Secure Image Encryption without Size Limitation Using Arnold Transform and Random Strategies

Zhenjun Tang and Xianquan Zhang
Department of Computer Science, Guangxi Normal University, Guilin 541004, P.R. China
Email: {tangzj230, zxq6622}@163.com

Abstract—Encryption is an efficient way to protect the contents of digital media. Arnold transform is a significant technique of image encryption, but has weaknesses in security and applications to images of any size. To solve these problems, we propose an image encryption scheme using Arnold transform and random strategies. It is achieved by dividing the image into random overlapping square blocks, generating random iterative numbers and random encryption order, and scrambling pixels of each block using Arnold transform. Experimental results show that the proposed encryption scheme is robust and secure. It has no size limitation, indicating the application to any size images.

Index Terms—image encryption, image scrambling, Arnold transform, random division, image hashing

I. INTRODUCTION

Since digital media such as image, audio, and video are easy to copy, edit and transfer, the emergence of powerful tools raises a series of problems. For example, one can easily process the copyright images and redistribute them. Thus, the content protection becomes an important problem. In general, there are two ways. One is watermark; the other is encryption. The watermark-based techniques embed an invisible signal into the media to form a watermarked version. At the receiver’s end, the integrity of media contents can be verified by authenticating the embedded signal [1]. For encryption algorithms, they usually convert the meaningful media into the meaningless media. In this work, we focus on image encryption.

Image encryption, also called image scrambling, produces an unintelligible or disorder image from the original image. The existing image encryption algorithms can be classified into two kinds. One is spatial-based method; the other is frequency-based method. The spatial-based algorithms are usually achieved by swapping the pixel positions or altering pixel values. Arnold transform [2] is an efficient technique for position swapping, and widely applied to image encryption [3-4]. Zhu et al. [3] used Arnold transform and exclusive OR operation to produce scrambled images. Shang et al. [4] exploited Logistic map to improve the security of Arnold transform. Conventional Arnold transform based schemes [2-4] have a common weakness that image height must equal image width. Considering pixel value modification, we [5] proposed an image encryption scheme based on bit shuffling of individual pixels. It doesn’t need iterative computations, and then reduce the run time. A well-known image encryption algorithm based on frequency domain is designed by Ville et al. [6]. They exploited discrete prolate spheroidal sequences to design an image scrambling scheme without bandwidth expansion. However, the decrypted image isn’t totally equal to the original image. In other words, the algorithm is losty. For wavelet domain, Watanabe et al. [7] proposed a method for partial-scrambling of JPEG 2000 images using public-key encryption. It has backward compatibility with a standard JPEG 2000. This means that the encrypted images can be decoded by a standard JPEG 2000 decoder.

To overcome the weaknesses of Arnold transform, we exploit three random strategies, and then make the proposed scheme secure and suitable for images of any size. The rest of the paper is organized as follows. Section 2 will give a brief introduction of Arnold transform. Section 3 and Section 4 will present the proposed scheme and the experimental results. Conclusions are made in Section 5.

II. ARNOLD TRANSFORM

Arnold transform, also called cat map transform, is only suitable for encrypting \(N \times N\) images. It is defined as

\[
\begin{bmatrix}
    x' \\
    y'
\end{bmatrix} = \begin{bmatrix}
    1 & 1 \\
    1 & 2
\end{bmatrix}
\begin{bmatrix}
    x \\
    y
\end{bmatrix} \pmod{N}
\] (1)

where \((x, y)\) and \((x', y')\) are the pixel coordinates of the original image and the encrypted image, respectively. Let \(A\) denote the left matrix in the right part of equation (1). \(I(x, y)\) and \(I(x', y')^{(n)}\) represent pixels in the original image and the encrypted image obtained by performing Arnold transform \(n\) times, respectively. Thus, image encryption using \(n\) times Arnold transforms can be written as

\[
I(x', y')^{(k)} = A \cdot I(x, y)^{(k-1)} \pmod{N}
\] (2)

where \(k = 1, 2, \ldots, n\), and \(I(x', y')^{(0)} = I(x, y)\). Obviously, one can multiply the inverse matrix of \(A\) at each side of equation (2) to obtain \(I(x, y)^{(k-1)}\). In other words, the encrypted image can be decrypted by iteratively calculating the following formula \(n\) times.

\[
J(x, y)^{(k)} = A^{-1} \cdot J(x', y')^{(k-1)} \pmod{N}
\] (3)

where \(J(x', y')^{(0)}\) is a pixel of the encrypted image, and \(J(x, y)^{(0)}\) is a decrypted pixel by performing \(k\) iterations. Figure 1 show an example of image encryption using...
Arnold transforms, where (a) is the standard test image Airplane sized 512×512, and (b) is the encrypted image with \( n = 50 \).

