

On the Security of Block Scrambling-Based EtC Systems against Extended Jigsaw Puzzle Solver Attacks

Tatsuya CHUMAN^{†a)}, Kenta KURIHARA[†], *Nonmembers*, and Hitoshi KIYA^{†b)}, *Fellow*

SUMMARY The aim of this paper is to apply automatic jigsaw puzzle solvers, which are methods of assembling jigsaw puzzles, to the field of information security. Encryption-then-Compression (EtC) systems have been considered for the user-controllable privacy protection of digital images in social network services. Block scrambling-based encryption schemes, which have been proposed to construct EtC systems, have enough key spaces for protecting brute-force attacks. However, each block in encrypted images has almost the same correlation as that of original images. Therefore, it is required to consider the security from different viewpoints from number theory-based encryption methods with provable security such as RSA and AES. In this paper, existing jigsaw puzzle solvers, which aim to assemble puzzles including only scrambled and rotated pieces, are first reviewed in terms of attacking strategies on encrypted images. Then, an extended jigsaw puzzle solver for block scrambling-based encryption scheme is proposed to solve encrypted images including inverted, negative-positive transformed and color component shuffled blocks in addition to scrambled and rotated ones. In the experiments, the jigsaw puzzle solvers are applied to encrypted images to consider the security conditions of the encryption schemes.

key words: jigsaw puzzle, JPEG, encryption, EtC system

1. Introduction

The use of images and video sequences has greatly increased recently because of the rapid growth of the Internet and multimedia systems. A lot of studies on secure, efficient and flexible communications have been reported [1]–[3]. For securing multimedia data, full encryption with provable security (like RSA, AES, etc) is the most secure option. However, many multimedia applications have been seeking a trade-off in security to enable other requirements, e.g., low processing demands, retaining bitstream compliance, and signal processing in the encrypted domain, so that a lot of perceptual encryption schemes have been studied as one of the schemes for achieving the trade-off. They can also be combined with the encryption methods with provable security.

In this paper, we focus on Encryption-then-Compression (EtC) systems, although the traditional way of secure image transmission is to use a Compression-then-Encryption (CtE) system. EtC systems allow us to close unencrypted images to network providers, because encrypted images can be directly compressed even when the images

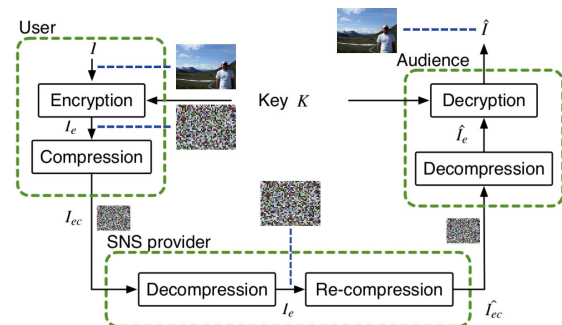


Fig. 1 Encryption-then-Compression system for image sharing in SNS

are multiply recompressed by providers. Therefore, we focus on block scrambling-based image encryption schemes, which have been proposed for EtC systems with the assumption of international compression standards, to consider the safety [4]–[8]. So far, the safety has been evaluated based on its key space assuming the brute-force attacks, so that the schemes have enough key spaces for protecting the attacks. However, each block in encrypted images has almost the same correlation as that of original images [9], [10]. Several efficient attacks on the permutation-only encryption have been studied [11], [12], but they are not available for the block scrambling-based encryption.

On the other hand, recently, jigsaw puzzle solvers, that utilize the correlation between pieces, have succeeded in solving puzzles with a large number of pieces [13]–[23]. In this paper, we regard the blocks of an encrypted image as pieces of a jigsaw puzzle and evaluate the safety of the encryption assuming the jigsaw puzzle solvers as crypto-attack methods. Existing jigsaw puzzle solvers are first reviewed in terms of attacking strategies on encrypted images. Then we point out to need new types of jigsaw puzzle solvers for the attacks, and propose a new solver to extend some limitations of conventional ones.

Finally, we evaluate the safety of the encryption by applying the jigsaw puzzle solvers to encrypted images. It is shown that some solvers can decrypt encrypted images even when the key space is large enough. On the other hand, it is also confirmed that an appropriate selection of the block size and the encryption methods makes the decryption of images difficult.

Manuscript received April 2, 2017.

Manuscript revised August 23, 2017.

Manuscript publicized October 16, 2017.

