Distributed Encrypted Image-Based Reversible Data Hiding

Yu-Chi Chen[1], Chih-Wei Shiu[2]

[1] Department of Computer Science and Engineering, Yuan Ze University, Taiwan
[2] Department of Education Industry and Digital Media, National Taitung University, Taiwan
wycchen@saturn.yzu.edu.tw, chihwei.shiu@gmail.com

Abstract

Sensor cameras over wireless sensor networks are common devices to realize surveillance or data gathering. In general, the sensed images will be routed from sensors to gateways which can deliver the collection of images to the server. For routing, the nodes which are located between the source and terminate need to add some information (e.g., tags) into the images. Assuming that the communications among sensors and gateways are via insecure channels, encryption is a simple way to keeping confidentiality. However, except for using homomorphic encryption, it is not achieved to perform computation over ciphertext, so nodes cannot easily add message into the encrypted images. In this paper, we focus on a cloud-edge model, and propose a new distributed encrypted image-based reversible data hiding scheme, where given an image the edge produces encrypted ones and distributes them to intermediate nodes which then generate the message-embedded encrypted image, and finally the cloud can recover the original image and extract the message. Our techniques mainly utilize exclusive-OR secret sharing as encryption, and apply difference expansion and simple preprocessing for the embedding strategy. The experimental results are provided to show the efficiency and effectiveness of this scheme.

Keywords: Data hiding, Reversible data hiding, Encrypted image, Secret sharing, Cloud-edge model

1 Introduction

Wireless sensor networks (WSNs) [1] with smart sensors have gained worldwide attention in the recent years. Sensors are with limited processing and computing resources, and their cost is much cheaper. Those sensor nodes can be used to sense, measure, and gather information from the environment, based on some local decision processes, and then can send the sensed data to the user. Concealed data aggregation in WSNs is a method to provide that sensed data are consolidated and aggregated on their way to the final destination. For example, the collectors arrange sensors around the forest, the sensors can take pictures and send the sensed data to the relay points, and the relay points transmit the data to the server. This WSNs are also considered to be a Cloud-Edge (CE) architecture, where the sensors and relay points are referred to as the edges, and the server is as the cloud.

Consider the sensed data as images by sensing cameras. Those images will be routed among edges and finally achieve the cloud side. Moreover, for preserving integrity over routing, the internal edges which are on a path between the source edge and cloud need to add some information (e.g., tags) into the images. Assuming that the communications among sensors and gateways are via insecure channels, encryption is a simple way to keeping confidentiality. However, except for using homomorphic encryption [2], it is not achieved to perform computation over ciphertext, so nodes cannot easily add message into the encrypted images. In this paper, we present a solution by introducing distributed reversible data hiding for encrypted image. It is efficient and suffices to capture the above cloud-edge scenario.

1.1 Related Work

Data hiding is an important issue in information security. Through data hiding, we can send secret message that cannot be detected by attackers [3-6]. Generally, the data hider embeds the message into a chosen image (referred to as a *cover-image*) to get the *stego-image*. No one can discover the image with the embedded message by human vision except the receiver who knows the extraction algorithm. There are two basic types of schemes in image-based data hiding: nonreversible data hiding and reversible data hiding (RDH). Only in RDH, the receiver can recover the cover-image by extracting the secret message. Therefore, if the cover image is significant, RDH occupies an important position.

Difference expansion (proposed by Tian [7]) and histogram shifting (proposed by Ni et al. [8]) are two major methods to handle reversible data hiding schemes. Lots of RDH schemes follow these two concepts to improve payload and image quality. Recently, encrypted image-based reversible data hiding (EIRDH) was first introduced by Zhang [9], which

[*] Corresponding Author: Yu-Chi Chen; E-mail: wycchen@saturn.yzu.edu.tw
DOI: 10.3966/160792642021012201010

maps to the following scenario. The image provider would like to keep privacy of the cover-image, but still requires the data hider to embed the secret message. Therefore, the data hider embeds the message into the encrypted image which is generated by the image provider from the cover-image. Finally, the receiver can recover the original cover-image and extract the secret message correctly.

