

Image Manipulation on Social Media for Encryption-then-Compression Systems

Tatsuya Chuman* and Kenta Iida* and Hitoshi Kiya*

* Tokyo Metropolitan University, Tokyo, 191-0065, Japan

Abstract—Encryption-then-Compression (EtC) systems have been proposed to securely transmit images through an untrusted channel provider. In this paper, the EtC systems are applied to social media like Twitter, that are known for carrying out some image manipulation. Block scrambling-based encryption schemes used in the EtC systems are evaluated in terms of the robustness against image manipulation on social media. This work aims to investigate how each social networking service (SNS) provider manipulates images, and to consider whether the encrypted images uploaded to SNS providers can avoid to include some distortion under the image manipulation. In the experiment, encrypted and non-encrypted JPEG images are uploaded to various SNS providers to confirm the robustness of EtC systems. It is shown that the EtC systems are applicable to almost all SNS providers.

I. 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 [4]–[8].

Image encryption is sometimes required to be performed prior to image compression in certain practical scenarios such as secure image transmission through an untrusted channel provider. This framework is carried out by EtC systems [3], [9], [10]. In this paper, we focus on the EtC systems, although the traditional way of securely transmit images is to use a Compression-then-Encryption (CtE) system. However, most studies on EtC systems assume the use of their own compression schemes that have no compatibility with international standards such as JPEG [3], [11]–[14]. In this paper, we focus on block scrambling-based image encryption schemes which have the compatibility with international compression standards [15]–[19].

On the other hand, almost all SNS providers support the JPEG standard, as one of the most widely used image compression standards [20]. However, whether the EtC systems are applicable to SNS providers or not has never been confirmed. Although some papers have studied image manipulation on social media [21]–[23], e.g., alternation of image filenames

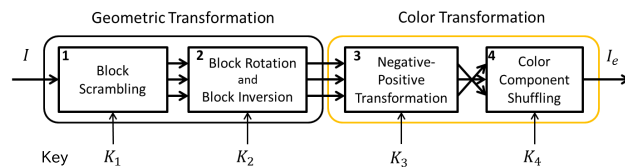


Fig. 1: Block scrambling-based image encryption

or the header of JPEG images, recompression parameters and the conditions of image manipulation have been still unclear. Therefore, we investigate how SNS providers manipulate images, and whether the EtC systems are applicable to SNS providers.

Finally, we upload a lot of images to each SNS provider to confirm the robustness of EtC systems. It is shown that encrypted images including some block distortion due to image manipulation on SNS providers heavily reduce the quality of decrypted images. Otherwise, it is also confirmed that the EtC systems are applicable to almost all SNS providers.

II. PREPARATION

A. Block scrambling-based image encryption

Block scrambling-based image encryption schemes have been proposed for EtC systems [16]–[19]. In the schemes [15]–[19], an image with $X \times Y$ pixels is first divided into non-overlapped blocks with $B_x \times B_y$ pixels, the number of blocks n is given by

$$n = \lfloor \frac{X}{B_x} \rfloor \times \lfloor \frac{Y}{B_y} \rfloor \quad (1)$$

where $\lfloor \cdot \rfloor$ is the function that rounds down to the nearest integer. Then, four block scrambling-based processing steps, as illustrated in Fig. 1, 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. In this paper, $B_x = B_y = 16$ is used as well as in [17].
- Step2: Rotate and invert randomly each block (see Fig. 2) using a random integer generated by a key K_2 , where K_2 is commonly used for all color components as well.

