Semi-fragile Watermarking Algorithm Based on Arnold Scrambling for Three-Layer Tamper Localization and Restoration

Bin Feng[1,2], Xiangli Li[1], Yingmo Jie[1,2], Cheng Guo[1,2(✉)], and Huijuan Fu[3,4]

[1] School of Software Technology, Dalian University of Technology, Dalian 116620, People's Republic of China
{fengbin,guocheng}@dlut.edu.cn, dllgdxlxl@foxmail.com, jymsf2015@mail.dlut.edu.cn

[2] Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Tuqiang Street, Dalian 116620, People's Republic of China

[3] School of Information Management, Wuhan University, Wuhan 430014, People's Republic of China
huijuanfu@163.com

[4] School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, People's Republic of China

Abstract. To protect the content integrity, authenticity and improve the effect of tamper localization and recovery, this paper designs and implements a semi-fragile watermark based on Arnold transformation, which is used to localize and recover tamper of confused image and plain-image. The sender encodes the watermark into the 2-bit least significant bit of the pixel of the original image, and the authentication watermark consists of the pixel value comparison result and the parity check code; the recovery watermark is the pixel value of the Torus image block. In the detection side, the plain-image adopts the stratified idea, carries on the three-level tamper localization and recovery, the third-party authentication institution can detect tamper of the scrambled image using the layer detection method, the receiver will detect the positioning result again. The experimental results show that the proposed algorithm can accurately locate tamper and realize the content recovery and effectively prevent the vector quantization attack. Compared with other algorithms, this algorithm has better effect of tamper localization and recovery.

Keywords: Semi-fragile watermark · Arnold scrambling
Hierarchical tamper localization · Tamper recovery
Torus self-isomorphism mapping

1 Introduction

In order to protect the integrity and authenticity of image, this paper proposes a new semi-fragile digital watermarking algorithm based on the literature [1,2],

L. Wang et al. (Eds.): QShine 2017, LNICST 234, pp. 71–80, 2018.
https://doi.org/10.1007/978-3-319-78078-8_8

which is used for image content integrity authentication, and according to the algorithm [3], we change our method for better effect.

In this chapter, we propose the related work of our method. We shall study the related work and change some parts of them to suit for our algorithm preferably [4].

1.1 Torus Automorphism Mapping

Torus isomorphic mapping is a typical chaotic map. In this method, a point is mapped to another different point, and for each point there is only one corresponding mapping point.

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = A \times \begin{bmatrix} x_n \\ y_n \end{bmatrix} \pmod{N} \tag{1}$$

A is a matrix of 2×2, like $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $\det A = 1$.

In this paper, we use this for the selection of watermark embedded position. Since the sequence of image blocks is a one-dimensional sequence, the Torus mapping is transformed into a one-dimensional transformation formula.

$$X^{'} = f(X) = (k \times X) \bmod N + 1 \tag{2}$$

$X, X^{'} (\in [1, N])$ are respectively the current serial number and the mapping number; $k (\in [0, N-1])$ must be a prime number and belong to a private key; $N (\in Z - \{0\})$ is the total number.

1.2 Arnold Image Scrambling Algorithm

Arnold Scrambling is proposed by Russian mathematician Vladimir l. Arnold, also known as cat face transformation. Arnold scrambling has a periodicity, and after multiple transformations, the image will become very chaotic, but after specific transformations, re-transformed into the initial image. Such transformation can be used as image encryption [9].

In Arnold scrambling, the image is digitized into a matrix, and the rows and columns of its elements correspond to the values of the arguments, and the values of the elements represent image information. The position (x', y') of the matrix in one transformation is

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} \pmod{N} \tag{3}$$

$x, y \in \{0, 1, 2, \cdots, N-1\}$ indicates the position of the pixel before transformation. Digital images can be seen as a two-dimensional matrix, and after Arnold transformation, the pixel position will be rearranged, so the image will appear chaotic to achieve the effect of scrambling encryption.

