Article

Coupling quantum random walks with long and short term memory for high pixel image encryption schemes

Junqing Liang [1], Zhaoyang Song [1],Zhongwei Sun[1], Mou Lv[2] and Hongyang Ma [3,*]

[1] School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266033, China;
[2] School of Environmental and Municipal Engineerin, Qingdao University of Technology, Qingdao 266033, China;
[3] School of Science, Qingdao University of Technology, Qingdao 266033, China;
* Hongyang Ma: hongyang_ma@aliyun.com;

Abstract: This paper proposes an encryption scheme for high pixel density images. Based on the application of the quantum random walk algorithm, the Long short-term memory (LSTM) can effectively solve the problem of low efficiency of the quantum random walk algorithm in generating large-scale pseudorandom matrices, and further improve the statistical properties of the pseudorandom matrices required for encryption. The LSTM is then divided into columns and fed into the LSTM in order for training. Due to the randomness of the input matrix, the LSTM cannot be trained effectively, so the output matrix is predicted to be highly random. The LSTM prediction matrix of the same size as the key matrix is generated based on the pixel density of the image to be encrypted, which can effectively complete the encryption of the image. In the statistical performance test, the proposed encryption scheme achieves an average information entropy of 7.9992, an average number of pixels changed rate (NPCR) of 99.6231%, an average uniform average change intensity (UACI) of 33.6029% and an average correlation of 0.0032. Finally, various noise simulation tests are also conducted to verify its robustness in real-world applications where common noise and attack interference are encountered.

Keywords: image encryption; high pixel density; neural networks; quantum random walk

1. Introduction

With the rapid development of Internet technology, more and more high-value data and information is being transmitted over the Internet, and therefore the security of data transmission is becoming more and more important. While ordinary data can be hidden and protected by classical encryption schemes such as DES [1]and AES[2], the information contained in an RGB image is represented by the pixel values. Because of the strong correlation between the neighbouring pixel values of RGB images and the amount of information stored in images, classical encryption schemes are often unable to achieve good encryption of image information, so the encryption of image information is separated from classical data encryption and becomes a separate research direction, focusing on image specific encryption schemes from the data information characteristics of images[4–9]. One very promising direction is the application of neural networks to image encryption. This is because cryptography places particular emphasis on the introduction of non-linear transformations, which is a distinctive feature of neural networks, and in addition to this, neural networks have characteristics such as ultra-fast parallel processing and operate in matrix form, all of which are extremely well suited to the field of image encryption, making neural networks increasingly interesting in the field of image encryption[3,10,11].

The LSTM [12]is a special type of recurrent neural network (RNN)[13] that uses the 'inner loop' of a neural network to preserve the contextual information of a time series, allowing the use of past signal data to infer an understanding of the current signal. Theoretically, RNN can retain information from any moment in time. However, in practice, the transfer of information tends to decay over long time intervals, and the effectiveness of the information is greatly reduced after a certain period of time. As a result, RNN is not well equipped to deal with the problem of long-term information dependence, resulting in a tendency to rely only on the most recent input information for inference. To overcome this problem, LSTM is proposed to solve the long-term dependency problem. In contrast to RNN, remembering the content of earlier moments is its default behaviour. Therefore it does not require a significant cost specifically and works better.

Quantum computing is a new computing mode that follows the laws of quantum mechanics to regulate quantum information units for computing[14]. Quantum algorithm [15–18]is an algorithm based on quantum computation. By using the unique behavior of quantum mechanics, such as superposition, entanglement and interference, some algorithms have achieved exponential acceleration compared with classical algorithms[17,19]. Quantum random walk(QW) is a quantum algorithm, which was first proposed by Aharonov et al[20]., including continuous time QW[21] and discrete time QW[22]. Compared with the classical random walk, the algorithm has a significant improvement in computational efficiency, and its time complexity is increased from O(n^2) to O(n). On the basis of one-dimensional QW, Baryshnikov et al. studied the difference between two-dimensional and one-dimensional coordinate space, and expounded the advantages and unique properties of two-dimensional QW[23]. Although QW is a quantum algorithm, its probability matrix can be solved by classical computers, and the algorithm complexity is still O(n), which makes QW can be applied in classical computers in advance.