[†]The authors are with Tokyo Metropolitan University, Hino-shi, 191-0065 Japan.

a) E-mail: chuman-tatsuya@ed.tmu.ac.jp

b) E-mail: kiya@tmu.ac.jp

DOI: 10.1587/transinf.2017MUP0001

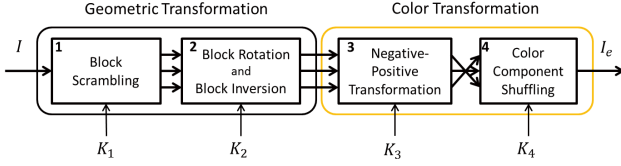


Fig. 2 Block scrambling-based image encryption

2. Preparation

2.1 Block Scrambling-Based Image Encryption

Block scrambling-based image encryption schemes have been proposed for EtC systems [5]–[8], in which a user wants to securely transmit image I to an audience, via a Social Networking Service (SNS) provider, as illustrated in Fig. 1. Since the user does not give the secret key K to the SNS provider, the privacy of image to be shared is under control of the user even when the SNS provider re-compresses image I . Therefore, the user is able to control image privacy for his own demand. On the other hand, in CtE systems, the user has to disclose unencrypted images to re-compress them.

In the schemes [4]–[8], an image with $X \times Y$ pixels is first divided into non-overlapped blocks with $B_x \times B_y$, then four block scrambling-based processing steps, as illustrated in Fig. 2, is applied to the divided image. The procedure of performing the image encryption to generate an encrypted image I_e is given as follows:

- Step1: Divide an image with $X \times Y$ pixels into blocks with $B_x \times B_y$ pixels, and permute randomly the divided blocks using a random integer generated by a secret key K_1 , where K_1 is commonly used for all color components.
- Step2: Rotate and invert randomly each block (see Fig. 3) using a random integer generated by a key K_2 , where K_2 is commonly used for all color components as well.
- Step3: Apply the negative-positive transformation to each block using a random binary integer generated by a key K_3 , where K_3 is commonly used for all color components. In this step, a transformed pixel value in i th block B_i , p' is computed by

$$p' = \begin{cases} p & (r(i) = 0) \\ p \oplus (2^L - 1) & (r(i) = 1) \end{cases} \quad (1)$$

where $r(i)$ is a random binary integer generated by K_3 and $p \in B_i$ is the pixel value of an original image with L bpp.

- Step4: Shuffle three color components in each block (the color component shuffling) using a random senary integer generated by a key K_4 .

2.2 Key Space Analysis

If an image with $X \times Y$ pixels is divided into blocks with

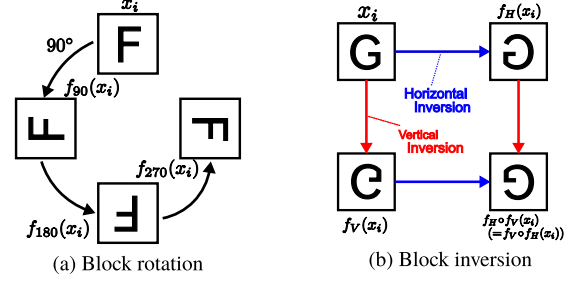


Fig. 3 Block rotation and inversion

$B_x \times B_y$ pixels, the number of blocks n is given by

$$n = \lfloor \frac{X}{B_x} \rfloor \times \lfloor \frac{Y}{B_y} \rfloor \quad (2)$$

where $\lfloor \cdot \rfloor$ is the function that rounds down to the nearest integer.

The key space of the block scrambling (Step1) N_S , which is the number of permutation of n blocks, is given by

$$N_S = {}_n P_n = n!. \quad (3)$$

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

$$N_R = 4^n, N_I = 4^n, N_{R\&I} = 8^n \quad (4)$$

$$N_N = 2^n, N_C = ({}_3 P_3)^n = 6^n \quad (5)$$

where N_R and N_I are the key spaces of the block rotation and block inversion, and $N_{R\&I}$ is the key space of the encryption combining them (Step2). Note that $N_{R\&I}$ is the key space considering the collision between block rotation and inversion. Namely, rotating pieces 180 degrees is the same operation as inverting ones horizontally and vertically. N_N and N_C are the key spaces of the negative-positive transformation (Step3) and the color component shuffling (Step4) respectively. Consequently, the key space of encrypted images by using all the proposed encryption steps, N_A , is represented by

$$N_A = N_S \cdot N_{R\&I} \cdot N_N \cdot N_C = n! \cdot 8^n \cdot 2^n \cdot 6^n. \quad (6)$$

The key space is expanded by combining some independent encryption steps. As a result, when an encrypted image has more than $n = 28$ blocks, the key space of the image is larger than that of the 256-bit key. Thus, the key space of the scheme is generally large enough against brute-force attacks.

However, an encrypted image has almost the same correlation among pixels in each block as that of the original image, whose property enables to efficiently compress images. Therefore, an attacker can utilize the correlation to decrypt the image in some way. The aim of this paper is to discuss the security of the encryption against jigsaw puzzle solver attacks based on the correlation.

Jigsaw puzzle solver attacks are discussed in addition to brute-force attacks, as ciphertext-only attack (COA). Other attacking strategies such as known-plaintext attack (KPA) and chosen-plaintext attack (CPA) should be considered for the security. Block scrambling-based image encryption becomes robust against KPA by assigning a different key to each image for the encryption. Besides, the keys used for encryption do not need to be disclosed because the encryption scheme is not public key cryptography. Therefore, the encryption can avoid the CPA unlike public key cryptography.

