Image encryption based on Kronecker inner product over finite fields and adversarial neural network

Shengliang Zou¹, Weixia Xia², Gailin Zhu², Yaru Liang³ and Jianhua Wu², ⁴

¹Gongqing College, Nanchang University, Jiujiang 332020, China
²School of Information Engineering, Nanchang University, Nanchang 330031, China
³College of Engineering, Jiangxi Agricultural University, Nanchang 330045, China
⁴Corresponding author’s e-mail: jhwu@ncu.edu.cn

Abstract. An image encryption algorithm based on Kronecker inner product matrix over the finite field and adversarial neural network (ANN) is designed. The Kronecker inner product matrix transform and the ANN fulfill the tasks of confusion and diffusion simultaneously. In addition, the look-up table method is used to complete the addition and multiplication operations over $\mathbb{F}_q^2$ finite fields, which can effectively improve the finite field computation speed while retaining its performance of non rounding errors. At the same time, relying on the secure hash function SHA-256 of the plain image to control the Logistic-Sine map greatly improves the diffusion and security of the encryption system. The simulation results verify that the proposed algorithm not only has high security and sufficient sensitivity, but also has a good resistance to various common attacks.

1. Introduction

In this era of information explosion, digital information is ubiquitous in our daily lives, making our lives more networked, informatized and digitized. Digital image information is widely used because of its strong intuitiveness and comprehensiveness. At the same time, digital images may be illegally stolen or tampered with by attackers during storage and transmission, which will greatly damage the interests of legal information holders. Therefore, it is crucial to develop a safe and reliable digital image encryption scheme.

At present, many signal processing methods have been applied to image encryption, such as chaotic map, Fourier transform, discrete cosine transform, Mellin transform and their fractional versions, etc, but the use of deep learning for image encryption is rarely reported.

In order to suppress the influence of quantization or round-off errors in the image encryption, many image encryption schemes have been proposed based on finite fields. Lima has studied many researches on finite field cosine transform (FFCT) for image encryption [1, 2]. In 2012, Lima studied the method of using the finite field cosine transform (FFCT) [1] to perform histogram uniformization on digital images, where the method includes dividing the image into blocks and applying FFCT to each block in a recursive manner. Later, Lima studied the fractional Fourier transform on the finite field (GFrFT) [2], and proposed a concept of graphical representations of elements in the finite field and an improved form of the finite field Fourier transform. Yang et al. [3, 4] extended the logistic map to finite fields, which solved the problem of increased computational complexity, the encryption scheme based on finite fields achieving a true lossless encryption without round-off errors. In 2020, Wu [5] proposed a new method of using permutation polynomials on finite fields, making full use of the inherent nonlinear
characteristics and real value of permutation polynomials. Later, Zhu successively proposed a new scheme that uses a Kronecker inner product matrix over a finite field combined with DNA operations [6] to improve the scheme, which can obtain a very good encryption effect.

At present, with the strong learning ability of deep learning, more and more scholars apply deep learning framework to image encryption. For example, Ding et al. [7] proposed an encryption and decryption network (namely DLEDNet) based on deep learning for medical images, where the network uses a cycle GAN network as a learning network to implement the encryption process. Wang et al. [8] introduced the perceptron of the neural network into the chaotic encryption system, and high-dimensional chaos are used to generate the chaotic sequence, which greatly extended the period of the chaotic system. Qin [9] proposed a deep learning-based face image encryption scheme, but the core idea of the algorithm is still the use of logical chaotic map for encryption of the pixel matrix, and does not really use neural networks for encryption operations. Similarly, in the image encryption schemes of [10] and [11], the neural network is only used to generate the masking sequence, and the encryption is realized by the XOR operation. However, the XOR operation is vulnerable to known plaintext attacks.