The EIRDH scheme of Zhang [9] depends on XOR operations, divides an encrypted image into many blocks, and handles each block one by one for hiding. However, this scheme does not work in the case that the block size is small (the false positive rate becomes higher). In order to remedy the above weakness, the scheme Hong et al. [10] provides an adaptive way to choose the block size. Since that, EIRDH has grabbed research attention, and some schemes have been presented [11-13] to give good payload and image quality.

1.2 Distributed Reversible Data Hiding in Encrypted Image

Very recently, a new model formalized by Wu et al. [14] is similar to the pure model by Zhang [9] except for the setting of the service provider. This model is referred to as a *distributed model* which is instead of one single P, many specific parties jointly provide the service. Indeed, these parties are one control center (denoted by C) and n storage and processing centers (denoted by $S_1, \dots, S_n$). Note that the such storage centers are duplicated to work with the identical task. In the model, each S_i will perform data embedding. However, summarizing the security definition of the model, the semi-honest adversary[1] is assumed to able to only corrupt a *threshold* number (by c) of storage centers, C and the other uncorrupted centers are honest. In addition, Wu et al. [14] relied secret sharing [15] as image encryption, and also presented the first secret sharing-based RDHEI scheme.

1.3 Contributions

Our starting point is an observation for Wu et al.'s scheme [14] including three critical points which we would like to improve.

- Polynomial recovery (also a.k.a. secret recovery) requires Gaussian elimination. The performance will become significantly slower if the degree is higher.
- The operations over share generation and secret recovery fully rely on a modulo $p = 251$. The such prime induces that either their scheme performs shrinking for pixels (mapping [0, ..., 255] to [0, ..., 250]) or it gives up five kinds of pixel values (e.g., 0, 1, 2, 254, 255) for data hiding.
- Such the prime modulo p must be strictly larger than the degree of the underlying polynomial. This limits the number of storage centers without doubt.

In this paper, we propose a new distributed RDHEI based on exclusive-OR secret sharing. This scheme pre-processes the image through the difference expansion before being encrypted. The encryption is the simple XOR secret sharing to produce encrypted images, where in particular XOR secret sharing only induces very low computational cost and is suited for light-weight edge devices sensors. The proposed XOR-based method also can avoid the upper bound of the number of storage devices. Due to page limitation, we omit to show some implementation results for image quality and embedding capacity. However, those factors are almost identical to those of Wu et al. [14] since the data hiding techniques are the same difference expansion.

1.4 Organization

In Section 2, we introduce the building block, secret sharing, and then show our system model. We present our new scheme in Section 3. The discussions are provided in Sections 4. Finally, the conclusions of this paper are given in Section 5.

2 Secret Sharing

Secret sharing [15] is a method for protecting an image by dividing it into multiple encrypted shares. We use exclusive-OR secret sharing as the building block to construct the distributed EIRDH scheme.

2.1 Exclusive-OR Secret Sharing

In (n, n) secret sharing scheme, a secret message secret will be split into n shares (also called shadows) $(share_1, share_2, \dots, share_n)$, where collecting n shares together can recover the secret, but any less than n shares cannot get any information about the secret. Let us briefly describe the (n, n) XOR secret sharing scheme.

– **Share Generation**
Taking $secret \in \{0,1\}^l$ as input, this algorithm chooses uniform $share_i \in \{0,1\}^l$ or all i, $1 \le i \le n-1$. Set the last share as $share = (\oplus_{i=1}^{n-1} share_i) \oplus secret$ where $\oplus$ is the bit-wise exclusive OR operation.

– **Secret recovery**
This algorithm recovers secret by computing $secret = \oplus_{i=1}^{n-1} share_i$.

3 The Proposed Scheme

In this section, we firstly describe the system model

[1] An adversary A is semi-honest if can only eavesdrop but cannot tamper. In cryptography, we say if A can tamper, then A is malicious.

including notations, and then present the details of the proposed scheme. Also, we provide an example to demonstrate the scheme.

3.1 System Model

The notations of the system are listed in Table 1.

Table 1. Notations

Notation	Description
R	receiver
H	data hider
P	image provider
ImageEnc	encrypting cover-image
Embedding	embedding secret bits into encrypted image
ImageDec	recovering the stego-image
Extracting	recovering the cover-image
CI	cover-image
EI	encrypted image
$EIwS$	encrypted image with embedding secret message
SI	stego-image

The system syntax is composed of four steps as follows:

- ImageEnc: P takes a cover-image CI, encrypts CI to produce encrypted images EI_1, ..., EI_n. P sends EI_1 to H_i.
- Embedding: H_i embeds secret bits into EI_1, then gets the encrypted image with secret $EIwS_i$.
- ImageDec: R receives $EIwS_1$, ..., $EIwS_n$ from H_1, ..., H_n, and then recovers the stego-image SI.
- Extracting: Finally, R performs data extraction to extract secret bits from SI, and then recovers the cover-image CI.

