Encryption and Data Insertion Technique using Region Division and Histogram Manipulation

Ryoma Ito*, KokSheik Wong†, Simying Ong‡, and Kiyoshi Tanaka§
* Graduate School of Science and Technology, Shinshu University, Japan.
† School of Information Technology, Monash University Malaysia, Malaysia.
‡ Faculty of Computer Science and Information Technology, University of Malaya, Malaysia.
§ Academic Assembly (Institute of Engineering), Shinshu University, Japan.

***Abstract*—A separable encryption and data insertion method is proposed in this paper. The input image is divided into 2 parts, where the first part is manipulated to mask the perceptual semantics, while the second part is processed to hide data. The binary image, which is the data to be inserted, further divides the second part of the input image into 2 regions called the 'zero' and 'one' regions. Pixels of the original image at position coinciding with the 'zero' region are darken, while those coinciding with the 'one' region are brightened. The darkening and brightening processes are performed by using histogram matching technique. The proposed joint method is separable, where the inserted binary image can be extracted directly from the masked image or from the reconstructed image. The proposed method is also commutative because the same results is achieved irregardless of the order of processing in encrypting and inserting data. Experiments were carried out to verify the basic performances of the proposed method.**

I. Introduction

Joint encryption and data insertion (JEDI) has received much attention in recent years thanks to the features it offers in addressing today's applications. For example, image is encrypted before transmission or online storage to avoid unauthorized viewing, while data is inserted to facilitate the claim of ownership, fingerprinting, authentication, management of content, to name a few [1], [2], [3].

Over the years, there are many innovations in achieving JEDI. In addition to the sequential approach of encryption-then-insertion and insertion-then-encryption, one of the most common ways to achieve JEDI is to split the content into 2 non-overlapping parts, where each part is manipulated to achieve different objectives. For example in Zhang's method [4], 5 most significant bit (MSB) planes of an image is manipulated to encrypt the image, while the remaining 3 bitplanes are divide into 2 groups where one of the groups is flipped to encode zeros and ones. An approximation of the original image can be obtained by considering the correlation among LSB bitplanes. Another class of approaches is to insert data to purposely mask the image. One such was is proposed by Ong et al. [1], where each pixel in some selected range is associated to another pixel value out side of the range, where the difference between the 2 values are huge. The original value is assumed when '0' is to be inserted, while the associated value is output when '1' is to be inserted. Departing from the spatial domain, researchers also transformed the input image into another domain. For example, Cancellaro et al. [5] transform the input image using Tree-Structure Haar transform, where the coefficients are decomposed into bitplanes for further processing. Specifically, the MSB bitplanes fo the coefficients are encrypted using AES, while the LSB bitplanes are manipulated to encode data. JEDI is also realized in other domains such compressed image and video [6], [7], [8], where syntax elements of the compression standards are judiciously manipulated for format compliance.

While the conventional JEDI methods are able to achieve its objectives, the usual approach taken to insert data is based on some association of pixel values. In addition, the number of bits that can be inserted (i.e., payload) in the conventional methods are relatively low. Therefore, in this work, a JEDI method based on histogram division and matching is proposed. Specifically, a few MSB bitplanes are manipulated to mask the image, while the remaining LSB bitplanes are modified to insert data. Notably, based on the binary image to be inserted, the pixels in the host image darken when they coincide with the zeros in the image, or brighten otherwise. The inserted data can be extracted from the encrypted or decrypted image, i.e., separable. For the rest of the presentation, for convenience, we utilize the term encryption to refer to perceptual masking.

II. Proposed JEDI method

Fig. 1 illustrates the architectural overview of the proposed JEDI method. The detailed information about each process is described in the following subsections. Without loss of generality, assume that the image A of dimension $M \times N$ has a bit-depth of 8. Let $A(x, y)$ refer to the pixel value at position (x, y) for $x \in [1, M]$ and $y \in [1, N]$.

A. Encryption and data insertion

Denote the most and least significant bits by b_7 and b_0, respectively. Let τ be an integer such that $1 \leq \tau \leq 7$ which divides $A(x, y)$ into two parts, i.e., $P_0(x, y) = a_7^{xy} a_6^{xy} \cdots a_\tau^{xy}$ and $P_1(x, y) = a_{\tau-1}^{xy} \cdots a_0^{xy}$. $P_0(x, y)$ will be processed to mask the perceptual semantic of A, while $P_1(x, y)$ will be manipulated to encode the external data. Specifically, $P_0(x, y)$ is processed as follows:

$$P_0'(x, y) = \mathrm{mod}(P_0(x, y) + r, 2^{8-\tau}), \tag{1}$$

where r is pseudorandomly generated integer in $[0, 2^{8-\tau} - 1]$ using key κ_1. This process changes the original pixel value significantly.