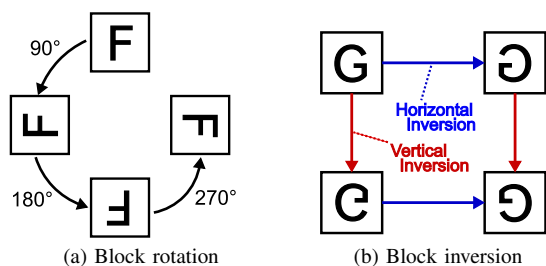


Fig. 2: Block rotation and inversion

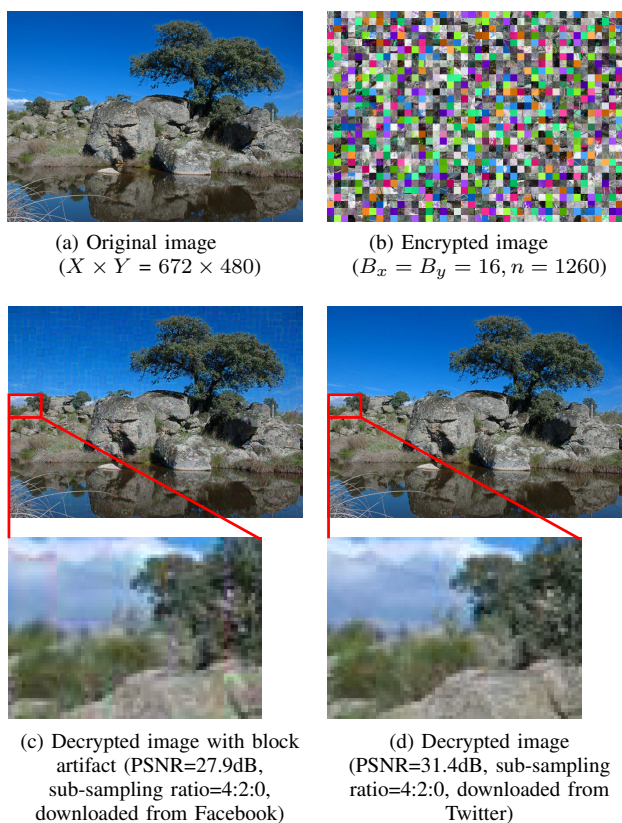


Fig. 3: Examples of encrypted image and decrypted images, downloaded from Facebook and Twitter

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

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

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 .

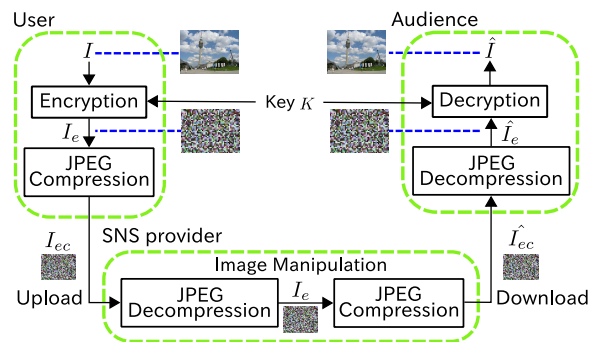


Fig. 4: EtC system

An example of encrypted images is illustrated in Fig. 3(b) where Fig. 3(a) is the original one. The key space of the block scrambling-based image encryption is generally large enough against the brute-force attacks [17]. On the other hand, jigsaw puzzle solver attacks, which utilize the correlation among pixels in each block, have been considered [24], [25]. It is confirmed that the appropriate selection of the block size and the combination of each encryption step can improve the strength of EtC systems.

B. Application to social media

Figure 4 illustrates the scenario of this paper, where a user wants to securely transmit image I to an audience, via a SNS provider. 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 decompresses image I . Therefore, the user is able to protect the privacy by him/herself. Even if encrypted images saved in the SNS servers are leaked by malicious users, the third party and general audiences could not see these images visually unless they have key.

Meanwhile, it is known that almost all SNS providers manipulate images uploaded by users, e.g., rescaling image resolution and recompressing with different parameters, for decreasing the data size of ones. Encrypting images might generate images with some distortion like Fig. 3(c) due to forcedly image manipulation by SNS providers. Although numerous papers were examined to clarify conditions for resizing images [22], [23], recompression parameters and conditions have been yet unpublished by SNS providers and researchers. Therefore, we investigate how each SNS provider manipulates images uploaded by users to apply EtC systems to social media.

III. IMAGE MANIPULATION AND ROBUSTNESS

In this section, we examine how each SNS provider manipulates images uploaded by users. Then, the conditions to avoid block distortion are discussed for applying EtC systems to social media.

A. Image manipulation on social media

We focus on two key points regarding image manipulation. The first point is the maximum resolution of uploaded images and the second is the parameters of recompression.