Figure 1. shows the flowchart of the system.

3.2 Details of the Scheme

For clearly describing the scheme, we directly instantiate a pixel pair (x, y) along with the following steps. In fact, we can repeat to apply the procedure for the whole image.

- **ImageEnc:** P pre-processes (x, y) to compute $l = \left\lfloor \frac{x+y}{2} \right\rfloor$ and $d = x - y$, and then computes $x' = l + d$ and $y' = l + d$. Note that d can be positive or negative, but does not influence on the results. By using sharing generation, for a pair (x', y'), P generates $\{\{x'_1, y'_1\}, \{x'_2, y'_2\}, \{x'_n, y'_n\}\}$ where x'_j, y'_j for all j, $1 \le i \le n-1$, are chosen uniformly, and $x'_n = x' \oplus_{i=1}^{n-1} x'_i$ and $y'_n = y' \oplus_{i=1}^{n-1} y'_i$. P then sends (x'_1, y'_i) to H_i.
- **Embedding:** For embedding a bit 1, we consider two cases:

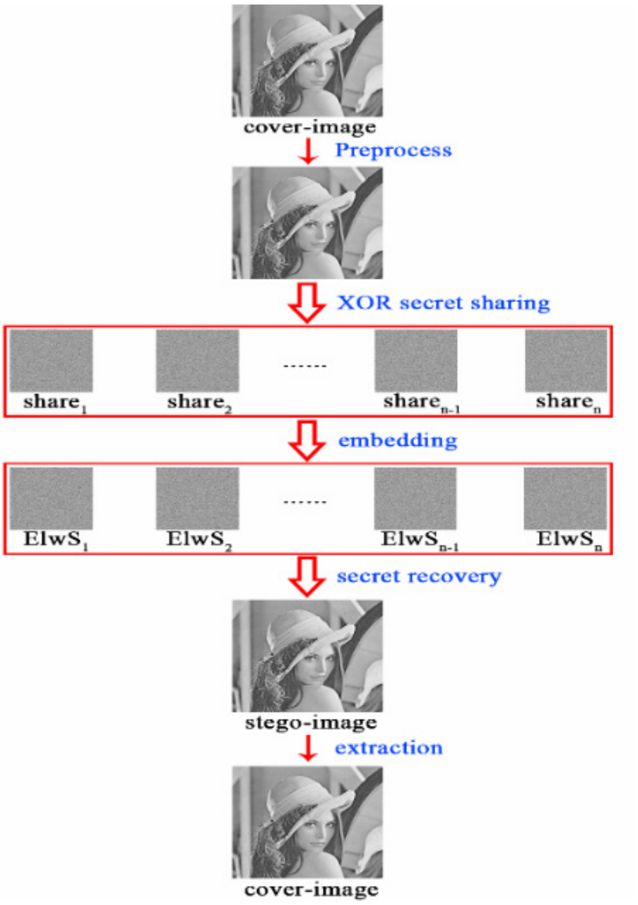


Figure 1. Steps of our new scheme

  – If n is odd, for all i, $1 \le i \le n$, H_i sets $x''_i = x'_i \oplus 0...01$ and $y''_i = y'_i$ to get the result $EIwS_i$

  – If n is even, for all i, $1 \le i \le n-1$, H_i sets $x''_i = x'_i \oplus 0...01$ and $y''_i = y'_i$. H_n does not perform any processing, and directly sets $x''_n = x'_n$ and $y''_n = y'_n$ to get the result $EIwS_i$.

  For embedding a bit 0, H_i does not perform any processing, and directly sets $EIwS_i(x''_i = x'_i$ and $y''_i = y'_i)$.