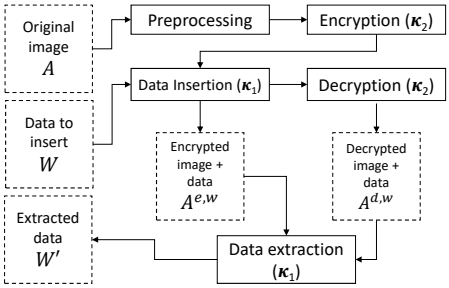


Fig. 1: The process flow of hiding data into an image.

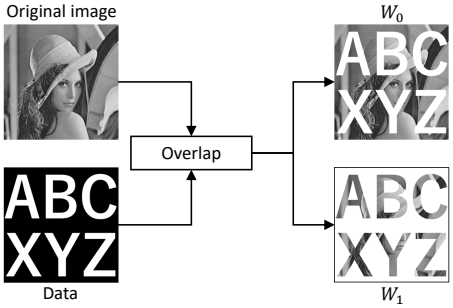


Fig. 2: Overlapping the watermark image.

Next, the set $\{P_1(x,y)\}$ are shuffled, i.e., the locations are changed, to obtain $P_1^*(x,y) = P_1(x',y')$. The positions (x',y') are determined by using key κ_2. The histogram $H_{P_1} = \{h[i]\}$ of P_1^* is constructed, where $h[i]$ denotes the number of occurrences for pixel value i. Note that the range of values for P_1 is $[0, 2^\tau - 1]$. This histogram is then divided into 2 histograms (i.e., divided exactly in the middle), namely H_0 and H_1 based on the threshold value $2^{\tau-1}$. Here, H_0 is exactly the same as H_{P_0}, except that $h[i] = 0$ for $i \geq 2^{\tau-1}$. Similarly, H_1 is exactly the same as H_{P_1}, except that $h[i] = 0$ for $i < 2^{\tau-1}$.

Suppose that the data to be embedded is a binary image W such that the value at position (x,y), denoted by $W(x,y)$, is either '0' or '1'. The shuffled set $\{P_1^*(x,y)\}$ is further divided into two parts based on W as follows:

$$\begin{aligned} W_0(x,y) &\leftarrow (1 - W(x,y)) \times P_1^*(x,y), \\ W_1(x,y) &\leftarrow W(x,y) \times P_1^*(x,y). \end{aligned} \tag{2}$$

An example is shown in Fig. 2. The histogram of W_0 and W_1 are constructed and denoted by H_{w0} and H_{w1}, respectively.

To encode data, exact histogram specification technique by Coltuc et al. [9] is deployed for changing the shape of the histogram H_0 into the shape of the histogram H_{w0}. Similarly, the shape H_1 is transformed to that of H_{w1}. With this construct, pixels in W_0 will be transformed into a smaller value (i.e., darken). Likewise, pixels in W_1 will be brighten. Figure 3 shows an example of how the values are modified using [9]. Basically, total ordering is achieved for each pixel by considering more criterion, and the value each pixel is

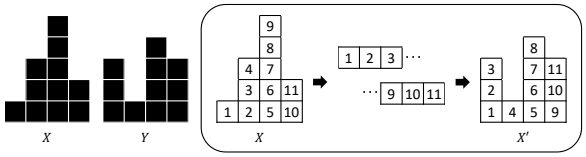


Fig. 3: Example of histogram shape conversion.

modified in the sequence induced by the ordering. Note that the number of pixels in H_0 and H_{w0} must be the same, i.e.,

$$\sum_{i=0}^{(2^{\tau-1})-1} h_0[i] = \sum_{i=0}^{2^\tau - 1} h_{w0}[i]. \tag{3}$$

When there is a mismatch, the histogram H_0 is scaled by the factor k, i.e., $H_0'[i] \leftarrow H_0[i] \times k$, where k is computed as follows:

$$k = \left(\sum_{i=0}^{2^\tau - 1} H_{w0}[i] \right) \div \left(\sum_{i=0}^{(2^{\tau-1})-1} H_0[i] \right). \tag{4}$$

Due to rounding error, the remaining numbers, if any left, will be taken care by the lower bins in the histogram. The modified values, after applying exact histogram specification, become $\{P_1'(x,y)\}$. The final image $A^{e,w}$, which is encrypted and containing data, is formed by concatenation, i.e.,

$$A^{e,w}(x,y) = P_0'(x,y) | P_1'(x,y). \tag{5}$$

In fact, based on the equation above, the proposed method is also *commutative* since P_0' and P_1' are completely independent. Specifically, *commutative* refers to the property where the exact same output can be obtained irregardless if the image is encrypted first followed by data insertion, or data is inserted first followed by encryption.