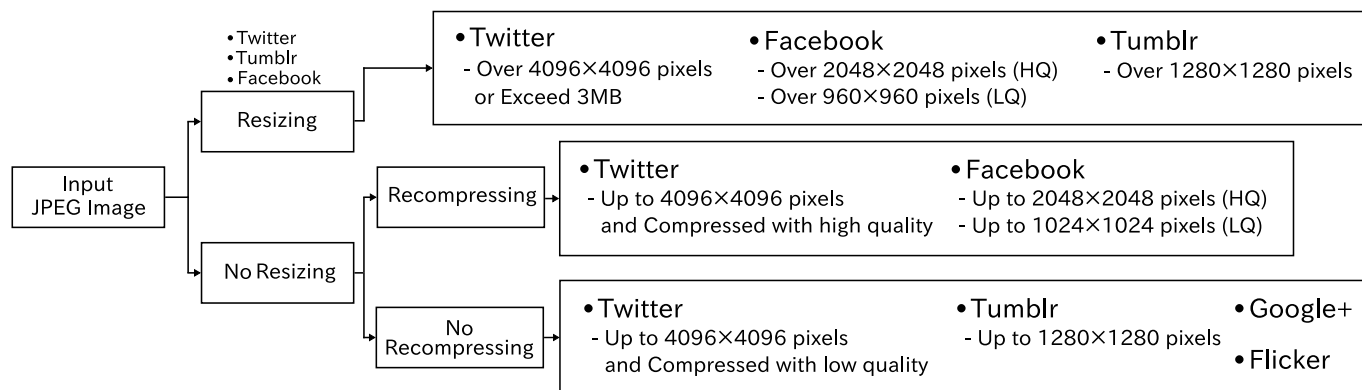


Fig. 5: Image manipulation on social media

(1) Maximum resolution

It is known that a number of SNS providers resize automatically images uploaded by users, when the size of images is over the maximum one set by the providers [22], [23]. When the resolution of encrypted images is changed in the encrypted domain, decrypting the images downloaded from providers become difficult.

Figure 5 shows the classification of SNS providers in terms of recompression and resizing. Twitter, Facebook and Tumblr apply resizing algorithms to uploaded images, if the images satisfy the following conditions. Twitter resizes images when uploaded images are over 4096×4096 pixels or exceeded 3MB. Facebook has two modes to control the maximum resolution, i.e., Low Quality (LQ) and High Quality (HQ). The selection of LQ enables users to upload up to images with the size of 960×960 without any resizing. Meanwhile, HQ allows to upload up to 2048×2048 pixels. Tumblr also resizes when the resolution of uploaded images is over 1280×1280 pixels. Unlike these three providers, Google+ and Flickr do not carry out any resizing operation, even when the resolution of uploaded images is large.

(2) Recompression

Next, we investigate how each SNS provider recompresses images uploaded by users. As illustrated in Fig. 3, the quality of images downloaded from providers depends on the providers. Block artifacts in the decrypted images might be generated due to the recompression, as shown in Fig. 3(c).

In terms of recompression, SNS providers are divided into two groups as shown in Fig. 5. Some providers, such as Google+, Tumblr and Flickr, manipulate only meta-data embedded in image files. Meanwhile, Facebook necessarily recompress images regardless of the data size and the resolution. Otherwise, Twitter recompresses images only when the data size of images uploaded by users is larger than a threshold.

Most of SNS providers support the JPEG standard [20], which is one of the most widely used image compression standards. Therefore, we focus on the JPEG standard in this paper. The JPEG encoding of color images consists of the following six steps:

- 1) Perform the color transformation from RGB space to YCbCr space.
- 2) Sub-sample the Cb and Cr components to reduce the spatial resolution.
- 3) Divide the image into 8-by-8 blocks.
- 4) Apply 2-D DCT to each block.
- 5) Carry out block-based quantizing with a quantization matrix Q .
- 6) Carry out Entropy coding using Huffman coding.

SNS providers reduce the data size of uploaded images by changing Quality factor $Q_f (1 \leq Q_f \leq 100)$, which is a parameters to control the matrix Q in step 5). $Q_f=100$ gives the best quality and $Q_f=1$ provides the worst quality.

There are three sub-sampling ratios in the JPEG standard, referred to as [4:2:0] (reduction by a factor of 2 in both the horizontal and vertical directions), [4:2:2] (reduction by a factor of 2 in the horizontal direction), and [4:4:4] (no sub-sampling) respectively. The sub-sampling conditions are also important to consider the effects of image manipulation on social media. By performing steps 3) to 6) for the brightness component Y and the sub-sampled chroma components Cb and Cr independently, the JPEG bitstream of a color image is generated.