3. Extended Jigsaw Puzzles Solver

Jigsaw puzzle solver is a method of assembling jigsaw puzzles. In the block scrambling-based encryption, if we regard the blocks as pieces of a jigsaw puzzle, decrypting encrypted images is similar to assembling the jigsaw puzzle. Therefore, jigsaw puzzle solvers are considered as one of the attack methods on the block scrambling-based encryption in this paper.

3.1 Related Works

Jigsaw puzzle solvers are broadly classified into three categories according to their assembly strategies, i.e., greedy methods, global methods and their hybrid methods [23]. The greedy methods start from initial pairwise matches and successfully build larger and larger components. On the other hand, the global methods directly search for a solution by maximizing a global compatibility function. Table 2 shows typical solvers. For example, the jigsaw puzzle solver [22] completely succeeded in assembling large puzzles which consist of 30745 pieces with the size of 28×28 , in 2016.

Table 1 Permutation of color components for a random integer

Random Integer	R	G	B	Transform Function f
0	R	G	B	f_{RGB}
1	G	R	B	f_{GRB}
2	R	B	G	f_{RBG}
3	B	G	R	f_{BGR}
4	B	R	G	f_{BRG}
5	G	B	R	f_{GBR}

Table 2 A summary of latest square jigsaw puzzle solvers. “o” indicates support for puzzles with unknown rotation or inversion. “x” indicates no support for puzzles with unknown rotation or inversion.

Methods	Authors	Rotation	Inversion	Year	Pieces	Piece size
Greedy	Pomeranz [13]	x	x	2011	3300	28×28
	Gallagher [14]	o	x	2012	9600	28×28
	Mondal [15]	o	x	2013	540	28×28
	Son [16]	o	x	2014	9801	28×28
	Son [16]	o	x	2014	221	10×10
	Paikin [17]	o	x	2015	22755	28×28
	Son [18]	o	x	2016	3300	28×28
	Son [18]	o	x	2016	1064	14×14
Global	Cho [19]	x	x	2010	432	28×28
	Andalo [20]	x	x	2012	3300	28×28
	Sholomon [21]	o	x	2014	22755	28×28
	Sholomon [22]	x	x	2016	30745	28×28
Hybrid	Rui [23]	o	x	2015	3300	28×28

On the other hand, a solver for puzzles including rotated pieces (pieces with unknown orientation) was first proposed in 2012 [14]. Thus, even when the number of blocks in an encrypted image is larger than 22755, there is a possibility that the image is completely decrypted if the piece size is large. In this paper, jigsaw puzzle solvers are considered as one of attacks on the image encryption.

3.2 Jigsaw Puzzle Solver for New Type Puzzles

Although puzzles including scrambled and rotated pieces were already considered, the existing jigsaw solvers do not support inverted, color component shuffled or negative-positive transformed pieces. Therefore, we define new types of jigsaw puzzles, as shown in Table 3, where Type I, Type N, Type IN and Type INC are new. Examples of encrypted images are illustrated in Fig. 4 (b), (c) and (d), where Fig. 4 (a) is the original one. As shown in Fig. 4 (f), (g) and (h), recognizing objects in three assembled images is difficult unlike Fig. 4 (e).

In this paper, the greedy method [14] will be extended as a new jigsaw puzzles solver to assemble puzzles including inverted pieces, negative-positive transformed ones or component shuffled ones. Many papers have selected the jigsaw puzzle solver proposed by Gallagher [14] as a benchmark. Therefore we use this solver as in other works [13]–[23] to compare the proposed method with conventional ones. Figure 5 shows the process of assembling jigsaw puzzles by using proposed jigsaw puzzle solver. The following is the procedure for solving the square jigsaw puzzles which consist of pieces with the size of $P \times P$.

3.2.1 Pairwise Compatibility

To calculate pairwise compatibility between pieces, we use Mahalanobis Gradient Compatibility (MGC) proposed by Gallagher [14]. Given the pieces x_i and x_j , $i, j = 1, 2, \dots, n$, the compatibility between the right side of x_i and the left side of x_j is expressed as $C_{LR}(x_i, x_j)$. The intensity of the pixel at location (s, t) , $s, t \in \{1, 2, \dots, P\}$ in x_i is defined as $x_i(s, t, c)$, $c \in \{R, G, B\}$, where P is the piece size. Therefore, the right edge of x_i is defined as $x_i(s, P, c)$, as shown in Fig. 6. The detail of calculating $C_{LR}(x_i, x_j)$ will be described below.