To overcome the vulnerability to chosen-plaintext attack and/or known-plaintext attack, a novel image encryption scheme is proposed in this work, which takes the advantages of both the Kronecker inner product matrix over a finite field and an adversarial neural network (ANN). The Kronecker inner product matrix transform and the ANN fulfil the tasks of confusion and diffusion simultaneously. In addition, the look-up table method is used to complete the addition and multiplication operations over GF(2^8) finite fields, which can effectively improve the finite field computation speed while retaining its performance of non-rounding errors. At the same time, relying on the secure hash function SHA-256 of the plain image to control the Logistic-Sine map greatly improves the diffusion and security of the encryption system. The experimental evaluation results verify that the proposed encryption scheme has a sufficient security performance.

The rest of this article is arranged as follows: the finite field, the Kronecker inner product and adversarial neural network are reviewed in Section 2. The detailed steps of the proposed image encryption scheme and ANN network model training are described in Section 3. Simulation results are given in Section 4. Finally, brief conclusions are drawn in Section 5.

2. Related background

2.1. Finite field

The number field is divided into finite field and infinite field according to the number of elements in the set. If a number field contains finite elements, it is called a finite field. $GF(q)$ or $F_q$ is used to denote a $q$-order finite field. A finite field is also called a Galois field.

In the area of cryptography, the finite field $GF(q)$ is a very important field. The number $q$ in $GF(q)$ is a prime one or an integer power of a prime. This is to ensure that all elements in the set (except 0) will have multiplication and addition inverse elements. For $GF(2^8)$. Due to the characteristics of the binary system, operations in a finite field can be very efficient on a computer. Especially $GF(2^8)$ is used most in image processing, because 8 is exactly the number of bits in a byte and the remainder of 256 ranges from 0 to 255, so it is very useful in image processing.

2.2. Kronecker inner product

The Kronecker inner product is a matrix block of selectable size obtained by operating two matrices of arbitrary sizes, so it can be used to encrypt images of different sizes. The definition is as follows.

If $A$ is a matrix of size $m \times n$, and $B$ is a matrix of size $p \times q$, the Kronecker inner product will get a matrix of size $mp \times nq$, denoted as $A \otimes B$, and defined as:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

(1)
2.3. Adversarial neural network
In a classic security scenario, as shown in Figure 1, three parties are involved: Alice, Bob, and Eve. Usually, Alice and Bob share a secret key $K$ and want to communicate securely, while Eve is an attacker who wants to eavesdrop on the communication between them, trying to obtain relevant information about $P$ through ciphertext $C$. In the ANN, Alice, Bob, and Eve are all neural networks. $P_{Bob}$ and $P_{Eve}$ are used to represent the results obtained by Bob and Eve after receiving the ciphertext $C$.

![Figure 1. A classic security scenario.](image)

In the framework of an ANN, the network structure of Alice and Bob are the same except that the input, output and internal parameters are different. The encryption process is performed by the Alice network, and the Bob network is used for decryption. At the same time, Eve cannot complete the decryption task without the key. The structure diagram of the Alice network is shown in Figure 2. The main structure is a fully connected layer plus four one-dimensional convolutional layers. All the activation functions of the convolutional layers are chosen as the Sigmoid function, except for the last convolutional layer that uses the Tanh activation function.

![Figure 2. The structure of the Alice or Bob network.](image)

3. Image encryption algorithm
3.1. ANC network model training
Before building an encryption system, one first needs to construct and train an ANN system. The training steps for an ANN are roughly as follows:

1. Building a complete ANN and configuration of the hyperparameters as follows: learning rate is 0.001, batchsize is 1024, and the optimizer selects the Adam's adaptive learning rate attenuation method.

2. Use of GPU to speed up the training of the network. When the reconstruction errors of Bob and Eve reach preset thresholds, the model training process stops and the best model is saved.

After constructing the ANN model according to Figures 1 and 2, we conduct the network training. The block length of the plaintext and ciphertext taken in the training results of this article are both 16 bits. The ideal result should be that Bob completely recovers the number of bits consistent with the length of the plaintext, that is, the number of errors is 0 bits, and the number of errors in Eve's recovery of the plaintext is 8 bits, that is, half of the bits are wrong. Why can't it be 16 bits? Because if all 16 bits are wrong, the attacker simply reverses the result to get the correct plaintext.

