Privacy-Preserving Reversible Watermarking for Data Exfiltration Prevention Through Lexicographic Permutations*

Ching-Chun Chang[1] and Chang-Tsun Li[2]

[1] Department of Computer Science, University of Warwick, United Kingdom
c.chang.2@warwick.ac.uk
[2] School of Computing and Mathematics, Charles Sturt University, Australia
chli@csu.edu.au

Abstract. Privacy-preserving reversible watermarking, as a subfield of secure signal processing, has received a growing research attention in the recent years due to privacy concerns in cloud computing. In this paper, we propose a novel reversible watermarking scheme for data exfiltration prevention. This scheme enables the cloud to embed labels that indicate the degree of confidentiality into the encrypted documents in such a way that the network administrator can monitor the document exfiltration through detecting the labels in the encrypted domain without compromising data privacy. An efficient watermarking algorithm is devised primarily based upon the concept of lexicographic permutations. In addition to this, a content-adaptive signal estimation mechanism is constructed for assisting host media recovery. Experimental results show that the proposed scheme outperforms the state-of-the-art with regards to watermarking capacity, fidelity, and recoverability.

Keywords: Cloud computing · Data exfiltration · Multimedia security · Privacy protection · Reversible watermarking · Stream cipher.

1 Introduction

Due to the advances in cloud computing technology, businesses and individuals have entrusted an increasing amount of data to the cloud for the purposes of processing and storage. In the meantime, there has been an increasing need for privacy protection. Although encryption is a widely used tool for protecting data against information leakage, conventional signal processing techniques become invalid in the encrypted domain. Towards addressing this problem, Rivest *et al.* introduced the concept of privacy homomorphisms [11], which opened up possibilities for performing computations upon encrypted data. As a challenging problem under this research field, privacy-preserving reversible watermarking

* This work was supported by Marie Skłodowska-Curie actions of EU Horizon 2020 programme through the project entitled 'Computer Vision Enabled Multimedia Forensics and People Identification' (Project No. 690907, Acronym: IDENTITY).

was primarily motivated by the fact that many watermarking algorithms are proprietary properties and thus any unauthorised use with commercial purposes may be considered as violation of copyrights. While one may entrust the task of watermarking to a licensed cloud service provider, privacy risks should be taken into account [1, 5, 10, 18]. From a theoretical point of view, the cloud is assumed to be an honest-but-curious or semi-honest party that is interested in learning the information from the protocol (*e.g.* the plaintext), but does not deviate from the protocol specification. This research problem is challenging since an imperceptible alteration in the ciphertext domain may cause a nontrivial distortion in the plaintext domain. If a cryptosystem is perfectly secure, it is theoretically not possible to foresee how a change in the ciphertext domain would result in a change in the plaintext domain.

The recent development of privacy-preserving watermarking schemes was primarily based upon homomorphic cryptosystems [2, 6, 12, 13, 16]. Despite a variety of mathematical operations permitted by homomorphic encryption, it is a resource-intensive task to implement the system due to high computational complexity and non-trivial ciphertext expansion. When taking the practicality into consideration, it is advisable to build watermarking schemes based upon conventional symmetric-key cryptosystems [4, 8, 9, 14, 17]. There are a variety of possible applications of privacy-preserving reversible watermarking. Consider a network administrator whose responsibility is to monitor data transmissions. It is of crucial importance to prevent classified documents from leakage beyond this point, and yet the authority to read the documents may not be granted to the administrator. To address this issue, we may embed a label that indicates the confidentiality of a given document as the watermark and design an algorithm that is able to detect the watermark without decrypting the file.