2 Proposed Method

In this paper, based on the literature [1,5], our algorithm is proposed for hierarchical tamper localization and restoration, which can be applied to both plain-image and scrambling images. Wherein tamper localization is based on the three-layer detection [1], and the effective recovery depends on the pixel information embedded in the Torus mapping block. The three-layer localization is carried out directly on the plain-image, and the tamper of the scrambling image can be detected on the cloud side, and we can decrypt the result and carry on the secondary detection for a better effect. The following sections describe the process of the watermark embedding, plain-image tamper detection, confused image tamper localization and recovery [6,8].

2.1 Based on Block Watermark Embedding

In this section, the original image is preprocessed to generate the watermark, and the watermark is embedded according to the Torus automorphism mapping. The watermark is embedded in the lowest 2 bits of each pixel.

2.1.1 Pretreatment

Assuming the original image I is 256 gray levels, its size is $M \times M$, where M is a multiple of 2. The image is segmented and the block mapping sequence $A \rightarrow B \rightarrow C \rightarrow D \rightarrow \ldots \rightarrow A$ is obtained by the Torus automorphism transformation. Each letter in the sequence represents a separate block. That means the pixel value of block A is embedded in block B, the pixel value of block B is embedded in block C, and so on.

Firstly, we divide the image I into 2×2 blocks and number them. Secondly, calculate their Torus mapping blocks.

2.1.2 Watermark Generation and Embedding

Assuming A and B are a pair of Torus automorphism mapping blocks in the image I.

The watermark of the image block B is represented by an array(v, p, r), where v, p are one bit, and r is 6 bits determined by the pixel value of A. The generation of watermark and the embedding process are as follows:

Step 1 : The 2-bit LSB of the pixels of B is set to zero.
Step 2 : Generates authentication watermark v of the block B.

$$v = \begin{cases} 1 \; B_{14} > B_{23} \\ 0 \; B_{14} \leq B_{23} \end{cases} \tag{4}$$

Step 3 : Calculate the 6 bit MSB average B_{avg} of image block B.
Step 4 : Calculate the quantity N of 1 in B_{avg}, and the parity watermark p.

$$p = \begin{cases} 1 \; N \rightarrow Even \\ 0 \;\; N \rightarrow Odd \end{cases} \tag{5}$$

Step 5 : The average A_{avg} of the 6 bit MSB of the image block A is as the recovery watermark r.

Step 6 : The watermark (v, p, r) are composed of 8 bits, and then embedded into the 8-bit LSB of the four pixels of the image block B.

Repeat the above steps (1) to (6) for the other blocks, obtain the embedded image I'.

2.2 Arnold Transformation

Divide the embedded image I' into 2×2 image blocks, and take the image block A for an example.

Step 1 : The coordinate of the first pixel point of the block A are (x_A, y_A), and the other coordinates are calculated as $(x_A, y_A+1), (x_A+1, y_A), (x_A+1, y_A+1)$.

Step 2 : Assume the private key is (a, b, N), and after Arnold transformation, the coordinate (x_A, y_A) is converted to $(x_A{}', y_A{}')$.

$$\begin{bmatrix} x_A{}' \\ y_A{}' \end{bmatrix} = \begin{bmatrix} 1 & b \\ a & ab+1 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \end{bmatrix} \pmod{M} \tag{6}$$

Step 3 : The pixels $(x_A, y_A+1), (x_A+1, y_A), (x_A+1, y_A+1)$ of the block A are respectively converted into $(x_A{}', y_A{}'+1), (x_A{}'+1, y_A{}'), (x_A{}'+1, y_A{}'+1)$.

The Arnold scrambling image I'_{arnold} can be obtained by repeating the above steps (1) to (3) on other image blocks.

2.3 Tamper Detection

2.3.1 Tamper Detection of Plain-Images

The tampered image I'_w is detected in three layers. In the first layer, we detect the 2×2 image blocks. And in the second layer, we mark the independent 4×4 blocks that has more than one marked 2×2 block. In the third layer, mark the independent blocks according to the surrounding image blocks.