As can be seen from Figure 3, when the training is just started, the numbers of error bits between Alice and Bob and between Alice and Eve are both 8 bits. For a 16-bit binary data, the randomly generated 16-bit error number in probability is 8 bits. At this time, neither Bob nor Eve has the ability to decrypt, but with the mutual confrontation between neural networks, each neural network is constantly...
adjusting parameter values, and Alice and Bob gradually gain strong encryption and decryption capabilities. When iterated about 4000 times, Bob predicted that the error rate of the plaintext would drop sharply. After 7000 iterations, the number of error bits in the plaintext recovered by Bob was almost 0. However, Eve's error number was still fluctuating around 8. It shows that Eve's decryption ability is basically a guess, and the communication between Alice and Bob is secure.

3.2. Encryption and decryption scheme
The image encryption and decryption processes proposed in this article are shown in Figure 4 and Figure 5, respectively. The control parameter $\gamma$ of the chaotic map in the encryption algorithm, the parameter $\mu$ generated by the normalization of the hash value, and the $\alpha_i$ and $\beta_i$ ($i = 1, 2, \ldots, 8$) over the finite field are used as the key of the image encryption algorithm.

![Figure 3](image3.png)
**Figure 3.** Neural network training results.

![Figure 4](image4.png)
**Figure 4.** Encryption process of the proposed scheme.

![Figure 5](image5.png)
**Figure 5.** Decryption process of the proposed scheme.
Considering an original image $P$ of size $N \times N$, the encryption process is described as follows:

**Step 1:** The original image $P$ is used as the input of the secure hash function SHA-256 to obtain a hash value, which is expressed as a binary number with a length of 256 bits and normalized to the value $\mu$ in the range $(0,1)$.

**Step 2:** The plain image $P$ is partitioned into groups of two pixels, and each group is further partitioned into a 16-bit signal $Q(x) \in R^{16}$, $x = 1, 2, \cdots, N^2/2, R \in \{-1, 1\}$, as an input of Alice.

**Step 3:** A pseudorandom sequence $h$ (class of Byte) of length $N^2$ is generated by a Logistic-Sine map under the control of parameters $\gamma$ and $\mu$. The pseudorandom sequence $h$ is also partitioned into groups of two bytes, and each group is further partitioned into a 16-bit mask $H(x) \in R^{16}$, $x = 1, 2, \cdots, N^2/2, R \in \{-1, 1\}$, as another input into the Alice neural network.

**Step 4:** The Alice network is used to encrypt $Q(x)$ with $H(x)$ as the objective function in the training stage, resulting in the encryption result $F(x) \in R^{16}$, $x = 1, 2, \cdots, N^2/2, R \in \{-1, 1\}$. Then, perform image reorganization on the encryption result to obtain a noise-like intermediate image $L$ with a size of $N \times N$.

**Step 5:** Then the intermediate image $L$ is scrambled and diffused. The process is carried out over the finite field, and the process is as follows:

First, the look-up table method is used to construct addition and multiplication operations based on the $GF(2^8)$ finite field. The look-up table method can effectively improve the operation speed. The addition and multiplication operation rules are roughly visualized in Figures 6 and 7, respectively.

Secondly, construct two new Kronecker inner product matrices over the finite field. These two matrices are second-order orthogonal matrices. Matrices $A$ and $B$ are constructed as follows:

$$A_i = \begin{bmatrix} a_i & 1 + a_i \\ 1 + a_i & a_i \end{bmatrix}$$

(2)

$$B_i = \begin{bmatrix} b_i & 1 + b_i \\ 1 + b_i & b_i \end{bmatrix}$$

(3)

where $a_i, b_i \in F_2^8/\{0,1\}, i = 1, 2, \cdots, n$, the value of $n$ is dependent on the size of the original image. All these parameters are used as cipher-keys, and their multiplication order is 255.