In this paper, we propose a novel reversible watermarking scheme compatible with a semantically secure symmetric-key cryptosystem. An overview of the proposed scheme is as follows: the encoding process utilises lexicographic permutations to embed watermarks into encrypted signals, whereas the decoding process extracts the watermarks and recovers the signals in aid of a content-adaptive signal estimation mechanism. Experimental results show a significant breakthrough over the state-of-the-art in watermarking capacity, fidelity and recoverability. The remainder of this paper is organised as follows. Section 2 presents the proposed watermarking scheme and signal estimation mechanism. Section 3 evaluates the scheme performance in comparison with the state-of-the-art. Section 4 concludes our work and outlines the directions for future research.

2 Proposed Scheme

An overview of the proposed scheme is illustrated in Fig. 1. To begin with, we introduce a privacy-preserving reversible watermarking scheme based upon lexicographic permutations and then present an updated scheme with detailed discussions. In addition to this, a content-adaptive signal estimation mechanism is constructed in order to realise the scheme in practice.

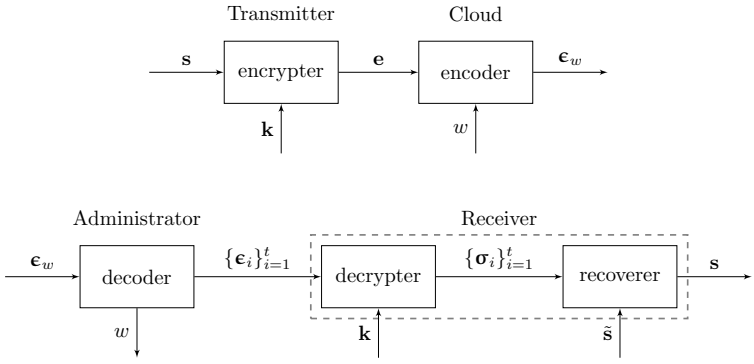


Fig. 1: An overview of the proposed scheme. The transmitter encrypts a host array of symbols $\mathbf{s}$ with a key $\mathbf{k}$ and then uploads the encrypted array $\mathbf{e}$ to the cloud. A watermark w is encoded into $\mathbf{e}$ producing a marked array $\boldsymbol{\epsilon}_w$, which is then sent to a network administrator for inspection. Depending on the decoded w, the transmission will either continue to move on or stop at this point. If the data is transmittable, at the receiving end, a permutation group containing the original encrypted array, denoted by $\{\boldsymbol{\epsilon}_i\}_{i=0}^t$, is generated. After decryption, a group containing the original array, denoted by $\{\boldsymbol{\sigma}_i\}_{i=0}^t$, is yielded. Eventually, the original array is restored with assistance of additional information, denoted by $\tilde{\mathbf{s}}$, obtained from a signal estimation mechanism.

2.1 A Permutation-Based Scheme

Consider the host signal as an 8-bit greyscale image. In order to satisfy the fidelity requirement, significant bit-planes should not be modified during watermark embedding process. To pave the way for presentation, we specify that the four most significant bit-planes are unmodifiable, though the scheme permits variations in implementation depending on different fidelity assessment models. Let us refer to the remaining four insignificant bit-planes as a nybble-plane, where the basic unit is a nybble, namely, a four-bit aggregation. While this nybble-plane is generally modifiable, only a portion of the nybbles are selected for carrying the payload and the rest part is kept intact for the purpose of reversing watermarking distortions. The selection follows a rule that each selected nybble is encircled by eight unselected immediate neighbours. The unselected nybbles will remain intact and be exploited for estimating the selected nybbles during the reverse process. A nybble can be represented by an integer between 0 and 15. Let a sequence of r modifiable nybbles be converted into an integer, referred to as a host symbol, between 0 and $N-1$, where $N = 2^{4r}$.