Arnold transform has a property that the original image will appear when the equation (2) is iteratively calculated \( m \) times. The periodicity makes the encryption algorithm directly using Arnold transform unsecure. This is because one can easily obtain the original image by iterative computations once the encryption algorithm is known. The periodicity value \( m \leq N/2 \) and some specific values under different image sizes \( N \) are listed in Table 1 [8].

![Airplane and Encrypted Image](image)

Figure 1. Example of image encryption using Arnold transforms

From the above analysis, we find that Arnold transform has two weaknesses. One is the periodicity; the other is the requirement that image height must equal image width. The periodicity makes it unsecure, while the requirement limits its applications.

<table>
<thead>
<tr>
<th>( N )</th>
<th>60</th>
<th>100</th>
<th>120</th>
<th>128</th>
<th>256</th>
<th>480</th>
<th>512</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>60</td>
<td>150</td>
<td>60</td>
<td>96</td>
<td>192</td>
<td>240</td>
<td>384</td>
</tr>
</tbody>
</table>

### III. PROPOSED SCHEME

In this section, we design three random strategies to improve security and extend applications of Arnold transform. The first strategy is random division, the second is random iterative numbers, and the third is random encryption order. These operations are done by using a random generator controlled by separate keys.

![Block Diagram](image)

Figure 2. Block diagram of our encryption scheme

The proposed scheme is composed of four steps, i.e., random division, iterative number generation, encryption order generation and Arnold transform based encryption. Figure 2 presents the block diagram of our encryption scheme. The detailed steps are as follows.

1. **Random division.** We perform random division on the \( N×M \) image using different squares controlled by a key, and then obtain a series of square blocks. These squares must satisfy two conditions. One is that the union of them covers all pixels. The other is that there is overlapping region between adjacent squares. The first condition ensures that all pixels are scrambled. The second one provides us an opportunity to improve security achieved by using random encryption order. This will be explained in the next-to-last paragraph of this section. In practice, we use three arrays, i.e., \( X[K], Y[K] \), and \( S[K] \), to record these squares, where \( (X[i], Y[i]) \) denote the top-left coordinates of the \( i \)-th square (\( i = 1, 2, \ldots, K \)), \( S[i] \) is the size of the \( i \)-th square, and \( K \) is the square number. Note that the \( K \) value is related to the image size. In general, the bigger the image size, the bigger the \( K \) value. Figure 3 shows an example of random division with six squares.

![Random Division](image)

Figure 3. An example of random division

2. **Random iterative numbers.** Use a random generator controlled by a different key to produce \( K \) pseudo-random numbers, and record them in an array \( T \). Then \( T[i] \) represents the used iterative number of Arnold transform when it is applied to the \( i \)-th square block.

3. **Random encryption order.** To improve security, we produce a random order for processing these square blocks. This is achieved by the following. Use a random generator controlled by another key to produce \( K \) pseudo-random numbers, and then sort these \( K \) number to produce an ordered sequence. Record the original positions of these ordered numbers in an array \( A \). Thus, we process the \( A[i] \)-th square block at the \( i \)-th time. Take Figure 3 for example. A random order of the squares is 4, 3, 6, 1, 2, and 5.

4. **Encryption.** Apply Arnold transform to the \( A[i] \)-th square block with \( T[A[i]] \) iterations, and then replace those original pixels in the \( A[i] \)-th square block with the scrambled pixels. Repeat the computation from \( i = 1 \) to \( i = K \). The final result is the encrypted image. Note that since these blocks are overlapped, it is necessary to write the scrambled block to the original image before the next block is processed. This ensures that one can correctly decrypt a scrambled image.

The security of our encryption scheme mainly depends on the random strategies. Since there are many
possibilities of random division, attacker is very difficult to guess the division pattern. These strategies make our scheme secure. The approach of dividing an image into non-overlapping random rectangles is firstly proposed by Venkatesan et al. [9] for extracting secure image features. It is an efficient and secure technique and widely used in the design of image hashing [10]. For iterative numbers of Arnold transforms, keeping them different and hard to guess can improve the security of our scheme. Since adjacent squares have overlapping regions, different scrambling orders of square blocks will produce different encrypted results. This is the reason why the overlapping squares are needed in random division. For K squares, the number of random encryption order is equal to the number of permutations, i.e. K!. The bigger the K value, the bigger the number of random order. More number makes the guess of encryption order harder to succeed and then make the proposed scheme more secure. In practice, without the knowledge of all random strategies, it is impossible to correctly decrypt the encrypted image. Security experiments will be validated in the next section.