Next, an orthogonal matrix of size $M \times N$ is generated through the Kronecker inner product matrix, in the same way, an orthogonal matrix of size $M \times N$ is generated as follows:

$$C_1 = A_1 \otimes A_2 \otimes A_3 \otimes \cdots \otimes A_n$$

$$C_2 = B_1 \otimes B_2 \otimes B_3 \otimes \cdots \otimes B_n$$

(4)

(5)

where $s_m = \log_2 M$, $s_n = \log_2 N$, $M$ and $N$ respectively represent the image height and width.

Subsequently, the orthogonal matrices $C_1$ and $C_2$ complete the scrambling and diffusion of the image at the same time, and are expressed as follows:

$$R = C_1 \times P \times C_2^T$$

(6)

Among them, the superscript $T$ in $C_2^T$ represents the transpose of the matrix.

Finally, the intermediate image $L$ is scrambled and diffused by the Kronecker inner product matrix transform over the finite field, the final encrypted image $C$ is obtained.

The decryption process is just the inverse of the encryption process and not reiterated here. In the above steps, the look-up table is used to complete the addition and multiplication operations over the $GF(2^8)$ finite field.
4. Experiments and analysis

4.1. Simulation results

During the simulation experiments, we selected multiple grayscale images for experimentation, but the final experimental results are not much different, so we only selected one of the images for display. The grayscale image ‘Baboon’ of size 256×256 is selected as test image. The original image and its encryption and decryption results are shown in Figure 8. It can be seen from Figure 8 that the decrypted image is basically the same as the plain image, and the histogram of the cipher image is almost uniform, so the attacker cannot perceive any useful information about the plain image.

![Figure 8](image_url)

**Figure 8.** Encryption results of image Baboon. (a) The original image with its histogram; (b) the encryption result with its histogram; (c) the decrypted image with its histogram.

4.2. Security analysis

A secure encryption algorithm can resist different security attacks. Here, we use several security analysis standards for evaluating our image encryption algorithm.

1) **Histogram analysis:** The histogram of an image is defined as: \( P(n_i), i = 0, 1, \ldots, L - 1 \), with \( P(n_i) \) being the number of pixels of pixel-value \( n_i \) in the image, and \( L \), the number of gray levels of the image. The histogram expresses the distribution characteristics of different gray-scale pixels in the image. Ideally, the pixel values of the encrypted image should be uniformly distributed.

2) **Correlation of adjacent pixels:** The correlation between adjacent pixels can reflect the extent of correlation between adjacent pixels of an image. Generally, the correlation between adjacent pixels of an image is much higher than other non-adjacent pixel pairs. In order to resist the attack of statistical analysis, the correlation between adjacent pixels of the encrypted results must be reduced greatly. For a secure encryption scheme, the encryption result should have low correlation values in all the vertical, horizontal, and diagonal directions. The formula for calculating the correlation coefficient \( r_{xy} \) of adjacent pixels in each direction is:
\[ E(x) = \frac{1}{n} \sum_{i=1}^{n} x_i \]  
\[ D(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i - E(x))^2 \]  
\[ Cov(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - E(x))(y_i - E(y)) \]  
\[ r_{xy} = \frac{Cov(x, y)}{\sqrt{D(x)D(y)}} \]  

where \( x \) and \( y \) are the gray values of two adjacent pixels, \( E(x) \) and \( E(y) \) is the expectation, \( D(x) \) is the variance of \( x \), and \( Cov(x, y) \) is the covariance, \( n \) represents the number of selected pixel pairs.

In order to illustrate this feature graphically, the corresponding joint distribution map of adjacent pixel pairs is drawn in Figure 9, and Table 1 summarizes the correlation coefficient values in these three directions. From the results in Figure 9 and Table 1, it can be seen that the correlation between adjacent pixel pairs of the cipher image is greatly reduced. Therefore, it shows that the algorithm proposed in this paper can effectively resist correlation analysis attacks.