Let us divide the host symbols into non-overlapping arrays of length n and each array can be processed independently. Let $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ be an array of modifiable host symbols and $\mathbf{k} = (k_1, k_2, \ldots, k_n)$ be an array of randomly

generated key symbols. The transmitter encrypts the former with the latter by

$$\mathbf{e} \equiv \mathbf{s} + \mathbf{k} \pmod{N}. \tag{1}$$

Note that the array arithmetic operations are carried out element by element. Then, the transmitter uploads the enciphered array $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ along with the watermark w to the cloud, in which the watermark encoding is realised through lexicographic permutations. Before proceeding further, let us define the number of permutations of a given set. If the set of size n consists of n distinct elements, the number of permutations is simply the factorial of n, denoted by $n!$. If the set consists of repeated elements, then the multiplicity of each element shall be taken into account. Let M be a multiset of size n consisting of l distinct elements and the multiplicities of the elements be m_1, m_2, ..., m_l. The number of permutations of M is then given by

$$t = \frac{n!}{m_1! m_2! \ldots m_l!}. \tag{2}$$

Let $G_{\boldsymbol{\epsilon}} = \{\boldsymbol{\epsilon}_0, \boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_{t-1}\}$ be a group consisting of all the possible permutations of $\mathbf{e}$ sorted with lexicographic order, where $\boldsymbol{\epsilon}_u = \mathbf{e}$ and $0 \leq u \leq t-1$. A possible watermarking strategy is to encode a payload of $\log_2 t$ bits into one of the possible permutations. For instance, if an encrypted array $\mathbf{e}$ and a message $0 \leq w \leq t-1$ are encoded into $\boldsymbol{\epsilon}_w$, during the decoding process we can efficiently determine w as the lexicographic order of $\boldsymbol{\epsilon}_w$. Note that the watermark extraction process is carried out in the encrypted domain. As a result, the network administrator can inspect the decoded watermark to decide whether the host file has been given approval to be transmit beyond this point without actually inspecting the file itself. In other words, this scheme prevents data exfiltration without compromising data privacy. Apart from knowing the watermark information, we can also be certain about that $\mathbf{e}$ is one of the possible permutations of $\boldsymbol{\epsilon}_w$, though we are not able to recognise which it is in the absence of further information. At the receiving end, one may want to remove the distortions caused by watermarking. Since it is theoretically not possible to make inferences from the encrypted data, we decipher each possible one by

$$\boldsymbol{\sigma}_i \equiv \boldsymbol{\epsilon}_i - \mathbf{k} \pmod{N}, \tag{3}$$

and obtain $G_{\boldsymbol{\sigma}} = \{\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{t-1}\}$. As a result, we can employ signal processing techniques to analyse each σ_i and draw an inference on the original one in which some distinguishable structures may inhere. However, failed inferences may occur with high probability when we happen to process a sequence of intrinsically similar host symbols and a sequence of intrinsically similar key symbols. For example, consider $\mathbf{s} = (s_1, s_2)$ and $\mathbf{k} = (k_1, k_2)$ such that $s_1 \approx s_2$ and $k_1 \approx k_2$. We encrypt $\mathbf{s}$ into $\mathbf{e}$ and then encode $\mathbf{e}$ into either $\boldsymbol{\epsilon}_0$ or $\boldsymbol{\epsilon}_1$ depending on whether $w = 0$ or $w = 1$. Assume that $\mathbf{e}$ is of the 0-th permutation order, namely, $\boldsymbol{\epsilon}_0 = \mathbf{e}$, and accordingly

$$\begin{aligned} \boldsymbol{\epsilon}_0 &= (e_1, e_2) = (s_1 + k_1, s_2 + k_2), \\ \boldsymbol{\epsilon}_1 &= (e_2, e_1) = (s_2 + k_2, s_1 + k_1). \end{aligned} \tag{4}$$

In order to recover the original permutation, we decrypt respectively $\boldsymbol{\epsilon}_0$ and $\boldsymbol{\epsilon}_1$ into $\boldsymbol{\sigma}_0$ and $\boldsymbol{\sigma}_1$, as given by

$$\begin{aligned} \boldsymbol{\sigma}_0 &= \boldsymbol{\epsilon}_0 - \mathbf{k} = (s_1, s_2), \\ \boldsymbol{\sigma}_1 &= \boldsymbol{\epsilon}_1 - \mathbf{k} = (s_2 + k_2 - k_1, s_1 + k_1 - k_2). \end{aligned} \tag{5}$$

If $s_1 \approx s_2$ and $k_1 \approx k_2$, then $\boldsymbol{\sigma}_0 \approx \boldsymbol{\sigma}_1$. We can observe that in this case $\boldsymbol{\sigma}_0$ is indistinguishable from $\boldsymbol{\sigma}_1$.