In the first detection, the image I'_w is divided into independent 2×2 image blocks. Take the block B' as an example and the specific steps are as follows:

Step 1 : The watermark (v, p) in the image block B' is extracted according to the embedding rules.

Step 2 : Set the 2 bit LSB of the pixels of B' to 0, and calculate the average pixel value B'_{avg} of B'.

Step 3 : Calculate the quantity N' of 1 in B'_{avg} and the parity code p'.

Step 4 : If $p' = p$, the image block B' is authenticated, otherwise the image block is marked.

Step 5 : When the parity code p' is verified, the image block B' is evaluated for the watermark v'.

Step 6 : If $v' = v$, the image block B' is authenticated, otherwise the image block B' is marked.

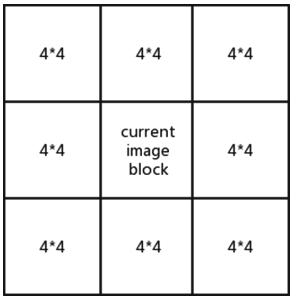

Fig. 1. Secondary tampering localization image block

Repeat the above steps (1) to (6) for other image blocks of I'_w, and the detection result I_{locate} is acquired.

In the second detection, the localization image I_{locate} is divided into independent 4×4 image blocks and each individual image block is divided into four 2×2 image blocks. And mark each individual 4×4 image block that has more than one marked 2×2 block, and finally obtain the second localization image I'_{locate}.

In the third detection, the second localization image I'_{locate} is divided into non-overlapping 4×4 image blocks, and as shown in Fig. 1, the image block is marked where there are more than five marked image blocks of the eight surrounding blocks. After that, we get the final localization image I''_{locate}.

2.3.2 Tamper Detection of Scrambled Images

Assume the scrambled image I'_{Arnold} requires tamper detection in an unsafe third party, the insecure cloud detection system is A_{cloud}, and the local security detection system is B_{locate}, then the first layer of the confused image is detected in the cloud detection system. A_{cloud} send detection results to the local security detection system for 2, 3 layer detection. The specific steps are as follows:

Step 1 : At the cloud system, calculate the localization image I^{cloud}_{locate} like Sect. 2.3.1.

Step 2 : In the local detection system, use the private key to decrypt I^{cloud}_{locate} to get the localization image $I^{location}_{locate}$.

Step 3 : In the second detection, the localization image $I^{location}_{locate}$ is divided into independent 4×4 image blocks and it is detected whether there is a marked independent 2×2 image block in each individual 4×4 image block. And finally get the second localization image $I'^{location}_{locate}$

Step 4 : In the third detection, the localization image $I'^{location}_{locate}$ is divided into 4×4 image blocks. Mark the image block where there are more than five marked surrounding image blocks, and finally get the localization image $I''^{location}_{locate}$.

2.4 Tampering Recovery

After the above tamper detection, we need to recover the image. So assume the image block B' has a tamper mark, and take B' for an example.

Step 1 : The image block $C^{'}$ is calculated according to the key k of the Torus transformation.

Step 2 : If the image block $C^{'}$ is not marked with tamper, extract the recovery watermark r, shift r left twice, and get r' to recover the image block B'.

Step 3 : The pixel value of the image block $B^{'}$ is replaced with $r^{'}$.

Step 4 : If the image block C' has a tamper mark, the image block B' is re-marked.

Repeat the steps (1) to (4) for all the image blocks, and finally obtain the recovery image $I_{recover}$. Because there are some image blocks that are not recovered, preform the following operations.

Step 1 : Calculate the average $B'_{surroud}$ of the surrounding recovered image blocks around the image block B'

Step 2 : Recover the image block B' according to $B'_{surround}$.

The above operations (1) to (2) are performed for each unrecovered image blocks to obtain the final recovery image $I'_{recover}$.