- ImageDec: R uses $\oplus_{i=1}^{n} EIwS_i$ to get secret. R recovers x'' and y'' by computing $x''_i = \oplus_{i=1}^{n} x''_i$ and $y'' = \oplus_{i=1}^{n} y''_i$.
- Extracting: If x'' and y'' are both odd or both even, R extracts $b = 0$ and recovers $x' = x''$ and $y' = y''$; if not, extracts $b = 1$ and recovers $x' = x'' \oplus b$ and $y' = y''$ since b is only embedded into the first pixel. R can easily compute $l = \left\lfloor \frac{x'+y'}{2} \right\rfloor$ and $d = \frac{x'-y'}{2}$ and obtain the original pixel pair of CI by computing $x = l \left\lfloor \frac{d+1}{2} \right\rfloor$ and $y = l - \left\lfloor \frac{d}{2} \right\rfloor$. At the end, this step will output SM and recovers CI.

3.3 Example

Assuming a pixel pair $(x, y) = (4, 2)$, we pre-process (x, y) to compute $l = \left\lfloor \frac{x+y}{2} \right\rfloor = 3$ and $d = x - y = 2$, and then compute $x' = l + d = 5$ and $y' = l - d = 1$. By using share generation, for a pair (x', y'), we generate $\{\{x_1', y_1'\}, \{x_2', y_2'\}, \{x_3', y_3'\}\}$ where x_1', y_1', x_2', y_2' are chosen uniformly from 0 to 255. Suppose $(x_1', y_1') = (4, 6), (x_2', y_1') = (254, 72)$. $x_3' = x' \oplus_{i=1}^{2} x_i' = 255$ and $y_3' = y' \oplus_{i=1}^{2} y' = 79$. For embedding a bit 1, we get $ELwS_1 = (x_1'', y_1'')$, $ELwS_2 = (x_2'', y_2'')$ and $ELwS_3 = (x_3'', y_3'')$ where $x_1'' = x_1' \oplus 0...01 = 5$, $x_2'' = x_2' \oplus 0...01 = 255$, $x_3'' = x_3' \oplus 0...01 = 254$, $y_1'' = y_1' = 6$, $y_2'' = y_2' = 72$ and $y_3'' = y_3' = 79.$. In ImageDec, we obtain the pair (x'', y'') by computing $x'' \oplus_{i=1}^{3} x_i'' = 4$ and $y'' \oplus_{i=1}^{3} y_i'' = 1$. Finally, we recover CI and get SM by extracting $b = 1 \cdot x' = x'' \oplus b = 5$ and $y' = y'' = 1.$. We can compute $l = \left\lfloor \frac{x'+y'}{2} \right\rfloor = 3$ and $d = \frac{x'-y'}{2} = 2$ and obtain the original pixel pair of CI by computing $x = l + \left\lfloor \frac{d+1}{2} \right\rfloor = 4$ and $y = l - \left\lfloor \frac{d}{2} \right\rfloor = 2$.

Our scheme can support partial decryption (see Figure 2). Here we directly present the partial decryption which is run by a node.

- **Partial-ImageDec:** It can partially recover the partially decrypted image from a pair of $EIwS_i$ and $EIwS_j$ or more to get a partially decrypted image by doing XOR them.

Here we remark that the proposed scheme can be modularly converted into a compilation that given a specific reversible data hiding in homomorphic encrypted domain, generates a distributed one. This technique is standard by [16] as long as the underlying scheme in homomorphic encrypted domain satisfies some properties.

4 Discussions

Now we implement two methods of Gaussian elimination decryption and XOR decryption respectively, and roughly compare their results. In Figure 3, we can find that the time of XOR encryption is linearly faster than the time of Gaussian elimination encryption about 4 times, the time of XOR decryption is faster than the time of Gaussian elimination decryption about 7 times. For the total time, XOR is faster than Gaussian elimination about 5-ish times. According to the experimental results, we can observe that our method is more efficient than [14].