2.2 An Updated Scheme

In the previous scheme, we permute $\mathbf{e}$ lexicographically and obtain a lexicon, or a codebook, for watermark encoding. To overcome the ambiguity in some extreme cases, we update the previous scheme by introducing an invertible transform to $\mathbf{e}$ prior to the creation of the lexicon. Let $\phi(N)$ be Euler's totient function which describes the number of positive integers up to N that relatively prime to N. A positive integer that is coprime to N is termed a totative of N. Suppose that $\mathbf{e}$ is the u-th permutation. We multiply $\mathbf{e}$ with the u-th totative, denoted by p_u, and obtain

$$\mathbf{e}' \equiv \mathbf{e} \cdot p_u \pmod{N}. \tag{6}$$

An important property of the above computation is that an inverse transform exists, which is given by

$$\mathbf{e} \equiv \mathbf{e}' \cdot q_u \pmod{N}, \tag{7}$$

where q_u is a unique modular multiplicative inverse of p_u with respect to the modulus N, that is,

$$p_u \cdot q_u \equiv 1 \pmod{N}. \tag{8}$$

A unique modular multiplicative inverse q_u exists if and only if p_u is coprime to N, that is, $\gcd(p_u, N) = 1$, where gcd stands for greatest common divisor. The number of permutations of $\mathbf{e}'$ is also t since the transform from $\mathbf{e}$ to $\mathbf{e}'$ is a bijective mapping.

We sort the permutations of $\mathbf{e}'$ lexicographically and form an ordered lexicon $G_{\boldsymbol{\epsilon}} = \{\boldsymbol{\epsilon}_0, \boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_{t-1}\}$ and encode a watermark w of $\log_2 t$ bits into one of the permutations yielding the marked result $\boldsymbol{\epsilon}_w$. In the decoding phase, w can be efficiently recognised by the order of $\boldsymbol{\epsilon}_w$. Let $\{p_0, p_1 \ldots, p_{t-1}\}$ be the first t totatives in $[0, N]$, and $\{q_1, \ldots, q_t\}$ be their respective modular multiplicative inverses. To restore the original array, we choose any $\boldsymbol{\epsilon}_i$ from the lexicon and multiply it with each multiplicative inverse yielding $G_{\boldsymbol{\alpha}} = \{\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{t-1}\}$, where

$$\begin{aligned} \boldsymbol{\alpha}_0 &\equiv \boldsymbol{\epsilon}_i \cdot q_0 \pmod{N}, \\ \boldsymbol{\alpha}_1 &\equiv \boldsymbol{\epsilon}_i \cdot q_1 \pmod{N}, \\ &\ldots \\ \boldsymbol{\alpha}_{t-1} &\equiv \boldsymbol{\epsilon}_i \cdot q_{t-1} \pmod{N}. \end{aligned} \tag{9}$$