Both LSTM and QW have applications in image encryption. He et al.[24] proposed an OF-LSTMS that replaces the matrix operation in LSTM with a xor operation to obtain an encrypted image after a single forward propagation. yang et al.[25] studied the properties of one-dimensional QW and applied it to quantum image encryption for the first time. Abd et al.[26] analyzed the statistical properties of the probability distribution matrix of two-dimensional quantum walks and applied it to image encryption; Ma et al.[27] combined the discrete cosine transform (DCT)[28] and the probability matrix of alternating quantum walks (AQW) for image encryption, etc.

Although QW probability matrices have been widely used in the field of image encryption, they still have shortcomings and are too inefficient when dealing with high pixel images. The time complexity of the one-dimensional AQW probability matrix is O(n), and the computational complexity of the AQW probability matrix is O(n^2), which is still polynomial in time complexity, but the time consumed to generate the QW probability matrix is unacceptable in practical applications to encrypt high pixel value images. At the same time, we also found that the statistical properties required for the encryption of the QW probability matrix are not satisfactory, so when QW are used for encryption, other algorithms are often used to improve the encryption, e.g. Ma used a discrete cosine transform algorithm to perform further dislocation encryption in the DCT domain after applying QW to confuse the pixel values. This does not increase the encryption efficiency too much, but the use of separate algorithms for the scrambling and obfuscation phases nullifies the advantage of having an infinite key matrix for the QW, as it can only participate in one of the scrambling and obfuscation phases, and the two phases are independent of each other.

In order to optimise the statistical properties of the QW probability matrix and its performance on high pixel precision image encryption for better encryption, we propose an image encryption scheme that combines neural networks with quantum algorithms. By combining the QW with the LSTM, the initial matrix is generated using the QW probability matrix, and after training through the LSTM, a suitable prediction matrix is output as the key matrix for encryption according to the required pixel accuracy of the image to be encrypted. We show that this combination can improve the efficiency of the key matrix generation, and at the same time, because the QW probability matrix has strong randomness, the LSTM can not effectively find its pattern to predict, so the generated prediction matrix is also disordered, and has better statistical properties than the QW probability matrix for encryption, which can be better used as the key matrix to complete the It can be used as a key matrix for encryption.

Section 2 of this paper presents the basics related to encryption schemes, including the study and analysis of LSTM and AQW. Section 3 presents specific encryption schemes. Section 4 presents the simulation and theoretical analysis of this paper for detecting the effectiveness of the encryption scheme and lists the comparison of similar schemes to the encryption scheme proposed in this paper. Section 5 concludes the work in this paper and also provides an outlook on the subsequent work. The most critical module of the LSTM is the cell state, which is represented by C_t, the current state at the current moment, and is generated by the state C_{t-1} at the previous moment together with the signal input x_t at the current moment, while C_t will continue to be passed to the next moment together with x_{t+1} to generate C_{t+1}.

2. Related work and background knowledge

2.1. LSTM

2.1.1. subsubsection Cell state

The LSTM functions mainly through a gate structure, which contains three main gate structures: forgetting gates, memory gates and output gates. Gates are a way of allowing information to pass selectively. In the LSTM, these 'gates' are used to introduce or remove information from the cell state C_t. Some of the gates are somewhat similar to filters in signal processing, allowing signals to pass partially or being processed by gates as they pass; others are also similar to logic gates in digital circuits, allowing signals to pass or not pass. These three gates are used to control the retention and transfer of information in the LSTM, which is ultimately reflected in the cell state C_t and the output signal h_t.