B. Decryption and data extraction

The proposed method is separable in which the decryption and extraction processes can take place independently. In other words, decryption can take place without remapping the pixel values to the corresponding (approximate) original values (i.e., reversing the data insertion process), and the data extraction process can take place without needing to decrypt the image first. Specifically, an approximation of the original image can be obtained by performing Eq. (6) on P_0' as follows:

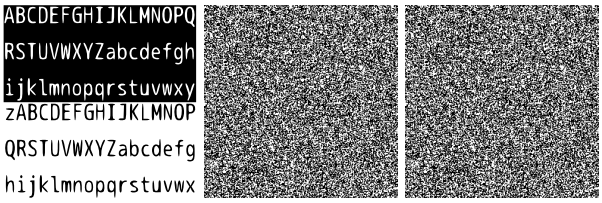
$$P_0''(x,y) \leftarrow \mathrm{mod}(P_0'(x,y) - r, 2^{8-\tau}). \tag{6}$$

Note that only κ_1 and τ are required here. The image $A^{d,w}$ is formed by combining the P_0'' and P_1'.

On the other hand, the inserted data can be extracted from P_1' of the encrypted image $A^{e,w}$ as well as the decrypted image $A^{d,w}$. The parameter value τ is required to determined whether the value of $P_1'(x,y)$ is encoding '0' or '1' (i.e., compared to the threshold value $2^{\tau-1} - 1$), and κ_2 is needed to reserve the shuffling process. Specifically, when a value $P_1'(x,y)$ is smaller than $2^{\tau-1}$ then the extracted data at position (x,y) is $W'(x,y) = 0$. Otherwise $W'(x,y) = 1$. The reshuffled values then formed the extracted data, i.e., $W'(x,y) \leftarrow W'(x',y')$.

(a) Lenna (b) Woman (c) Boat (d) Airplane (e) Cameraman (f) Mandrill (g) Bridge (h) Peppers

Fig. 4: The original test images.



(a) Original (b) $\kappa_2 = 138$ (c) $\kappa_2 = 846$

Fig. 5: The binary image extracted from different images.

III. Experiments

The proposed joint encryption and data insertion method is implemented using Matlab version R2017b (9.3.0.713579). Eight standard test images, which are shown in Fig. 4, are considered to verify the performance of the proposed JEDI method. The binary image shown in Fig. 5(a) is used as the data to be embedded into the host image throughout the evaluation process. For all experiments conducted, when using the correct key κ_2, it is verified that the extracted data is exactly the same as the original one. This happens irregardless if the data is extracted from $A^{e,w}$ and $A^{d,w}$. Note that the payload is $M \times N \times C$ bits, where $M \times N$ is the image dimension, and C is the number of color channels.

First, the encrypted images with data inserted (using $\tau = 6$) are shown in Fig. 6. These images suggest that the proposed JEDI method is able to completely mask the perceptual semantic of the corresponding original image. That is, there is no trace of the original image. The corresponding PSNR and SSIM [10] values achieved by other τ values are summarized in Table I. Since the PSNR for the test image 'Woman' is > 10dB, the condition $\tau \geq 2$ should be enforced.

Second, each processed image is decrypted to regenerate an approximation of the original image $A^{d,w}$. Note that even after decryption, a perfect reconstruction is not possible because part of the image, viz., P_1', is reserved for data insertion. Using the Lenna image as a representative example, Fig. 7 shows the decrypted images $A^{d,w}$ when the different τ values are considered. The PSNR and SSIM between A and $A^{d,w}$ are recorded in Table II. It is observed that when τ is small, the PSNR value is high, and vice versa. Results also suggest that the condition $\tau < 5$ should be considered maintain high quality decrypted image.