Then, we sort the elements in each array with the lexicographic order in accordance to the array index and yield an updated group of arrays $G_{\boldsymbol{\beta}} = \{\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{t-1}\}$. For instance, the elements in $\boldsymbol{\alpha}_i$ is sorted with the i-th lexicographic order yielding $\boldsymbol{\beta}_i$. Note that the choice of $\boldsymbol{\epsilon}_i$ does not affect the resultant $G_{\boldsymbol{\beta}}$; in other words, any $\boldsymbol{\epsilon}_i$ yields the same group of results. The u-th array in $G_{\boldsymbol{\alpha}}$ is the original encrypted array with scrambled elements, whereas the u-th array in $G_{\boldsymbol{\beta}}$ is exactly the original encrypted array, namely $\boldsymbol{\beta}_u = \mathbf{e}$. Since it is not possible to distinguish the original array in the encrypted domain, we decipher each array in $G_{\boldsymbol{\beta}}$ and obtain $G_{\boldsymbol{\sigma}} = \{\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{t-1}\}$. With the aid of signal analysis techniques, we can retrieve the original one, namely $\boldsymbol{\sigma}_u$, with relatively low error rate. Let us see how this updated scheme is able to resolve the aforementioned ambiguity. Again, consider two host symbols and two key symbols such that $s_1 \approx s_2$ and $k_1 \approx k_2$. Assume that $\mathbf{e}$ is of the 0-th permutation order and accordingly $\mathbf{e}' = \mathbf{e} \cdot p_0$. Then, we encode $\mathbf{e}'$ into either $\boldsymbol{\epsilon}_0$ or $\boldsymbol{\epsilon}_1$ depending on the watermark bit. In the recovering phase, two possible candidates are generated by

$$\begin{aligned} \boldsymbol{\sigma}_0 &= \boldsymbol{\beta}_0 - \mathbf{k} = \mathrm{sort}(\boldsymbol{\alpha}_0, 0) - \mathbf{k} = \mathrm{sort}(\boldsymbol{\epsilon}_i \cdot q_0, 0) - \mathbf{k}, \\ \boldsymbol{\sigma}_1 &= \boldsymbol{\beta}_1 - \mathbf{k} = \mathrm{sort}(\boldsymbol{\alpha}_1, 1) - \mathbf{k} = \mathrm{sort}(\boldsymbol{\epsilon}_i \cdot q_1, 1) - \mathbf{k}, \end{aligned} \tag{10}$$

where $\mathrm{sort}(\mathbf{x}, i)$ denote a sorting function that sorts the elements of an array $\mathbf{x}$ according to the i-th lexicographic permutation, and $\boldsymbol{\epsilon}_i$ can be either $\boldsymbol{\epsilon}_0$ or $\boldsymbol{\epsilon}_1$. We further derive that

$$\mathrm{sort}(\boldsymbol{\epsilon}_i \cdot q_0, 0) - \mathbf{k} = ((s_1 + k_1)p_0q_0 - k_1, (s_2 + k_2) \cdot p_0q_0 - k_2) = (s_1, s_2), \tag{11}$$

and $\boldsymbol{\sigma}_1$ equals to either

$$\mathrm{sort}(\boldsymbol{\epsilon}_i \cdot q_1, 1) - \mathbf{k} = ((s_1 + k_1)p_0q_1 - k_1, (s_2 + k_2) \cdot p_0q_1 - k_2), \tag{12}$$

or

$$\mathrm{sort}(\boldsymbol{\epsilon}_i \cdot q_1, 1) - \mathbf{k} = ((s_2 + k_2)p_0q_1 - k_1, (s_1 + k_1) \cdot p_0q_1 - k_2). \tag{13}$$

In either case, two candidates are not likely to be similar since the term p_uq_v, where $u \neq v$, thoroughly randomise the incorrect candidate; in other words, the original array should be very distinguishable from a sequence of random numbers with high probability.