(1) Forget gate: The function is to selectively forget part of the information, i.e. to decide which information from C_{t-1} will be passed to C_t. The forgetting gate consists of a sigmoid neural network layer σ and a per-bit multiplication operation. The output signal h_{t-1} from C_{t-1} is received together with the input signal x_t from C_t, and after σ :

$$\sigma(x) = \frac{1}{1+e^x} \tag{1}$$

The sigmoid neural network layer function is similar to the sigmod function representation in the form of $\sigma(x) = 0$, which means that $[\mathrm{h}_{t-1}, \mathrm{X}_t]$ is not allowed to pass, and $\sigma(x) = 1$, which means that $[\mathrm{h}_{t-1}, \mathrm{X}_t]$ is allowed to pass all, after dimensionality reduction and calculation to obtain f_t :

$$f_t = \sigma\left(\mathrm{W}_f \cdot [\mathrm{h}_{t-1}, \mathrm{X}_t] + \mathrm{b}_f\right) \tag{2}$$

where W_f the weight parameter of the forgetting gate and $\mid b_f$ denotes the bias of the forgetting gate.

(2) Memory gate: The function of the memory gate is the opposite of the forgetting gate, i.e. it determines which of the input information x_t and h_{t-1} will be retained. The gate consists of two neural network layers, a sigmoid neural network layer σ, which is identical to the forgetting gate, and a tanh neural network layer tanh.

The role of the Sigmoid neural network layer is the same as its role in the forgetting gate, which receives xt and ht-1 as inputs and then outputs a value it between 0 and 1 to control the extent to which the information is updated:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

The role of the Tanh neural network layer is to couple the input x_t and h_{t-1} and then create a completely new state $\tilde{C}_t$:

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

where W_c is the weight matrix of the memory gate and b_c is the bias of the memory gate.

Calculation of i_t and $\tilde{C}_t$, combined with the f_t obtained from the oblivion gate, for cell state update:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

(3) Output gate: The function of the output gates is to couple the cell state C_{t-1}, which has been selected and processed by the forgetting and memory gates, with x_t as the output signal at the current moment. The coupling process consists of two stages: first a sigmoid neural network layer with an input value ot between 0 and 1;C_t passes through a tanh function to obtain a value between -1 and 1, which is multiplied by ot to become the output signal ht,ht which is passed on to the next stage as the input signal for the next moment.

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{6}$$

where W_o is the weight matrix of the output gate and b_o is the bias of the output gate.

The chain structure diagram of the LSTM is shown in Figure 1, which illustrates the chain relationship between the three adjacent substructures and the composition of each LSTM substructure.

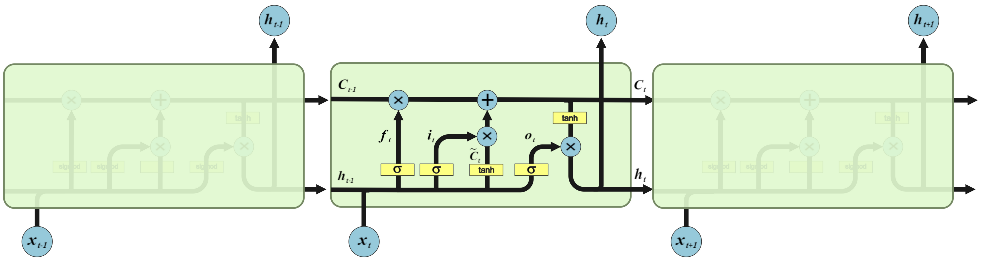

Figure 1. Chain model for LSTM.

2.2. *Quantum random walk*

This paper is based on the theory of discrete-time QW. The discrete-time QW consists of four main elements: the walker, the coins carried by the walker, the coin toss and the walk rule.