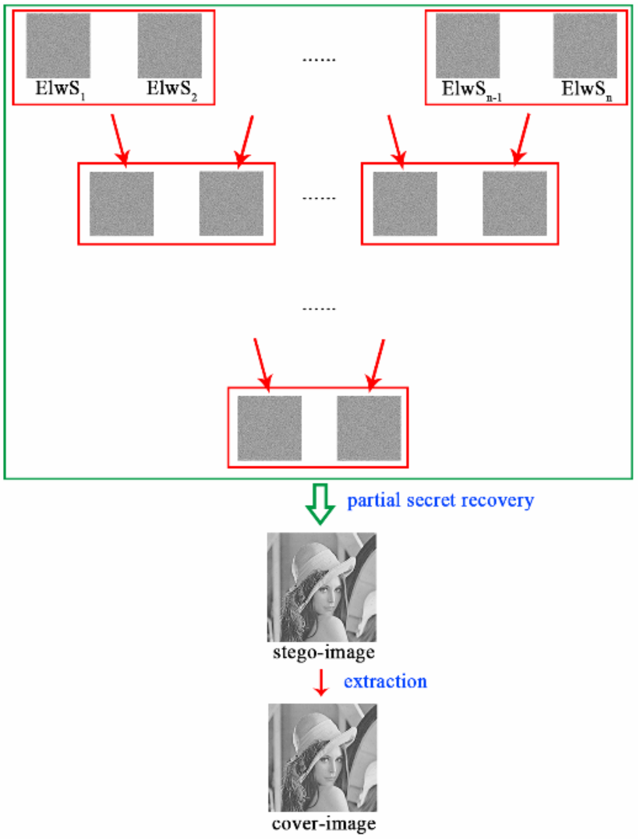


Figure 2. Partial decryption

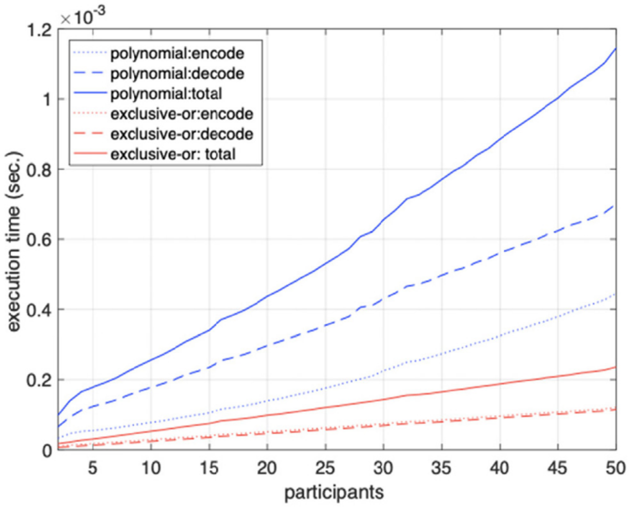


Figure 3. Performance

Experimental results are discussed as follows. Image quality and payload are two significant factors to evaluate a data hiding scheme. Thus, we take USC-SIPI image database [17] to measure image quality and payload of the proposed scheme. For payload, we consider the capacity as the following formulation.