2.3 A Content-Adaptive Estimator

To complete the proposed scheme, we devise a signal estimation mechanism for assisting host signal recovery. As aforementioned, for an 8-bit greyscale image we embed payloads into selected low nybbles while each of which is encircled by eight unselected immediate neighbours, as illustrated in Fig. 2. The aim is to estimate the low nybble of a pixel p_0 with the aid of the high nybble of p_0 and the neighbouring pixels $p_1, p_2, \ldots, p_8$. Image regions can be roughly divided into smooth patches, edges and complex textures. Due to the fact that the statistical

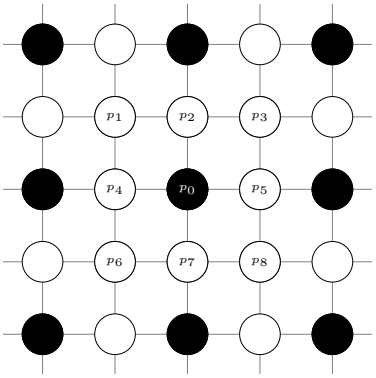


Fig. 2: Pixels at the black positions are modifiable in terms of their low nybbles, whereas those at the white positions are unmodifiable.

distribution of pixel values varies a lot in different regions, we have to identify the class the observed p_0 belongs to. Let

$$M(X) = \frac{1}{\gamma}\sum_{i=1}^{\gamma}|x_i - \mu(X)| \tag{14}$$

represents the mean absolute deviation (MAD), where $\mu(X)$ represents the arithmetic mean and γ represents the number of elements in a given set X. The score for smooth patches is given by

$$\delta_{smth} = M(p_1, p_2, \ldots, p_8), \tag{15}$$

and the scores for different degrees of edges are given by

$$\begin{aligned}
\delta_{0^\circ} &= \frac{M(p_1,p_2,p_3) + M(p_4,p_5) + M(p_6,p_7,p_8)}{3},\\
\delta_{45^\circ} &= \frac{M(p_2,p_4) + M(p_3,p_6) + M(p_5,p_7)}{3},\\
\delta_{90^\circ} &= \frac{M(p_1,p_4,p_6) + M(p_2,p_7) + M(p_3,p_5,p_8)}{3},\\
\delta_{135^\circ} &= \frac{M(p_2,p_5) + M(p_1,p_8) + M(p_4,p_7)}{3}.
\end{aligned} \tag{16}$$

Let the minimum value of $\{\delta_{smth}, \delta_{0^\circ}, \delta_{45^\circ}, \delta_{90^\circ}, \delta_{135^\circ}\}$ be denoted by δ_{min}. If δ_{min} is no greater than a threshold θ (empirically $\theta = 15$), then we calculate an anticipated value for p_0 by

$$\tilde{p}_0 = \begin{cases}
\mu(p_1, p_2, \ldots, p_8) & \text{if } \delta_{min} = \delta_{smth},\\
\mu(p_4, p_5) & \text{if } \delta_{min} = \delta_{0^\circ},\\
\mu(p_3, p_6) & \text{if } \delta_{min} = \delta_{45^\circ},\\
\mu(p_2, p_7) & \text{if } \delta_{min} = \delta_{90^\circ},\\
\mu(p_1, p_8) & \text{if } \delta_{min} = \delta_{135^\circ}.
\end{cases} \tag{17}$$

Otherwise, an anticipated value for p_0 is determined by the closest value in the neighbouring area, that is,

$$\tilde{p}_0 = \arg\min_{p_i} \mid p_i - p_0 \mid, \tag{18}$$

where $i \in \{1, 2, \ldots, 8\}$. Finally, we estimate p_0's low nybble in such a way that the resultant pixel value approeaches the anticipated value $\tilde{p}_0$, as formulated by

$$\tilde{p}_0^* = \arg\min_{p_{0,j}} \mid p_{0,j} - \tilde{p}_0 \mid, \tag{19}$$

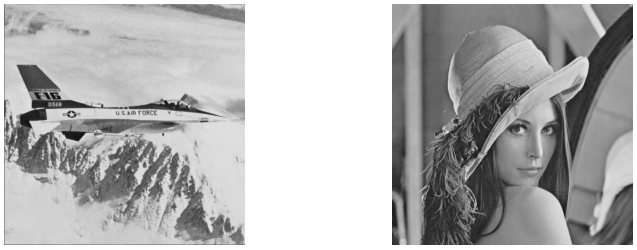
where $j \in \{0, 1, \ldots, 15\}$ and $p_{0,j}$ denotes a value generated by setting p_0's low nybble to one of the possible patterns.