The color gradient near the right side of x_i is defined by

$$G_{iL}(s, c) = x_i(s, P, c) - x_i(s, P - 1, c). \quad (7)$$

Table 3 Jigsaw puzzle types

Type	Scramble	Rotation	Inversion	Negative- Positive Transformation	Color Component Shuffling
Type 1	✓				
Type 2	✓	✓			
Type I	✓	✓	✓		
Type N	✓	✓		✓	
Type IN	✓	✓	✓	✓	
Type INC	✓	✓	✓	✓	✓

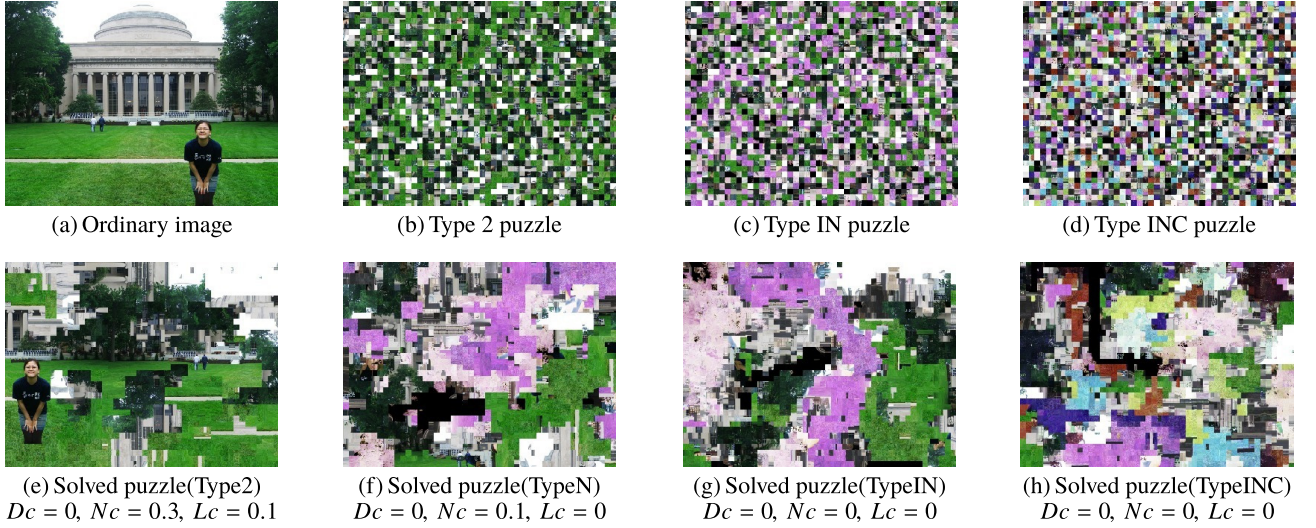


Fig. 4 Examples of encrypted images and assembled images ($n = 1728$, $B_x \times B_y = 14 \times 14$)

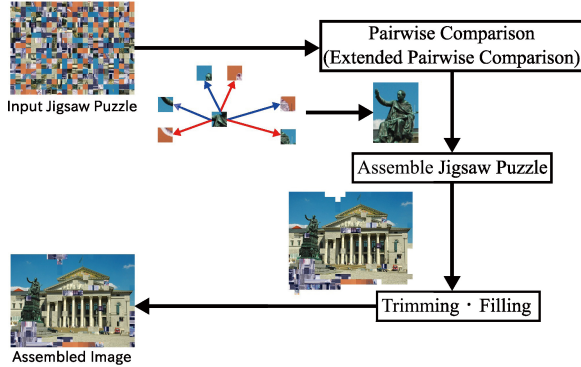


Fig. 5 The process of assembling jigsaw puzzles

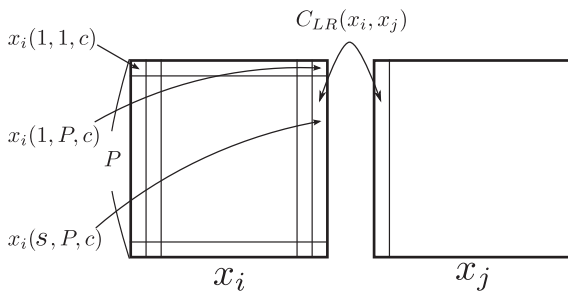


Fig. 6 The example of compatibility between x_i and x_j

The average gradient $\mu_{iL}(c)$ is also calculated by

$$\mu_{iL}(c) = \frac{1}{P} \sum_{s=1}^P G_{iL}(s, c). \quad (8)$$

The gradient from the right edge of x_i to the left edge of x_j is defined below.

$$G_{ijLR}(n, c) = x_j(n, 1, c) - x_i(n, P, c) \quad (9)$$

A vector with three elements G_{ijLR} is given by

$$G_{ijLR}(n) = [G_{ijLR}(n, R) - \mu_{iL}(R), G_{ijLR}(n, G) - \mu_{iL}(G), G_{ijLR}(n, B) - \mu_{iL}(B)]^T. \quad (10)$$

Using Eq. (10) and S_{iL} , the compatibility from the right side of x_i to the left side of x_j is defined as

$$D_{LR}(x_i, x_j) = \sum_{n=1}^P G_{ijLR}(n)^T S_{iL}^{-1} G_{ijLR}(n) \quad (11)$$

where S_{iL} is the 3×3 covariance of $G_{iL}(s, c)$.