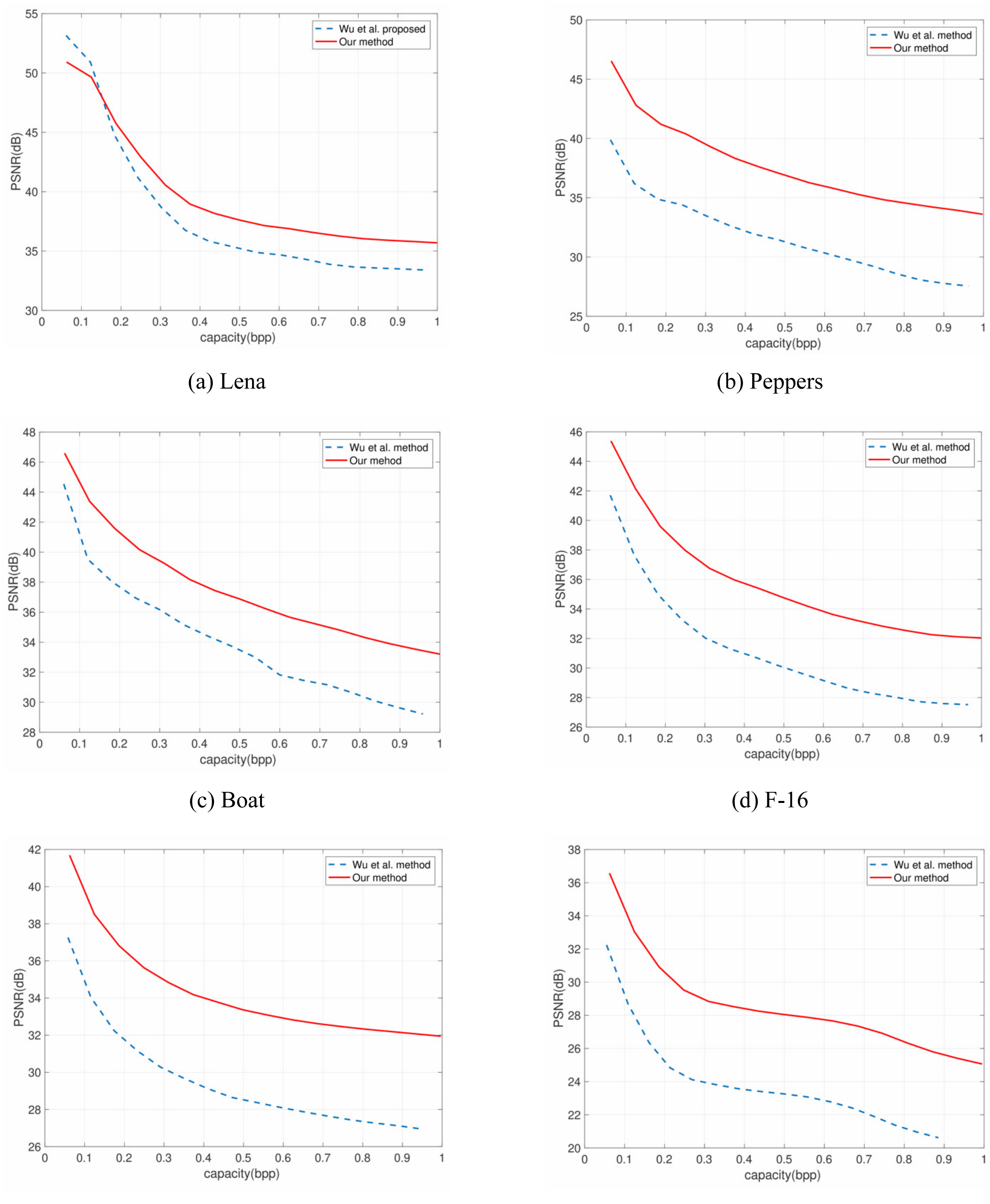
$$Cap = \frac{\text{Total number of all embedded bits of secret message}}{\text{Total number of pixels in a cover-image}}$$

At an intuitive level, Cap is 0.5 bpp if all of pixel pairs are suitable for hiding in our scheme. Since unsuitable ones usually exist in natural images, the payload should less than 0.5 bpp in practice. In addition to payload, image quality is considered to measure mean squared error (MSE) and peak signal-to-noise ratio (PSNR) as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(pi - pi')^2,$$

$$PSNR = 10\log_{10}(\frac{255^2}{MSE}),$$

where n is the total number of pixels, pi is the original pixel value in the cover-image, and pi is the modified pixel value in the stego-image. Figure 5 shows the six cover-images (512×512 pixels). In Figure 4, we show PRSR and payload of our and Wu et al.'s schemes.



(a) Lena

(b) Peppers

(c) Boat

(d) F-16

(e) House

(f) Baboon

Figure 4. PRSR and payload

(a) Lena

(b) Peppers

(c) Boat

(d) F-16

(e) house

(f) Baboon

Figure 5. Cover-images

4.1 Alternative Candidate

There is an alternative method to design the distributed EIRDH. Instead of XOR-based and Shamir secret sharing, we choose an additive secret sharing over modulo prime p for n-out-of-n setting. For example, the secret is $s \pmod p$, and then the first n-1 shares are randomly chosen over modulo p (denoted by $s_1, \ldots, s_{n-1}$). So, the last share is $s_n = s - \sum_{i=1}^{n-1} s_i \pmod p$. The secret recovery is trivial by collecting $s_1, \ldots, s_n$. The modification of our distributed EIDRH scheme includes the following two parts. The other steps are omitted, since they are very straight.

1. The operation is addition over the modulo p.
2. For embedding, each storage center computes $\frac{1}{n} \pmod p$ and then obtains $x_i'' = x_i' + \frac{1}{n}$ if the bit is 1.