3 Experiments

In the experiments, greyscale images of size 512×512 with 256 tonal options are used as the host media, as shown in Fig. 3. The scheme utilises a synchronous stream cipher to encrypt images and is therefore semantically secure. Let the number of host symbols in each array be fixed to $n = 4$. Each symbol is formed by r modifiable nybbles, where r is set to 2, and correspondingly the symbol values lie in the range from 0 to 255. We compare the proposed scheme with the state-of-the-art schemes [3, 7, 15] in terms of the fidelity and recoverability. Let $p_{i,j}$ denote the pixel at the i-th row and the j-th column, and $\hat{p}_{i,j}$ denote its noisy counterpart. We evaluate the image quality by peak signal-to-noise ratio (PSNR). As can be seen from Fig. 4, the proposed scheme outperforms the previous methods with regard to the fidelity of marked images under the same embedding rate. As reported in Fig. 5, the proposed scheme also achieves the best results with respect to the fidelity of recovered images given the same amount of payload. Overall, it is evident that the proposed scheme achieves a substantial improvement in algorithm performance.

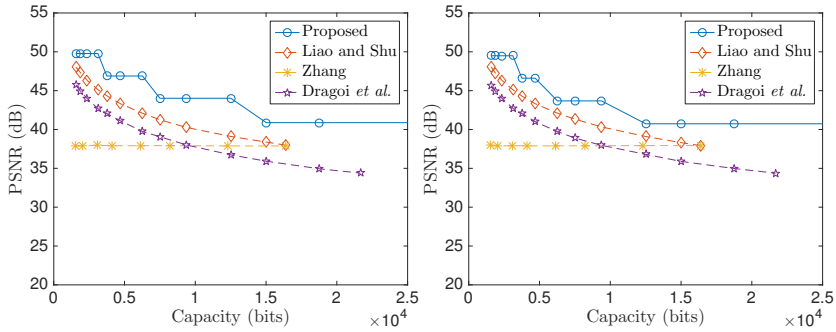
4 Conclusions

In this paper, a novel reversible watermarking scheme is proposed to embed payload into encrypted images via lexicographic permutations. The scheme is compatible with a synchronous stream cipher and is therefore semantically secure. We derive further an updated version of the scheme in order to minimise the error rate in host recovery. In addition to this, a content-adaptive signal estimation mechanism is devised for supporting the recovery process. Experimental results show a remarkable breakthrough over the state-of-the-art in capacity, fidelity, and recoverability. It is expected that the research in this field will continue to move forwards in the future, and from our perspective, further minimisation of error rate in host recovery entails further investigation.



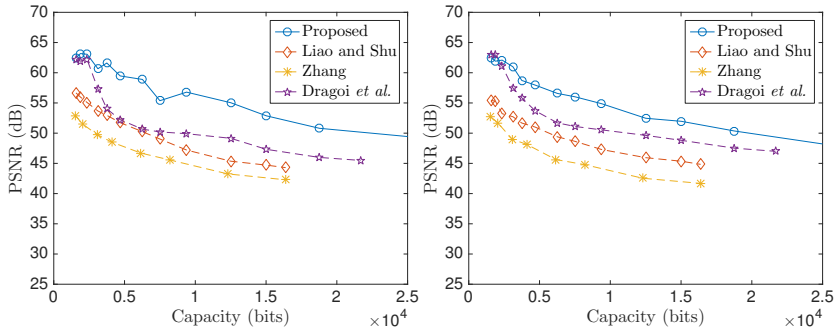
(a) Airplane (b) Lena

Fig. 3: Test images.