![Figure 9](image-url)  
Figure 9. The joint distribution of adjacent pixels in the original image and the encryption result in three directions: (a) horizontal; (b) vertical; and (c) diagonal directions.

Table 1. Comparison of correlation coefficient between adjacent pixels.

| Algorithm          | Image            | Horizontal | Vertical | Diagonal |
|--------------------|------------------|------------|----------|----------|
| Plaintext Baboon   | 0.6494           | 0.7470     | 0.6642   |
| Ref. [12] Encrypted Baboon | −0.0001         | 0.0024     | 0.0040   |
| Ref. [13] Encrypted Baboon | 0.0110          | 0.0019     | 0.0118   |
| Proposed algorithm | Encrypted Baboon | −0.0064    | −0.0183  | −0.0119  |

3) Differential attack analysis: The relationship between the plaintext and the ciphertext can be measured by the difference caused by a specific change in the plain image. The diffusion characteristics can be evaluated by two indicators: The Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI), defined respectively as

\[ NPCR = \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} D(i,j)}{W \times H} \times 100\% \]  

(11)
\[ UACI = \frac{1}{W \times H} \left[ \sum_{i=1}^{W-1} \sum_{j=1}^{H-1} \left| C_1(i,j) - C_2(i,j) \right| \right] \times 100\% \]  
\[ D(i,j) = \begin{cases} 0, & \text{if } C_1(i,j) = C_2(i,j) \\ 1, & \text{if } C_1(i,j) \neq C_2(i,j) \end{cases} \]  

where \( C_1 \) and \( C_2 \) are the pixel values of the two encrypted images, \( W \times H \) is the size of the images.

For two random images, the expected values of NPCR and UACI for 8-bit grayscale images are: NPCR=99.6094\% and UACI=33.4635\% [14]. The NPCR and UACI test results obtained in this experiment were 99.6338\% and 33.5452\%, respectively. They are very near to the theoretically ideal values. This shows that the algorithm has good diffusion characteristics and can resist differential attack analysis.

4) Information entropy: Generally, for an image with 256 gray levels, when pixel gray-levels are uniformly distributed, the ideal value of its information entropy is 8 bits. The information entropy of the plain image and the cipher image generated by this algorithm is shown in Table 2, from which it can be seen that the information entropy of all cipher images is very close to the ideal value of 8 bits and the average information entropy of the ciphertext is 7.9973. This means that the proposed encryption system is safe and will not be subject to entropy attacks.

| Image   | Baboon | Barbara | Boat   | Lake   | Peppers |
|---------|--------|---------|--------|--------|---------|
| Plaintext | 7.1081 | 7.3893 | 7.0932 | 7.1537 | 7.2041 |
| Ciphertext | 7.9972 | 7.9972 | 7.9973 | 7.9976 | 7.9974 |

5) Key space and sensitivity: Key sensitivity includes not only encryption key sensitivity, but also decryption key sensitivity. Only when the key space is large enough can the performance against brute force attacks be better. The key of the algorithm is the parameters \( \alpha_i \) and \( \beta_i \) (\( i = 1, 2, \ldots, 8 \)) of the finite field, the control parameter \( \gamma \) of the chaotic map, and the parameter \( \mu \) generated by the normalization of the hash value. For each group of keys \( \alpha_i \) and \( \beta_i \) of the finite field, its key space is \( (2^8 - 2)^8 \). For keys \( \gamma \) and \( \mu \), its key space is \( 10^{14} \) and \( 2^{256} \), respectively. Therefore, the key space of this encryption scheme is at least \( 2^{237} \). Obviously, this is a sufficiently large key space to resist brute force attacks.

![Figure 10](image-url)  

**Figure 10.** A decrypted image obtained by changing one encryption key or one decryption key among the 18 keys: (a) change only one encryption key; (b) change only one decryption key.