Finally, the symmetric compatibility between the right side of x_i and the left side of x_j is computed by

$$C_{LR}(x_i, x_j) = D_{LR}(x_i, x_j) + D_{RL}(x_j, x_i). \quad (12)$$

Given the Type 1 puzzle pieces x_i and x_j , four compatibilities will be calculated such as $C_{UD}(x_i, x_j)$, $C_{DU}(x_i, x_j)$ and $C_{RL}(x_i, x_j)$ in addition to $C_{LR}(x_i, x_j)$. For example, $C_{UD}(x_i, x_j)$ is the compatibility between the down side of x_i and the up side of x_j .

MGC can be also applied to hybrid methods as well as greedy methods [18], [23]. Next, we apply MGC to extended pairwise comparisons for new puzzle types (Type I, N, IN and Type INC).

3.2.2 Extended Pairwise Comparison

We define the transform function that rotates x_j 0° , 90° , 180° or 270° as $f_R(x_j)$, $R \in \{0, 90, 180, 270\}$ shown in Fig. 3 (a). Using this function for rotated pieces (i.e. Type 2 puzzles), Eq. (12) is extended as

$$C_{LR}(x_i, f_R(x_j)) = D_{LR}(x_i, f_R(x_j)) + D_{RL}(x_j, f_R(x_i)). \quad (13)$$

Equation (13) is reduced to Eq. (12) under the condition of $R = 0$. While the number of calculating compatibilities of Type 1 puzzles is $2n(n-1)$ times, $8n(n-1)$ times is needed to solve Type 2 puzzles.

The function that inverts x_j horizontally(H) or vertically(V) is defined as $f_I(x_j)$, $I \in \{H, V, 0\}$ as in Fig. 3 (b), where $f_0(x_j)$ is the function that indicates non-inverted. $f_N(x_j)$, $N \in \{N, 0\}$ is the function whether applies negative-positive transformation(N) to x_j . In accordance with Table 1, the function that applies x_j to color component shuffling is given as $f_C(x_j)$, $C \in \{RGB, GRB, RBG, BGR, BRG, GBR\}$. Finding the correct pairwise adjacencies become available even if pieces were encrypted respectively by definition of transform functions as well as Eq. (13).

In addition to four transform functions, i.e., $f_R(x_j)$, $f_I(x_j)$, $f_N(x_j)$, $f_C(x_j)$, the combination of them gives other transformations. Then, a rotated, inverted, negative-positive transformed and color component shuffled piece is represented as $f_R \circ f_I \circ f_N \circ f_C(x_j)$, which is the composite function of the four transform functions. For example, when a jigsaw piece x_j is inverted horizontally and negative-positive transformed, the transformed piece is represented as $f_0 \circ f_H \circ f_N \circ f_{RGB}(x_j)$. Therefore, the compatibility between the right side of x_i and the left side of x_j which was inverted horizontally and negative-positive transformed is given as

$$\begin{aligned} C_{LR}(x_i, f_0 \circ f_H \circ f_N \circ f_{RGB}(x_j)) = \\ D_{LR}(x_i, f_0 \circ f_H \circ f_N \circ f_{RGB}(x_j)) \\ + D_{RL}(x_j, f_0 \circ f_H \circ f_N \circ f_{RGB}(x_i)). \end{aligned} \quad (14)$$

The process of finding the piece x_k , $k = 1, 2, \dots, n$ which has the minimum compatibility with x_i is described below. The minimum compatibility for right side of x_i is defined by

$$\min C_{LR}(x_i, x_j) = \min_{f_{R,I,N,C}} \{C_{LR}(x_i, f_R \circ f_I \circ f_N \circ f_C(x_j))\} \quad (15)$$

Given x_i and x_j , four compatibilities will be calculated such as $\min C_{LR}(x_i, x_j)$, $\min C_{UD}(x_i, x_j)$, $\min C_{DU}(x_i, x_j)$ and $\min C_{RL}(x_i, x_j)$. The piece x_k which has the minimum compatibility with x_i is found by calculating the compatibility between x_i and x_j ($j \neq i$) as follows

$$x_k = \operatorname{argmin}_j \{ \min C_{LR}(x_i, x_j), \min C_{UD}(x_i, x_j), \\ \min C_{DU}(x_i, x_j), \min C_{RL}(x_i, x_j) \}. \quad (16)$$

In the proposed solver, the jigsaw puzzles including inverted pieces, negative-positive transformed pieces and color component shuffled pieces could be solved.

These procedures enable to find the minimum pairwise piece x_k for x_i . These minimum compatibility values are used to assemble jigsaw puzzle by using the tree-based assembly method [14]. Finally, the procedures of trimming and filling are applied to the assembled images to produce the same size images as input ones. As shown in Fig. 5, the same approach as Gallagher [14] is used in this paper except for the step of calculating compatibilities.