(a) Airplane (b) Lena

Fig. 4: Fidelity comparisons. The horizontal axis displays the watermarking capacity, whereas the vertical axis shows the PSNR of marked images.



(a) Airplane (b) Lena

Fig. 5: Recoverability comparisons. The horizontal axis depicts the watermarking capacity, whereas the vertical axis presents the PSNR of recovered images.

References

1. Cao, X., Du, L., Wei, X., Meng, D., Guo, X.: High capacity reversible data hiding in encrypted images by patch-level sparse representation. IEEE Trans. Cybern. **46**(5), 1132–1143 (May 2016)
2. Chen, Y.C., Shiu, C.W., Horng, G.: Encrypted signal-based reversible data hiding with public key cryptosystem. J. Visual Commun. Image Representation **25**(5), 1164–1170 (July 2014)
3. Dragoi, I.C., Coanda, H.G., Coltuc, D.: Improved reversible data hiding in encrypted images based on reserving room after encryption and pixel prediction. In: Proc. European Signal Process. Conf. (EUSIPCO). pp. 2186–2190. Kos, Greece (Aug 2017)
4. Hong, W., Chen, T.S., Wu, H.Y.: An improved reversible data hiding in encrypted images using side match. IEEE Signal Process. Lett. **19**(4), 199–202 (Apr 2012)
5. Huang, F., Huang, J., Shi, Y.Q.: New framework for reversible data hiding in encrypted domain. IEEE Trans. Inf. Forensics Security **11**(12), 2777–2789 (Dec 2016)
6. Li, M., Li, Y.: Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding. Signal Process. **130**, 190–196 (Jan 2017)
7. Liao, X., Shu, C.: Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. J. Visual Commun. Image Representation **28**, 21–27 (Apr 2015)
8. Puech, W., Chaumont, M., Strauss, O.: A reversible data hiding method for encrypted images. In: Proc. SPIE. vol. 6819, pp. 68191E–1–68191E–9. San Jose, CA, USA (Feb 2008)
9. Qian, Z., Zhang, X.: Reversible data hiding in encrypted images with distributed source encoding. IEEE Trans. Circuits Syst. Video Technol. **26**(4), 636–646 (Apr 2016)
10. Qian, Z., Zhang, X., Wang, S.: Reversible data hiding in encrypted JPEG bitstream. IEEE Trans. Multimedia **16**(5), 1486–1491 (Aug 2014)
11. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. In: DeMillo, R.A., et al. (eds.) Foundations of Secure Computation, pp. 169–180. Academic Press (1978)
12. Wu, H.T., Cheung, Y.M., Huang, J.: Reversible data hiding in Paillier cryptosystem. J. Visual Commun. Image Representation **40, pt. B**, 765–771 (Oct 2016)
13. Wu, X., Chen, B., Weng, J.: Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer. J. Visual Commun. Image Representation **41**, 58–64 (Nov 2016)
14. Zhang, X.: Reversible data hiding in encrypted image. IEEE Signal Process. Lett. **18**(4), 255–258 (Apr 2011)
15. Zhang, X.: Separable reversible data hiding in encrypted image. IEEE Trans. Inf. Forensics Security **7**(2), 826–832 (Apr 2012)
16. Zhang, X., Long, J., Wang, Z., Cheng, H.: Lossless and reversible data hiding in encrypted images with public-key cryptography. IEEE Trans. Circuits Syst. Video Technol. **26**(9), 1622–1631 (Sept 2016)
17. Zhang, X., Qian, Z., Feng, G., Ren, Y.: Efficient reversible data hiding in encrypted images. J. Visual Commun. Image Representation **25**(2), 322–328 (Feb 2014)
18. Zhou, J., Sun, W., Dong, L., Liu, X., Au, O.C., Tang, Y.Y.: Secure reversible image data hiding over encrypted domain via key modulation. IEEE Trans. Circuits Syst. Video Technol. **26**(3), 441–452 (Mar 2016)