yeung, S. Kei, 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.

[4] K. Shimizu and T. Suzuki, "Cube-based encryption connected prior to Motion JPEG standard," in *Proc. of APSIPA ASC 2017*, Kuala Lumpur, Malaysia, Dec. 2017, pp. 1–4.

[5] T. Chuman, K. Kurihara, and H. Kiya, "On the security of block scrambling-based etc systems against extended jigsaw puzzle solver attacks," *IEICE Trans. Inf. & Syst.*, vol. E101-D, no. 1, pp. 37–44, Jan. 2018.

[6] "Xiph.org Video Test Media [derf's collection]," https://media.xiph.org/video/derf/.

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