3 Results and Analysis

In this paper, the gray images Peppers, Lena, Plane, Baboon are used as test images. The peak signal to noise ratio and the structure similarity of the image are used to measure the ability of localization and recovery [7].

3.1 Peak Signal-to-Noise Ratio and Image Structure Similarity

3.1.1 Peak Signal-to-Noise Ratio

Assume the images are the reference image f and the test image g, whose size is $M \times N$, the calculation formula between f and g is as follows.

$$MSE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_{ij} - g_{ij})^2 \tag{7}$$

$$PSNR(f,g) = 10\log_{10}(255^2/MSE(f,g)) \tag{8}$$

When MSE approaches zero, the PSNR is near infinity, that indicates higher PSNR provides higher image quality. The peak signal-to-noise ratio can reflect the mean square error between the watermark image and the original image. The larger value shows the smaller difference between the embedded image and the original image.

3.1.2 Image Structure Similarity

Structured similarity is not designed using a traditional error summation method, but by modeling any image distortion as a combination of three factors, which are correlation loss, luminance distortion, and contrast distortion. Assume the images are the reference image f and the test image g whose size is $M \times N$, the formula is as follows.

$$\begin{cases} l(f,g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1} \\ c(f,g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2} \\ s(f,g) = \frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3} \end{cases} \tag{9}$$

$$SSIM(f,g) = l(f,g) \times c(f,g) \times s(f,g) \tag{10}$$

μ_f, μ_g are the average of the reference image f and the test image g, σ_f, σ_g stand for their standard deviation, σ_f^2, σ_g^2 are their variance, σ_{fg} is the covariance. In order to avoid the above formula denominator to 0, C_1, C_2 and C_3 are constants. In general, $C_1 = (K_1 \times L)^2$, $C_2 = (K_2 \times L)^2$, $C_3 = C_2/2$. Usually, $K_1 = 0.01$, $K_2 = 0.03$, $L = 255$.

We use the peak signal to noise ratio and structured similarity of the image to measure the image quality. Figure 2 shows the original image of Lena, Peppers, Plane and Baboon, and the image after adding watermark. As shown in Table 1, this method increases the PSNR after embedding the watermark, and this paper has great superiority in embedding the watermarking invisibility.

As we can see, the performance of our algorithm has the high peak signal-to-noise ratio and image structure similarity, which reflect the mean square error between the watermark image and the original image. The larger value shows the smaller difference between the embedded image and the original image.

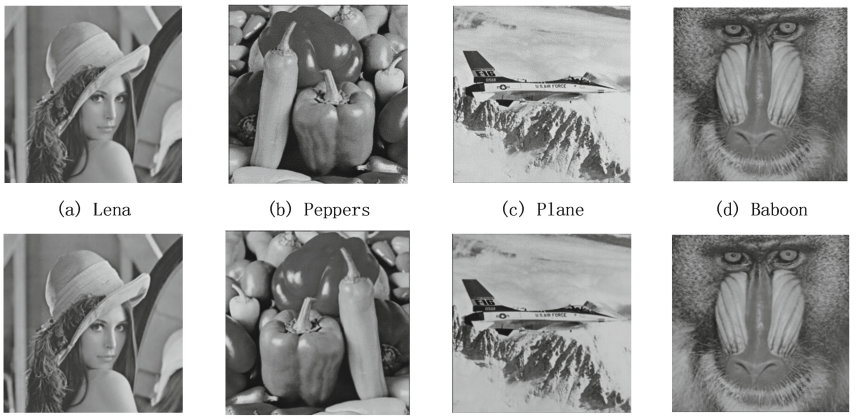

Fig. 2. The effect of the images embedded watermark

Table 1. The $PSNR$ and $SSIM$ of images

Image	Lena	Peppers	Plane	Baboon
PSNR	47.16	47.10	47.29	47.53
SSIM	0.9795	0.9825	0.9777	0.9930