B. Recompression parameters

Table I shows the relationship between uploaded images and downloaded ones in terms of sub-sampling rates and quality factors. This relationship in Table I, which is firstly investigated by this work, have been confirmed by uploading and downloading a lot of JPEG images to each SNS provider through a personal computer. For instance, if a user uploads JPEG images with 4:4:4 sampling to Facebook, an audience will receive manipulated JPEG files with 4:2:0 sampling and specific quality factors ($71 \leq Q_f \leq 85$). Let us consider image manipulation on Facebook and Twitter in more detail.

a) Image manipulation on Facebook

Facebook recompresses uploaded JPEG files with the size of up to 2048×2048 (HQ) or up to 960×960 (LQ) as below.

- 1) Decompress JPEG files as color images in the spatial domain.

TABLE I: Relationship between uploaded JPEG files and downloaded ones in terms of sub-sampling ratios

SNS provider	Uploaded JPEG file		Downloaded JPEG file	
	Sub-sampling ratio	Q_f	Sub-sampling ratio	Q_f
Twitter (Up to 4096×4096 pixels)	4:4:4	low	No recompression	
		high	4:2:0	85
	4:2:0	1,2,...,84	No recompression	
		85,86,...,100	4:2:0	85
Facebook (HQ, Up to 2048×2048 pixels)	4:4:4	1,2,...,100	4:2:0	71,72,...,85
Facebook (LQ, Up to 960×960 pixels)	4:2:0			
Tumblr (Up to 1280×1280 pixels) Google+ Flicker	4:4:4		No recompression	
	4:2:0			

TABLE II: Condition of JPEG images to avoid block artifact generated by Facebook and Twitter (× :Block artifact is generated, ○ : No block artifact)

SNS provider	Twitter		Facebook	
Sub-sampling ratio (Uploaded JPEG file)	4:4:4	4:2:0	4:4:4	4:2:0
Block artifact	○	○	○	×

- 2) Compress the images under 4:2:0 sub-sampling ratio and specific $Q_f(71 \leq Q_f \leq 85)$ in accordance with the Facebook compression algorithm.
- 3) Save the recompressed JPEG files to a server to publish them for audiences.

As shown above, all uploaded images by users are converted to JPEG files with 4:2:0 sampling regardless of the data size of images.

b) Image manipulation on Twitter

Twitter recompresses uploaded JPEG files, according to the sub-sampling ratios. When a user uploads JPEG files compressed under high quality and 4:4:4 sampling to Twitter, images are recompressed as below.

- 1) Decompress JPEG files as color images in the spatial domain.
- 2) Compress the images under 4:2:0 sub-sampling ratio and $Q_f=85$.
- 3) Save the recompressed JPEG files to a server to publish them for audiences.

Meanwhile, Twitter recompresses uploaded JPEG files, if they were compressed under high quality ($Q_f \geq 85$) and 4:2:0 sampling as below.

- 1) Reconstruct DCT coefficients by entropy decoding.
- 2) Quantize the DCT coefficients by the quantization matrix Q with $Q_f=85$.
- 3) Carry out entropy coding using Huffman coding.
- 4) Save the recompressed JPEG files to a server to publish them for audiences.

Note that Twitter manipulates only meta-data including in the header of JPEG images if uploaded JPEG ones were compressed under low quality such as $Q_f = 60$.

C. Requirements to avoid distortion

Some block distortion is often generated in decrypted images, depending on the relationship between encryption and recompression conditions. Requirements to avoid the distortion are discussed here. It is concluded that any block distortion are not generated due to image manipulation on social media if encrypted images satisfy both of below the two conditions.

- a) The resolution of encrypted images is unchanged on SNS providers.
- b) The encrypted images uploaded by users are compressed with 4:4:4 sub-sampling ratio.

First, requirement a) means that the resolution of encrypted images need to be smaller than the maximum resolutions that each provider decides as a resizing condition. As shown in Fig. 5, users need not consider the maximum resolution of encrypted images when uploading to Google+ and Flickr.

Meanwhile, requirement b) means that we have to consider the sub-sampling ratios of encrypted images. JPEG images with 4:2:0 sub-sampling ratio are performed to increase the spatial resolution for chroma components in the decoding process. This interpolation processing is carried out by using the relationship among blocks. Therefore, the encrypted images with 4:2:0 sub-sampling ratio are affected by this interpolation, while JPEG images with 4:4:4 sampling do not need any interpolation.