4. Experiments and Results

4.1 Experimental Conditions

Image I_d assembled by jigsaw puzzle solvers from Type I,

N, IN or Type INC puzzle was compared with the original image I . The following three measures [14], [19] were used to evaluate the results.

Direct comparison (Dc): represents the ratio of the number of pieces which are in the correct position. Dc for image I_d , namely, $Dc(I_d)$ is calculated as

$$\begin{aligned} Dc(I_d) &= \frac{1}{n} \sum_{i=1}^n d_c(i), \\ d_c(i) &= \begin{cases} 1, & \text{if } I_d(i) \text{ is in the correct position} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (17)$$

where $I_d(i)$ represents the position of a piece i in image I_d .

Neighbor comparison (Nc): is the ratio of the number of correctly joined blocks. Nc for image I_d , namely, $Nc(I_d)$ is calculated as

$$\begin{aligned} Nc(I_d) &= \frac{1}{B} \sum_{k=1}^B n_c(k), \\ n_c(k) &= \begin{cases} 1, & \text{if } b_k \text{ is joined correctly} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (18)$$

where B is the number of boundaries among pieces in I_d , and b_k is the k th boundary in I_d . For an image with $u \times v$ blocks, there are $B = 2uv - u - v$ boundaries in the image.

Largest Component (Lc): is the ratio of the number of the largest joined blocks which have correct adjacencies to the number of blocks in an image. Lc for image I_d , namely, $Lc(I_d)$ is calculated as

$$Lc(I_d) = \frac{1}{n} \max_j \{l_c(I_d, j)\}, j = 1, 2, \dots, m \quad (19)$$

where $l_c(I_d, j)$ is the number of blocks in the j th partial correctly assembled area, and m is the number of partial correctly assembled areas.

In the measures, $Dc(I_d)$, $Nc(I_d)$, $Lc(I_d) \in [0, 1]$, a larger value means a higher compatibility as illustrated in Fig. 4.

We used 20 images from MIT dataset, provided by Cho [19]. The most papers about jigsaw puzzle solvers [13]–[23] aim to solve puzzles where the piece size is $Bx = By = 14, 28$ pixels. Therefore, the two piece sizes ($Bx = By = 14, 28$) are used to compare other papers in the field of jigsaw puzzle solvers. Note that we already attempted to assemble the puzzles ($Bx = By = 16$) and the result was very similar to puzzles ($Bx = By = 14$). Ten different encrypted images were generated by random keys from one ordinary image for each Type puzzle. We assembled the encrypted images by using jigsaw puzzle solvers and chose the image which had the highest sum of $Dc(I_d)$, $Nc(I_d)$ and $Lc(I_d)$ in those of ten images. We performed these procedures for each type puzzle independently, and the average of 20 images was calculated for $Dc(I_d)$, $Nc(I_d)$ and $Lc(I_d)$.

Table 4 Evaluation of the existing jigsaw puzzle solver [14] (n = 432)

Piece Size	28 × 28 pixels					14 × 14 pixels				
	Type 2	Type I	Type N	Type IN	Type INC	Type 2	Type I	Type N	Type IN	Type INC
$Dc(I_d)$ (Average)	0.822	0.581	0.021	0.013	0.014	0.377	0.077	0.103	0.013	0.013
$Nc(I_d)$ (Average)	0.904	0.519	0.136	0.066	0.016	0.626	0.226	0.088	0.050	0.015
$Lc(I_d)$ (Average)	0.889	0.614	0.125	0.058	0.021	0.551	0.203	0.067	0.042	0.019

Table 5 Evaluation of the extended jigsaw puzzle solver (n = 432)

Piece Size	28 × 28 pixels				14 × 14 pixels			
	Type I	Type N	Type IN	Type INC	Type I	Type N	Type IN	Type INC
$Dc(I_d)$ (Average)	0.702	0.821	0.459	0.067	0.074	0.296	0.028	0.011
$Nc(I_d)$ (Average)	0.696	0.813	0.563	0.148	0.218	0.437	0.169	0.049
$Lc(I_d)$ (Average)	0.735	0.837	0.590	0.171	0.194	0.437	0.167	0.050

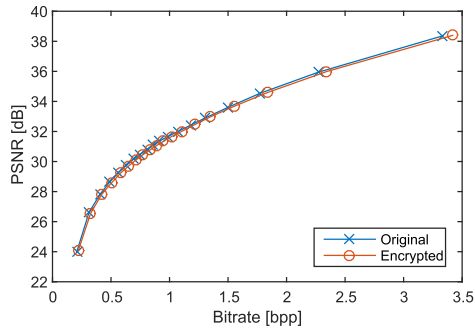


Fig. 7 RD curves of original images and encrypted ones ($X \times Y = 672 \times 504$, $B_x \times B_y = 16 \times 16$)