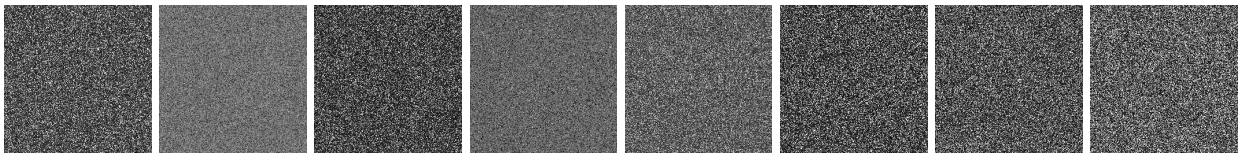
Next, to verify the robustness of the inserted data with respect to unauthorized extraction, the data is extracted using different key κ_2'. For this particular evaluation, $\kappa_2 = 336$ is used for data insertion. The data extraction process is carried out using different keys. The mean square error (MSE) between the original data W and the extracted one W' is computed for each key, and the results are plotted in Fig. 8. Fig. 5 also shows the extracted watermark with correct seed "336" and incorrect seed "138" and "846". It is obvious that only the correct key, i.e., $\kappa_2' = 336$, is able to extract the inserted data correctly. Similarly, when decryption is attempted by using the wrong key κ_1', it is found that output is completely gibberish. Due to space limitation, the results and discussions are omitted.

When compared to the conventional JEDI methods, the proposed method offers a more rounded performance. For example, Zhang's method [4], although it is able to control the insertion rate by changing the block size, bit error rate is relatively high when small block size is considered (i.e., when more data can be accommodated). The error rate reduces to less than 1% when the block size is ≥ 16, which translates to 3.90625×10^{-3} bits per pixel (bpp). Hong et al. [11] further suppresses the error rate in Zhang's method, but the payload is still small due to the block size considered, i.e., 8×8, which translates to 1.5625×10^{-2} bpp. Although Ong et al.'s method [1] is able to insert 2.88 bpp, their method is neither commutative nor separable. Therefore, the proposed JEDI method offers more features with balanced performance.

IV. Conclusions

In this work, a joint encryption and data insertion method is put forward. In particular, the input image is divided into two, where one part is processed to mask the perceptual semantic of the image, and another part is manipulated to insert data. Unlike the conventional methods, pixel values are modified significantly, viz., brightened or darkened, depending on the data to be inserted. Experiments results suggest that, when compared with the conventional JEDI methods, the proposed method offers well-balances performance, as well as the commutative and separable properties.

As future work, the possibilities in inserting multiple images into a single image will be explored. The quality of the decrypted image will also be further improved by means of post-processing, since an approximation of the histogram of the original image is available. Last but not least, the encryption process will be revisited to improve security and secrecy.



(a) Lenna (b) Woman (c) Boat (d) Airplane (e) Cameraman (f) Mandrill (g) Bridge (h) Peppers

Fig. 6: Encrypted images using $\tau = 6$.

TABLE I: PSNR (dB) and SSIM for encrypted image containing data.

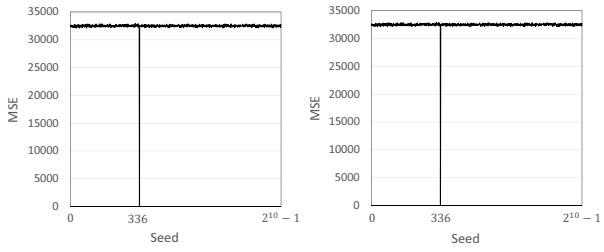
τ	Lenna		Woman		Boat		Airplane		Cameraman		Mandrill		Bridge		Peppers	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	9.05	0.011	9.18	0.032	9.24	0.009	9.05	0.023	9.40	0.013	8.7556	0.0067488	8.12	0.010	7.98	0.008
2	9.11	0.012	9.32	0.032	9.32	0.009	9.15	0.024	9.42	0.012	8.8691	0.0068541	8.26	0.010	8.11	0.008
3	9.23	0.012	9.60	0.032	9.47	0.009	9.37	0.024	9.46	0.013	9.1104	0.0070479	8.41	0.011	8.33	0.008
4	9.22	0.012	9.60	0.033	9.47	0.009	9.35	0.024	9.46	0.014	9.1234	0.0071203	8.43	0.011	8.34	0.008
5	9.24	0.013	9.51	0.035	9.46	0.009	9.34	0.026	9.39	0.013	9.1952	0.0068938	8.45	0.011	8.42	0.009
6	9.26	0.009	9.75	0.049	9.01	0.009	9.24	0.011	9.15	0.012	9.3763	0.0059287	8.69	0.014	8.66	0.012
7	9.06	0.007	14.24	0.104	9.97	0.017	10.77	0.024	8.49	0.008	10.071	0.0058592	9.26	0.022	9.27	0.017

TABLE II: PSNR (db) and SSIM for decrypted images with inserted data intact.