5 Conclusions

In this paper, we present a cloud-edge model of securely image gathering. We propose a new distributed encrypted image-based reversible data hiding scheme from exclusive-OR secret sharing. This scheme only relies on XOR operations, which is with very low computational cost and suited for light-weight edge sensor devices. We provide implementations of our and Wu et al.'s schemes with respect to performance of encryption and decryption, and show our scheme is more efficient. Moreover, it preserves image quality and embedding capacity.

Acknowledgements

This work was supported in part by Ministry of Science and Technology of Taiwan, under grant MOST 106-2218-E-155-008-MY3 and 109-2628-E-155-001-MY3. We thank Zhong-Yi Guo, Song-Yi Hsu, and Jun-Rui Wang (Yuan Ze U.) for initial discussions for this work.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, *Computer Networks*, Vol. 38, No. 4, pp. 393-422, March, 2002.

[2] P. Paillier, Public-key Cryptosystems Based on Composite degree Residuosity Classes, in: J. Stern (Ed.), *Advances in*

Cryptology - EUROCRYPT '99, International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, 1999, pp. 223-238.

[3] J. Fridrich, D. Soukal, Matrix embedding for large payloads, *IEEE Transactions on Information Forensics and Security*, Vol. 1, No. 3, pp. 390-395, September, 2006.

[4] C. Munuera, Steganography and error-correcting codes, *Signal Processing*, Vol. 87, No. 6, pp. 1528-1533, June, 2007.

[5] J. Mielikainen, LSB matching revisited, *IEEE Signal Processing Letters*, Vol. 13, No. 5, pp. 285-287, May, 2006.

[6] X. Zhang, S. Wang, Efficient Steganographic Embedding by Exploiting Modification Direction, *IEEE Communications Letters*, Vol. 10, No. 11, pp. 781-783, November, 2006.

[7] J. Tian, Reversible Data Embedding Using a Difference Expansion, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 8, pp. 890-896, August, 2003.

[8] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Reversible data hiding, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 3, pp. 354-362, March, 2006.

[9] X. Zhang, Reversible Data Hiding in Encrypted Image, *IEEE Signal Processing Letters*, Vol. 18, No. 4, pp. 255-258, April, 2011.

[10] W. Hong, T. S. Chen, H. Y. Wu, An improved reversible data hiding in encrypted images using side match, *IEEE Signal Processing Letters*, Vol. 19, No. 4, pp. 199-202, April, 2012.

[11] M. S. Abdul Karim, K. Wong, Universal data embedding in encrypted domain, *Signal Processing*, Vol. 94, pp. 174-182, January, 2014.

[12] X. Zhang, Separable Reversible Data Hiding in Encrypted Image, *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 2, pp. 826-832, April, 2012.

[13] W. Zhang, K. Ma, N. Yu, Reversibility improved data hiding in encrypted images, *Signal Processing*, Vol. 94, pp. 118-127, January, 2014.

[14] X. Wu, J. Weng, W. Yan, Adopting secret sharing for reversible data hiding in encrypted images, *Signal Processing*, Vol. 143, pp. 269-281, February, 2018.

[15] A. Shamir, How to share a secret, *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, November, 1979.

[16] Y. C. Chen, T. H. Hung, S. H. Hsieh, C. W. Shiu, A New Reversible Data Hiding in Encrypted Image Based on Multi-Secret Sharing and Lightweight Cryptographic Algorithms, *IEEE Transactions on Information Forensics & Security*, Vol. 14, No. 12, pp. 3332-3343, December, 2019

[17] Usc-sipi image database. http://sipi.usc.edu/database/, 1977.

Biographies

Yu-Chi Chen received the B.S., M.S., and Ph.D. degrees from the Department of Computer Science and Engineering, National Chung-Hsing University, Taiwan, in 2008, 2009, and 2014, respectively. In 2013, he was a Visiting Scholar with the Department of Electrical Engineering, University of Washington. He was a Post-Doctoral Fellow at the Institute of Information Science, Academia Sinica, Taiwan, from 2014 to 2017. He is currently an Associate Professor with the Department of Computer Science and Engineering, Yaun Ze University, Taiwan. His research interests include cryptography and information security.

Chih-Wei Shiu received the Ph.D. degree from the Department of Computer Science and Engineering, National Chung Hsing University, Taiwan, in 2014. He is currently an Assistant Professor with the Department of Education Industry and Digital Media, National Taitung University, Taitung, Taiwan. His current research interests include steganography, digital game design, and visual reality.