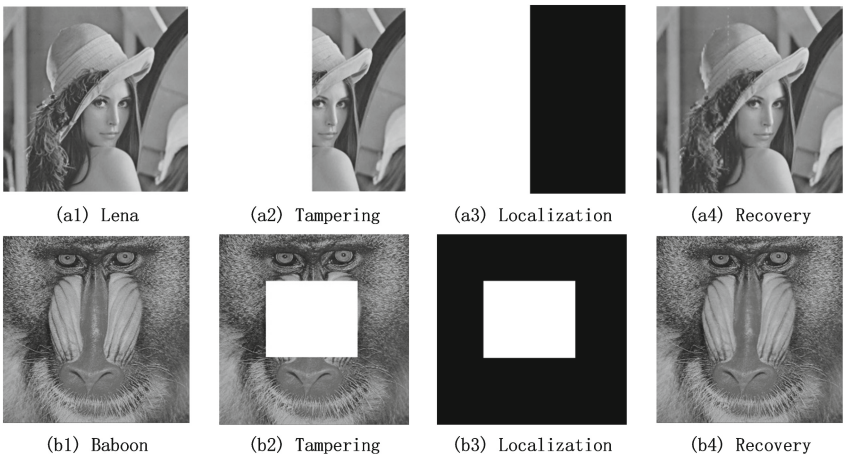

Fig. 3. The effect of localization and recovery

3.2 Test and Analysis

3.2.1 Result in Plain-Image

As shown in Fig. 3(a2), (b2) are the tampering image after attack and Fig. 3(a3), (b3) are the localization results. And Fig. 3(a4), (b4) are the recovery results, we can see this method has a great advantage in resisting attack.

After the shear attack, we can see that the image Lena has the half lost and our algorithm can localize the attack precisely and perfectly. Besides, through recovery of our method, the image has very little difference compared with the original image, which shows that our algorithm has unparalleled superiority.

3.2.2 Result in Scrambling Image

The algorithm can directly locate the scrambling image, and it has the same effect compared with the plain-image. The experimental results are shown in Fig. 4. As we can see, tamper can be localized in the scrambling image and the image can be recovery accurately, that is a great innovation, and the result in the scrambling image is still as well as the plain-image

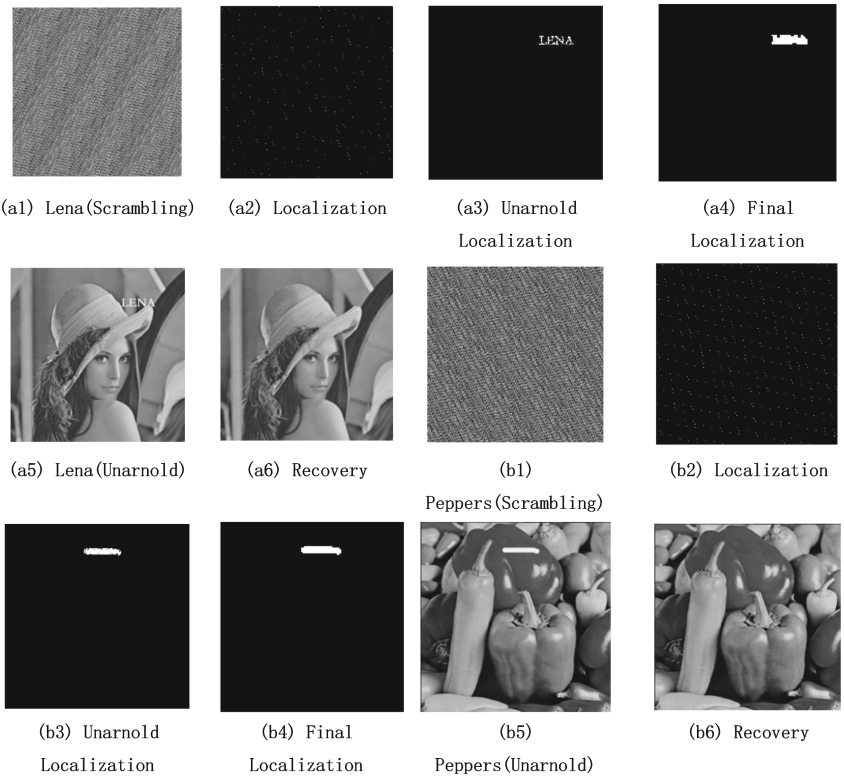

Fig. 4. Localization and recovery of scrambling image

4 Conclusion

This paper presents a semi-fragile image digital watermarking algorithm for plain-image and scrambling image. The main features of this algorithm include the following aspects:

(1) The hierarchical idea is used to locate the tampering position, and it has better anti-shear attack ability, and adds a recovery watermark for tampering recovery. The authentication watermark is composed of the parity check code and the comparison result; the recovery watermark is the average pixel value.

(2) The embedded algorithm uses the well-known spatial domain LSB algorithm. The aim is to improve the tampering recovery effect. The algorithm is simple in principle, has higher localization accuracy and better recovery effect.

(3) This paper use three layers to detect and localize tamper. In this algorithm, the experimental verification can detect the location of tampering in the image, and can effectively recover the tampering content, and can effectively prevent the vector quantization attack.

(4) This algorithm can directly localize tamper in the scrambling conditions, and it can detect tamper without revealing the plain-image, greatly improve privacy and security of the image.

Acknowledgements. This paper is supported by the National Science Foundation of China under grant No. 61401060, 61501080, 61572095 and 61771090, the Fundamental Research Funds for the Central Universities' under No. DUT16QY09, and the Social Science Foundation of Jiangxi Province, China No. 15JY48.

References

1. Celik, M.U., et al.: Hierarchical watermarking for secure image authentication with localization. IEEE Trans. Image Process. **11**(6), 585 (2002). A Publication of the IEEE Signal Processing Society
2. Potdar, V.M., Han, S., Chang, E.: A survey of digital image watermarking techniques (2005)
3. Cox, I.J., Kilian, J., Leighton, F.T., et al.: Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. **6**(12), 1673–1687 (1997). A Publication of the IEEE Signal Processing Society
4. Walton, S.: Information authentication for a slippery new age. Dr. Dobbs J. **20**(4), 18–26 (1995)
5. Fridrich, J., Goljan, M.: Images with self-correcting capabilities. In: Proceedings of the International Conference on Image Processing, ICIP 1999, vol. 3, pp. 792–796. IEEE (2002)
6. Sikder, I., Dhar, P.K., Shimamura, T.: A semi-fragile watermarking method using slant transform and LU decomposition for image authentication. In: International Conference on Electrical, Computer and Communication Engineering, pp. 881–885. IEEE (2017)
7. Hore, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: International Conference on Pattern Recognition, pp. 2366–2369. IEEE (2010)
8. Liu, Q., Jiang, X., et al.: A unified digital watermark algorithm based on singular value decomposition and spread spectrum technology. Acta Electron. Sin. **4**, 621–624 (2005)
9. Arnold, V.I.: Geometrical Methods in the Theory of Ordinary Differential Equations, 2nd edn., 351 pp. Springer, New York (1988). https://doi.org/10.1007/978-1-4612-1037-5. Rota, G.C. Adv. Math. **80**(2), 269 (1990)
10. Qiu, T., Zhao, A., Xia, F., Si, W., Wu, D.O.: ROSE: robustness strategy for scale-free wireless sensor networks. IEEE/ACM Trans. Networking **25**(5), 2944–2959 (2017)
11. Qiu, T., Qiao, R., Wu, D.O.: EABS: an event-aware backpressure scheduling scheme for emergency internet of things. IEEE Trans. Mobile Comput. **17**(1) (2017). https://doi.org/10.1109/TMC.2017.2702670
12. Guo, C., Zhuang, R., Jie, Y., Ren, Y., Wu, T., Choo, K.-K.R.: Fine-grained database field search using attribute-based encryption for E-healthcare clouds. J. Med. Syst. **40**(11), 235:1–235:8 (2016)