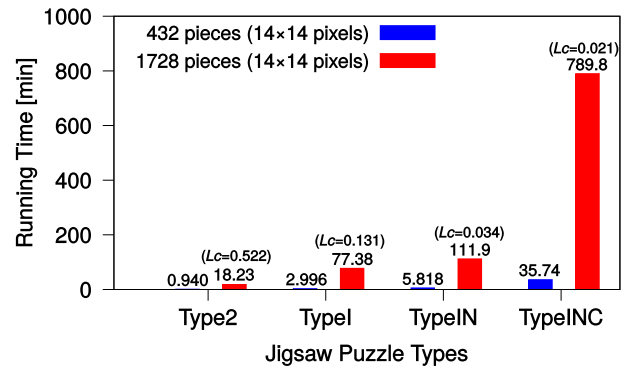


Fig. 8 Running time of assembling various jigsaw puzzles by the extended jigsaw puzzle solver

4.2 Experiment Result

A. Compression performance of the EtC system

Figure 7 shows the Rate-Distortion (RD) curves of JPEG compressed images without any encryption and with the block scrambling-based image encryption [5], [6], where the average bitrate and PSNR values of 20 images were plotted, after decrypting the images. It is certified that the compression efficiency of the encrypted images is approximately equivalent to that of the original images. Therefore, it is certified that images encrypted by the block scrambling-based image encryption are not affected by the JPEG compression.

B. Existing jigsaw Puzzles Solver

Table 4 shows the scores of images assembled by the existing jigsaw puzzle solver [14], which is a well-known benchmark method. It is shown that the use of encrypted images with small size of blocks makes assembling the images difficult. Thereby, it is important to select the appropriate block size for the security of the encryption. Also, the assembly of encrypted images including inverted, negative-positive transformed or color component shuffled pieces is more difficult than that of Type 2 puzzles because the existing solvers do not support these pieces.

C. Extended jigsaw Puzzles Solver

Table 5 summarizes the scores of the extended jigsaw puzzles solvers discussed in 3.2. As shown in Table 5, Type IN puzzles ($B_x \times B_y = 28 \times 28$) were assembled about 0.5 (as N_c or L_c) by using the extended jigsaw puzzle solver. However, in the case of Type INC puzzles, the scores become much lower than other types. It is confirmed that combining the encryption steps makes puzzle solvers more difficult than single use of each step. The Type IN puzzles ($B_x \times B_y = 14 \times 14$) could be assembled only 0.15 (as N_c or L_c). This result also confirmed that the scores of the solvers strongly depend on the size of pieces.

D. Running time to assemble jigsaw puzzles

Figure 8 indicates the running time to assemble jigsaw puzzles by using the extended jigsaw puzzle solver, where the average time of assembling 20 images was plotted. The algorithms were implemented in MATLAB2015a on a PC with a 3.2GHz processor and a main memory of 8Gbytes (Processor: Intel Core i5-6500 3.2GHz, OS: Ubuntu 16.04 LTS). As shown in Fig. 8, although Type INC puzzles with 432 pieces were solved in 35 minutes, the scores of these images are very low as $L_c = 0.050$ (see Table 5). As mentioned in Sect. 3.2.2, it is the reason that the number of calculating compatibilities increases due to many encryption

steps. As a result, it is more difficult to assemble puzzles in terms of both the computational complexity and the accurate correlation calculation than Type 2 puzzles. Compared to jigsaw puzzles with 432 pieces, the running time of assembling puzzles with 1728 pieces increased exponentially. For example, about 13 hours were needed to assemble Type INC puzzles for 1728 pieces. Moreover, the scores of assembled puzzles decrease in accordance with the increase in the number of pieces as $L_c = 0.021$. It was also confirmed that the running time of encryption and decryption still remains small, even when the number of pieces increases.

5. Conclusion

In this paper, the safety of the block-scrambling based image encryption schemes was discussed. We focused on jigsaw puzzle solvers as one of attack methods on the encryption, and regarded blocks of an encrypted image as pieces of a jigsaw puzzle, although the safety has been evaluated so far on the size of the key space, assuming the brute-force attacks. Moreover, an existing jigsaw puzzle solver was extended to be adapted to the encryption schemes, and some jigsaw puzzle solvers including the proposed one were applied to encrypted images. In the simulations, the studies presented evidence that the appropriate selection of the block size and the combination of each encryption step can improve the strength of EtC against jigsaw puzzle solver attacks.

Acknowledgements

This work was partially supported by Grant-in-Aid for Scientific Research(B), No.17H03267, from the Japan Society for the Promotion Science.