In order to verify the key-sensitivity of the proposed scheme, this experiment takes the image ‘Baboon’ as an example to conduct a set of tests. This scheme has a total of 18 keys. Only one key is changed each time, and the other keys remain unchanged to decrypt the image. A key-sensitivity test example of decrypted image obtained is shown in Figure 10 by changing only one encryption key or only one decryption key among the 18 keys. From the test results, it is impossible to visually recognize any valid information about the plain image. That is, when one of the key values changes within a small range, the plain image cannot be recovered. Therefore, the image encryption scheme is secure in terms of key-sensitivity.
5. Conclusion
This paper proposes an image encryption scheme based on an adversarial neural network and Kronecker inner matrix product over the finite field. In this scheme, the secure hash function SHA-256 is used to highly correlate the plain image with the key, and the hash value generated by SHA-256 is normalized to control the Logistic-Sine chaotic map, and then the plain image is mapped into the finite field, and the pixels are scrambled and diffused based on the Kronecker inner product matrix over the finite field. Finally, the ANN is used for further scrambling. Due to the nonlinear characteristic of the neural networks, the proposed encryption system is highly nonlinear. Using the look-up table method based on the finite field effectively improves the calculation speed while retaining the non-rounding-error performance. The simulation results and security evaluation show that the algorithm can encrypt different types of images with a high security.

Acknowledgments
The work of this manuscript was partly financially supported by the National Natural Science Foundation of China [Grant numbers 62041106, 61662047], the Science and Technology Program of Jiangxi Provincial Department of Education [Grant number GJJ190203].

References
[1] Lima J B, Souza R M C D 2012 Histogram uniformization for digital image encryption. 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images 55–62
[2] Lima J B, Souza R M C D 2012 The fractional Fourier transform over finite fields. Signal Processing 92(2) 465–476
[3] Yang B, Liao X F 2018 A new color image encryption scheme based on logistic map over the finite field ZN. Multimedia Tools and Applications 77(16) 21803–21821
[4] Yang B, Liao X F 2018 Some properties of the Logistic map over the finite field and its application. Signal processing 153(12) 231–242
[5] Wu J H, Liu H, Zhu X S 2020 Image encryption based on permutation polynomials over finite fields. Optica Applicata 50(3) 357–376
[6] Zhu X S, Liu H, Liang Y R, et al. 2020 Image encryption based on Kronecker product over finite fields and DNA operation. Optik - International Journal for Light and Electron Optics 164725
[7] Ding Y, Wu G Z, Chen D J, et al. 2021 DeepEDN: A deep learning-based image encryption and decryption network for internet of medical things. IEEE Internet of Things Journal 8(3) 1504–1518
[8] Wang X Y, Yang L, Liu R, et al. 2010 A chaotic image encryption algorithm based on perceptron model. Nonlinear Dynamics 62(3) 615–621
[9] Qin Y, Zhang C, Liang R, et al. 2019 Research on face image encryption based on deep learning. IOP Conference Series: Earth and Environmental Science 252(5) 052007
[10] Li X L, Jiang Y R, Chen M R, et al. 2018 Research on iris image encryption based on deep learning. EURASIP Journal on Image and Video Processing 2018(1) 126–135
[11] Yu W W, Cao J D 2006 Cryptography based on delayed chaotic neural networks. Physics Letters A 356(4–5) 333–338
[12] Wu J H, Zhang M X, Zhou N R 2017 Image encryption scheme based on random fractional discrete cosine transform and dependent scrambling and diffusion. Journal of Modern optics 64(4) 334–346
[13] Askar S, Karawia A, Al-Khedhairi A, et al. 2019 An algorithm of image encryption using logistic and two-dimensional chaotic economic maps. Entropy 21(1) 44
[14] Hu T, Liu Y, Gong L H, et al. 2017 Chaotic image cryptosystem using DNA deletion and DNA insertion. Signal Processing 134(5) 234–243