However, JPEG files with 4:2:0 sub-sampling ratio can sometimes avoid to generate block distortion even if the interpolation is carried out. Table II indicates the conditions of JPEG files uploaded by users to avoid block artifact. As discussed in Sec. III-B, Facebook performs the interpolation in the spatial domain when JPEG images with 4:2:0 sampling ratio are uploaded by users. Consequently, images with artifact are generated such as Fig. 3(c). Meanwhile, Twitter manipulates JPEG images in the DCT domain for some operations such as quantization. Therefore, JPEG images with 4:2:0 sampling uploaded to Twitter can avoid to generate block distortion due to the recompression. Thus users do not need to consider the sampling ratios of encrypted images when uploading to Twitter.

IV. EXPERIMENTAL RESULTS

We evaluated the effectiveness of the EtC systems for social media by a number of simulations. In the experiment,

TABLE III: Experimental result (Facebook)

Dataset	Sampling ratio	$Q_f = 80$		$Q_f = 85$		$Q_f = 90$		$Q_f = 95$	
		No-encryption PSNR (dB)	Encryption PSNR (dB)	No-encryption PSNR (dB)	Encryption PSNR (dB)	No-encryption PSNR (dB)	Encryption PSNR (dB)	No-encryption PSNR (dB)	Encryption PSNR (dB)
(a)MIT	4:2:0	32.000	31.392*	30.885	30.432*	32.974	32.074*	33.253	32.056*
	4:4:4	32.162	32.137	31.028	31.023	33.065	32.866	33.321	32.830
(b)FHD	4:2:0	33.813	32.779*	32.694	31.815*	35.566	33.468*	35.940	33.442*
	4:4:4	33.978	33.943	32.868	32.783	35.660	34.730	36.027	34.685

* Distorted by image manipulation

TABLE IV: Experimental result (Twitter)

Dataset	Sampling ratio	$Q_f = 80$		$Q_f = 85$		$Q_f = 90$		$Q_f = 95$	
		No-encryption PSNR(dB)	Encryption PSNR(dB)	No-encryption PSNR(dB)	Encryption PSNR(dB)	No-encryption PSNR(dB)	Encryption PSNR(dB)	No-encryption PSNR(dB)	Encryption PSNR(dB)
(a)MIT	4:2:0	34.139	34.114	35.119	35.097	33.942	33.947	35.071	35.048
	4:4:4	34.361	34.338	35.031	34.958	34.264	34.226	35.045	34.968
(b)FHD	4:2:0	36.076	36.059	37.126	37.099	35.910	35.887	37.054	37.027
	4:4:4	36.233	36.217	37.047	37.019	36.157	36.112	37.051	37.023

encrypted and compressed JPEG files were uploaded to SNS providers.

A. Simulation conditions

The following procedure is carried out to evaluate the robustness of EtC systems based on Fig. 4.

- 1) Generate an encrypted image I_e from an original image I in accordance with Fig. 1.
- 2) Compress the encrypted image I_e .
- 3) Upload the encrypted JPEG image I_{ec} to SNS providers.
- 4) Download the recompressed JPEG image \hat{I}_{ec} from the providers.
- 5) Decompress the encrypted JPEG file \hat{I}_{ec} .
- 6) Decrypt the manipulated image \hat{I}_e .
- 7) Compute the PSNR between the original image I and \hat{I} .

We made use of the JPEG implementation from the Independent JPEG Group (IJG) [26] in steps 2) and 5). Then, we compressed each image under 4:2:0 or 4:4:4 sampling ratio and $Q_f = 80, 85, 90, 95$. To calculate PSNR values in step 7), the original image I was compressed without any encryption, and then uploaded. In order to reduce dispersion, we employed 2 datasets as below.

- (a) 20 images from resized MIT dataset (672×480) [27].
- (b) 20 FHD images from Ultra-Eye dataset (1920×1080) [28].

We focused on Facebook and Twitter, because the SNS providers always recompress, as illustrated in Fig. 5, if images uploaded by users meet the conditions. The encrypted and non-encrypted JPEG files with 4:2:0 and 4:4:4 sampling from dataset (a) and (b) were uploaded to the providers.

B. Compression performance of EtC system

Table III and IV show the experimental results, where the average PSNR values of 20 images were calculated. As shown in Table III, the PSNR values of decrypted images were low when images with 4:2:0 sub-sampling ratio were uploaded to Facebook. It is confirmed that some block distortion due to

recompression in the spatial domain heavily reduce the quality of decrypted images.