References

- [1] C.T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varadharajan, and C.-C.J. Kuo, "Survey on securing data storage in the cloud," *APSIPA Transactions on Signal and Information Processing*, vol.3, p.e7, 2014.
- [2] R.L. Legendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Process. Mag.*, vol.30, no.1, pp.82–105, 2013.
- [3] J. Zhou, X. Liu, O.C. Au, and Y.Y. Tang, "Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation," *IEEE Trans. Inf. Forensics Security*, vol.9, no.1, pp.39–50, 2014.
- [4] O. Watanabe, A. Uchida, T. Fukuhara, and H. Kiya, "An encryption-then-compression system for jpeg 2000 standard," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.1226–1230, 2015.
- [5] K. Kurihara, S. Shiota, and H. Kiya, "An encryption-then-compression system for jpeg standard," in *Picture Coding Symposium (PCS)*, pp.119–123, 2015.
- [6] 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, 2015.
- [7] K. Kurihara, O. Watanabe, and H. Kiya, "An encryption-then-compression system for jpeg xr standard," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp.1–5, 2016.
- [8] K. Kurihara, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for lossless image compression standards," *IEICE Trans. Inf. & Syst.*, vol.E100-D, no.1, pp.52–56, 2017.
- [9] T. Chuman, K. Kurihara, and H. Kiya, "On the security of block scrambling-based etc systems against jigsaw puzzle solver attacks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.2157–2161, 2017.
- [10] T. Chuman, K. Kurihara, and H. Kiya, "Security evaluation for block scrambling-based etc systems against extended jigsaw puzzle solver attacks," in *IEEE International Conference on Multimedia and Expo (ICME)*, pp.229–234, 2017.
- [11] C. Li and K.-T. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal processing*, vol.91, no.4, pp.949–954, 2011.
- [12] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Trans. Inf. Forensics Security*, vol.11, no.2, pp.235–246, 2016.
- [13] D. Pomeranz, M. Shemesh, and O. Ben-Shahar, "A fully automated greedy square jigsaw puzzle solver," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.9–16, 2011.
- [14] A.C. Gallagher, "Jigsaw puzzles with pieces of unknown orientation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.382–389, 2012.
- [15] D. Mondal, Y. Wang, and S. Durocher, "Robust solvers for square jigsaw puzzles," in *Proceedings of the International Conference on Computer and Robot Vision (CRV)*, pp.249–256, 2013.
- [16] K. Son, J. Hays, and D.B. Cooper, "Solving square jigsaw puzzles with loop constraints," in *European Conference on Computer Vision (ECCV)*, vol.8694, pp.32–46, 2014.
- [17] G. Paikin and A. Tal, "Solving multiple square jigsaw puzzles with missing pieces," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4832–4839, 2015.
- [18] K. Son, D. Moreno, J. Hays, and D.B. Cooper, "Solving small-piece jigsaw puzzles by growing consensus," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1193–1201, 2016.
- [19] T.S. Cho, S. Avidan, and W.T. Freeman, "A probabilistic image jigsaw puzzle solver," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.183–190, 2010.
- [20] F.A. Andalo, G. Taubin, and S. Goldenstein, "Solving image puzzles with a simple quadratic programming formulation," in *Graphics, Patterns and Images (SIBGRAPI)*, pp.63–70, 2012.
- [21] D. Sholomon, O.E. David, and N.S. Netanyahu, "A generalized genetic algorithm-based solver for very large jigsaw puzzles of complex types," in *National Conference on Artificial Intelligence (AAAI)*, pp.2839–2845, 2014.
- [22] D. Sholomon, O.E. David, and N.S. Netanyahu, "An automatic solver for very large jigsaw puzzles using genetic algorithms," *Genetic Programming and Evolvable Machines*, vol.17, no.3, pp.291–313, 2016.
- [23] R. Yu, C. Russell, and L. Agapito, "Solving jigsaw puzzles with linear programming," *arXiv preprint arXiv:1511.04472*, 2015.



Tatsuya Chuman received his B.Eng. degree from Toyo University, Japan in 2016. From 2016, he has been a Master course student at Tokyo Metropolitan University. His research interests are in the area of image processing.



Kenta Kurihara received his B.Eng. and M.Eng. degrees from Tokyo Metropolitan University, Japan in 2015 and 2017, respectively. He graduated from Tokyo Metropolitan University, Japan in 2017. His research interests include image processing.



Hitoshi Kiya received his B.Eng. and M.Eng. degrees from Nagaoka University of Technology, Japan, in 1980 and 1982, respectively, and his D.Eng. degree from Tokyo Metropolitan University in 1987. In 1982, he joined Tokyo Metropolitan University as an Assistant Professor, where he became a Full Professor in 2000. From 1995 to 1996, he attended the University of Sydney, Australia as a Visiting Fellow. He was/is the Chair of IEEE Signal Processing Society Japan Chapter, an Associate

Editor for IEEE Trans. Image Processing, IEEE Trans. Signal Processing and IEEE Trans. Information Forensics and Security, respectively. He also served as the President of IEICE Engineering Sciences Society (ESS), the Editor-in-Chief for IEICE ESS Publications, and a Vice President of APSIPA. He currently serves as the President-Elect of APSIPA and Regional Director-at-Large for Region 10 of IEEE Signal processing Society. He received IEEE ISPACS Best Paper Award in 2016, IWAIT Best Paper Award in 2014 and 2015, ITE Niwa-Takayanagi Best Paper Award in 2012, the Telecommunications Advancement Foundation Award in 2011, and IEICE Best Paper Award in 2008. He is a Fellow of IEEE, IEICE and ITE.