τ	Lenna		Woman		Boat		Airplane		Cameraman		Mandrill		Bridge		Peppers	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	51.12	0.997	51.13	0.997	51.13	0.998	51.14	0.997	51.14	0.996	51.16	0.999	51.11	0.999	51.16	0.997
2	46.35	0.990	46.28	0.992	46.36	0.994	46.35	0.990	46.44	0.988	46.31	0.996	49.19	0.999	46.39	0.991
3	40.74	0.966	40.78	0.973	40.68	0.977	40.82	0.965	40.77	0.959	40.71	0.987	40.01	0.990	40.77	0.968
4	34.58	0.886	35.04	0.911	34.71	0.922	34.76	0.885	34.68	0.866	34.93	0.954	34.82	0.968	34.79	0.894
5	29.00	0.721	29.15	0.755	29.44	0.804	29.51	0.738	29.65	0.712	28.69	0.853	28.66	0.891	29.54	0.748
6	22.92	0.476	23.27	0.499	22.43	0.542	21.99	0.482	23.28	0.454	23.76	0.688	22.83	0.711	22.03	0.441
7	16.76	0.246	16.63	0.223	14.05	0.229	19.13	0.364	14.22	0.213	16.52	0.361	17.38	0.452	18.01	0.285

(a) $\tau = 1$ (b) $\tau = 2$ (c) $\tau = 3$ (d) $\tau = 7$

Fig. 7: The decrypted images $A^{d,w}$ for different values of τ



(a) From $A^{e,w}$ (b) From $A^{d,w}$

Fig. 8: Graphs of MSE vs κ_2 (seed value). Note that a smaller MSE implies better performance.

ACKNOWLEDGEMENT

This research is supported by the E-Science fund under the project - *Innovative High Dynamic Range Imaging - from Information Hiding to Its Applications* (Grant No. 01-02-10-SF0327).

REFERENCES

[1] SimYing Ong, KokSheik Wong, and Kiyoshi Tanaka, "A scalable reversible data embedding method with progressive quality degradation functionality," *Signal Processing: Image Communication,* vol. 29, no. 1, pp. 135149, 2014.

[2] SimYing Ong, KokSheik Wong, and Kiyoshi Tanaka, "Scramblingembedding for jpeg compressed image," *Signal Processing,* vol. 109, pp. 3853, 2015.

[3] Mustafa S. Abdul Karim and KokSheik Wong, "Data embedding in random domain," *Signal Processing,* vol. 108, pp. 5668, 2015.

[4] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters,* vol. 18, no. 4, pp. 255258, April 2011.

[5] M. Carli G. Boato F. G. B. De Natale A. Neri M. Cancellaro, F. Battisti, "A joint digital watermarking and encryption method," 2008.

[6] Jen-Chun Chang, Yi-Zhi Lu, and Hsin-Lung Wu, "A separable reversible data hiding scheme for encrypted jpeg bitstreams," *Signal Process.,* vol. 133, no. C, pp. 135143, Apr. 2017.

[7] Dawen Xu, RangdingWang, and Yun Q. Shi, "Data hiding in encrypted h.264/avc video streams by codeword substitution," *Trans. Info. For. Sec.,* vol. 9, no. 4, pp. 596606, Apr. 2014.

[8] Yiqi Tew, KokSheik Wong, Raphael C.-W. Phan, and King Ngi Ngan, "Separable authentication in encrypted hevc video," *Multimedia Tools and Applications,* pp. 2416524184, Sep. 2018.

[9] D. Coltuc, P. Bolon, and J. M. Chassery, "Exact histogram specification," *IEEE Transactions on Image Processing,* vol. 15, no. 5, pp. 11431152, May 2006.

[10] ZhouWang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Trans. Img. Proc.,* vol. 13, no. 4, pp. 600612, Apr. 2004.

[11] W. Hong, T. S. Chen, and H. Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters,* vol. 19, no. 4, pp. 199202, Apr. 2012.