On the other hand, Twitter recompresses images with 4:2:0 sub-sampling ratio in the DCT domain. Thus, the PSNR values of decrypted images uploaded to Twitter were almost same as non-encryption ones even if images were compressed with 4:2:0 sub-sampling ratio. In terms of images with 4:4:4 sub-sampling ratio, the PSNR values of ones were almost same as 4:2:0 sampling when encrypted images were uploaded to Twitter. However, the PSNR values of decrypted images with 4:4:4 sampling were also low due to the Facebook compression algorithm.

V. CONCLUSION

This paper has proposed the application of EtC systems, which enable users to send images securely to audience through SNS providers. Moreover, we also investigate how SNS providers manipulate JPEG images uploaded by users in terms of the maximum resolution of uploaded images and the parameters of recompression. The simulation results showed that some block distortion due to recompression by SNS providers heavily reduce image quality. On the other hand, it is confirmed that the EtC systems are applicable to almost all SNS providers if encrypted images meet some conditions.

ACKNOWLEDGMENTS

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, pp. e7, 2014.
- [2] R. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Magazine*, 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 transactions on information forensics and security*, vol. 9, no. 1, pp. 39-50, 2014.

- [4] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 118–129, 2003.
- [5] I. Ito and H. Kiya, "A new class of image registration for guaranteeing secure data management," in *IEEE International Conference on Image Processing (ICIP)*, 2008, pp. 269–272.
- [6] H. Kiya and I. Ito, "Image matching between scrambled images for secure data management," in *16th European Signal Processing Conference (EUSIPCO)*, 2008, pp. 1–5.
- [7] I. Ito and H. Kiya, "One-time key based phase scrambling for phase-only correlation between visually protected images," *EURASIP Journal on Information Security*, vol. 2009, no. 841045, pp. 1–11, 2010.
- [8] Z. Tang, X. Zhang, and W. Lan, "Efficient image encryption with block shuffling and chaotic map," *Multimedia Tools Applications*, vol. 74, no. 15, pp. 5429–5448, 2015.
- [9] Z. Erkin, A. Piva, S. Katzenbeisser, R. L. Lagendijk, J. Shokrollahi, G. Neven, and M. Barni, "Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing," *EURASIP Journal on Information Security*, vol. 2007, no. 78943, pp. 1–20, 2007.
- [10] N. Nimbokar G. and S. Sarode V., "Article: A survey based on designing an efficient image Encryption-then-Compression system," *IJCA Proceedings on National Level Technical Conference X-PLORE 2014*, vol. XPLORE2014, pp. 6–8, 2014.
- [11] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 2992–3006, 2004.
- [12] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted grayscale images," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 1097–1102, 2010.
- [13] X. Zhang, "Lossy compression and iterative reconstruction for encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 53–58, 2011.
- [14] R. Hu, X. Li, and B. Yang, "A new lossy compression scheme for encrypted gray-scale images," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7387–7390.
- [15] 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)*, 2015, pp. 1226–1230.
- [16] K. Kurihara, S. Shiota, and H. Kiya, "An encryption-then-compression system for jpeg standard," in *Picture Coding Symposium (PCS)*, 2015, pp. 119–123.
- [17] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for jpeg/motion jpeg standard," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 98, no. 11, pp. 2238–2245, 2015.
- [18] 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)*, 2016, pp. 1–5.
- [19] K. Kurihara, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for lossless image compression standards," *IEICE transactions on information and systems*, vol. E100-D, no. 1, pp. 52–56, 2017.
- [20] G.K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, pp. 30–44, 1991.
- [21] R. Caldelli, R. Becarelli, and I. Amerini, "Image origin classification based on social network provenance," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1299–1308, 2017.
- [22] O. Giudice, A. Paratore, M. Moltisanti, and S. Battiato, "A classification engine for image ballistics of social data," *arXiv preprint arXiv:1610.06347*, 2016.
- [23] M. Moltisanti, A. Paratore, S. Battiato, and L. Saravo, "Image manipulation on facebook for forensics evidence," in *International Conference on Image Analysis and Processing (ICIAP) 2015*, 2015, pp. 506–517.
- [24] 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)*, 2017, pp. 2157–2161.
- [25] 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)*, 2017, pp. 229–234.
- [26] "Independent jpeg group," <http://www.ijg.org/>.
- [27] T. Cho, S. Avidan, and W. Freeman, "A probabilistic image jigsaw puzzle solver," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 183–190.
- [28] H. Nemoto, P. Hanhart, P. Korshunov, and T. Ebrahimi, "Ultra-eye: Uhd and hd images eye tracking dataset," in *Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*, 2014, pp. 39–40.