For our decryption scheme, it can be also composed of four steps. The first three steps are the same with those of the proposed encryption scheme. In the fourth step, we apply inverse Arnold transform, i.e., the equation (3), to the A[i]-th square block with T[A[i]] iterations, and then replace those pixels in the A[i]-th square block with the decrypted result. Note that the computation is repeated from i = K to i = 1. These are the main differences between the encryption scheme and decryption scheme. Clearly, the sender (Alice) and the receiver (Bob) in our schemes must share three key sequences, i.e., X[K], Y[K] and S[K], and two keys controlling the random generators for iterative numbers and encryption order.

IV. EXPERIMENTAL RESULTS

To validate the effectiveness, robustness, and security of the proposed encryption scheme, various experiments are done. Standard test images sized 512×512 such as Airplane, Baboon, Lena, and Peppers, are used. The square number K is 6. Figure 4 shows the original images and their encrypted images. In experiments, the used random division, iterative numbers, and encryption order for different image are the same, i.e., the keys are unchanged. From these encrypted results, we observe that image contents are unintelligible. This indicates the effectiveness of our scheme. To show the usefulness of random division, we apply our encryption scheme to Lena by changing the key for random division while keeping other keys unchanged. The results are listed in Figure 5, where (a), (c), and (e) are three different divisions, and (b), (d), and (f) are their encrypted images, respectively. Clearly, different divisions make different encrypted results.
We attack the encrypted image of Airplane shown in Figure 4 (b) by erasing blocks, scratching image, and adding Gaussian white noise with mean 0 and variance 0.01, and then decrypt these attacked images using our decryption scheme. The results are presented in Figure 6, where (a), (c), and (e) are the attacked images, (b), (d), and (f) are their corresponding decrypted images respectively. We find that the effects on the decrypted images caused by the attacks are randomly distributed. A typical example is Figure 6 (b). Although there are some blocks missing in the encrypted image shown in Figure 6 (a), no missing blocks are found in the decrypted image. This is because our encryption scheme inherits Arnold transform’s property that all pixels can be uniformly scrambled after iterative computations.

We apply our encryption scheme to Figure 7 (a) sized 480×360, and obtain its encrypted image, as shown in Figure 7 (b). The meaningless contents of Figure 7 (b) indicate a successful encryption. Since Arnold transform requires that image height is equal to image width, it can’t encrypt Figure 7 (a). No limitation of image size is a main advantage of our scheme. To validate the security of our scheme, we decrypt Figure 7 (b) by changing one key and keeping other keys unchanged. Figure 7 (c), (d), and (e) are the decrypted results, where (c), (d), and (e) use a wrong key for random division, iterative numbers, and encryption order, respectively. It is observed that all decryptions are wrong. This means that without the knowledge of all keys, one can’t obtain the original image from its encrypted version. Since Arnold transform has periodicity, one can easily reconstruct the original image by iterative computations. So our scheme is more secure than Arnold transform.
V. CONCLUSIONS

In this paper, we combine Arnold transform and three random strategies to design an image encryption scheme. The security of our scheme depends on the random strategies. Since there are many possibilities of random division, iterative numbers, and encryption order, attacker is difficult to correctly guess all random strategies at the same time. Thus, the security is guaranteed. Although the proposed scheme is based on Arnold transform, it has no size limitation, meaning that it can be applied to encrypt images of any size. This is because the random division can cover all pixels by using a series of squares. Therefore, compared with conventional Arnold transform, the proposed scheme is more secure and has more applications. In addition, by using Arnold transform to encrypt each square block, the proposed scheme has inherited the robustness of Arnold transform. This makes our encryption scheme robust against blocks missing, scratching and Gaussian white noise.

ACKNOWLEDGMENT

This work was partially supported by the Scientific Research Foundation of Guangxi Normal University for Doctors, the Natural Science Foundation of China (60963008, 60763011), the Natural Science Foundation of Guangxi (0832104, 0447035), the Project of the Education Administration of Guangxi (200911MS55, 200607MS135), the Scientific Research and Technological Development Program of Guangxi (10123005–8), the Scientific and Technological Research Projects of Chongqing’s Education Commission (KJ081309), and the Innovation Foundation of Guangxi Normal University for College Students (20100011). The authors would like to thank the anonymous referees for their valuable comments and suggestions.

REFERENCES


Zhenjun Tang received the B.S. degree and the M.Eng. degree from Guangxi Normal University, Guilin, P.R. China, in 2003 and 2006, respectively, and the Ph.D. degree from Shanghai University, Shanghai, China, in 2010. He is now with the Department of Computer Science, Guangxi Normal University. His research interests include image processing and multimedia security. He served as a session chair of 2010 International Conference on Computer and Computational Intelligence (ICCCI 2010). He is also a reviewer of IET Image Processing and Multimedia Tools and Applications.

Xianquan Zhang is currently a Professor with the Department of Computer Science, Guangxi Normal University, Guilin, P. R. China. His research interests include image processing and computer graphics.