The Hilbert space $\hat{H}$ of a one-dimensional discrete-time QW tensor consists of the walker position space H_w and the coin space H_Γ :

$$\hat{H} = H_w \otimes H_\Gamma \tag{7}$$

In a QW, each step of the walk is determined by a unique coin flip operator Γ :

$$\Gamma = \begin{pmatrix} \cos\beta & \sin\beta \\ \sin\beta & -\cos\beta \end{pmatrix} \tag{8}$$

After the coin toss is completed, the movement of the walker is specified by the conditional displacement operator S_i :

$$S_i|\hat{x}\rangle = \left|\hat{x} + (-1)^{\Gamma}\right|, \Gamma \in 0, 1 \tag{9}$$

The $|\hat{x}\rangle(\hat{x} \in Z)$ in the above equation forms the base vector of the walker's position space; the two base vectors $|0\rangle, |1\rangle$ form the coin space. We specify: when the coin state is $|0\rangle$, the walker is manipulated to move one unit in the forward direction; when the coin state is $|1\rangle$, the walker is manipulated to move one unit in the reverse direction.

In the AQW used in this paper, the walker controlled by the coin operator alternates between two arbitrarily chosen vertical directions $\tilde{x}$ and $\tilde{y}$, and the walking operator $\hat{U}$ for the whole QW process can be described as:

$$\hat{U} = \hat{S}_{\tilde{y}}(I \otimes H_\Gamma)\hat{S}_{\tilde{x}}(I \otimes H_\Gamma) \tag{10}$$

where $\hat{S}_{\tilde{y}}, \hat{S}_{\dot{x}}$ are the displacement operators of the walker at each point on the $\tilde{x}$ and $\tilde{y}$ axes:

$$\begin{aligned}
\hat{S}_{\tilde{y}} &= \sum_{\tilde{x},\tilde{y}}^{N}(|\tilde{x},(\tilde{y}+1) \bmod \varpi, 0\rangle\langle\tilde{x},\tilde{y},0|) \\
&\quad + \sum_{\tilde{x},\tilde{y}}^{N}(|\tilde{x},(\tilde{y}-1) \bmod \varpi, 1\rangle\langle\tilde{x},\tilde{y},1|) \\
\hat{S}_{\dot{x}} &= \sum_{\tilde{x},y}^{N}(|(\tilde{x}+1) \bmod \varpi, \tilde{y}, 0\rangle\langle\tilde{x},\tilde{y},0|) \\
&\quad + \sum_{\tilde{x},\tilde{y}}^{N}(|(\tilde{x}-1) \bmod \varpi, \tilde{y}, 1\rangle\langle\tilde{x},\tilde{y},1|)
\end{aligned} \tag{11}$$

where, ϖ indicates the prescribed walking boundary.

Suppose the initial moment:The walker's location is $\left(0_{\tilde{x}}, 0_{\tilde{y}}\right)$ and the coin is in the superposition state $H_\Gamma = \cos\alpha|0\rangle + \sin\alpha|1\rangle$, then the initial moment system state is :

$$|\psi_0\rangle = |\varphi_0\rangle_w \otimes (\cos\alpha|0\rangle + \sin\alpha|1\rangle)_\Gamma \tag{12}$$

The system state after a T walk can be expressed as:

$$|\psi_T\rangle = \hat{U}^T|\psi_0\rangle \tag{13}$$

3. Algorithm description

3.1. *The encryption process*

3.1.1. Preparation of quantum random walk probability distribution matrix

The data of the corresponding element in the matrix is the probability $P(\delta, \vartheta, T)$ of the walker appearing at the coordinates $\left(\delta_x, \vartheta_y\right)$ of the location, as can be deduced from the above:

$$P(\delta,\vartheta,T) = \left|\left\langle\delta,\vartheta,0\middle|\hat{U}^T\middle|\psi_0\right\rangle\right|^2 + \left|\left\langle\delta,\vartheta,1\middle|\hat{U}^T\middle|\psi_0\right\rangle\right|^2 \left(\delta_x, \vartheta_y\right) \tag{14}$$

The resulting probability distribution matrix M and its four sub-matrices M_1, M_2, M_3, M_4 after equiproportional partitioning:

$$\begin{gathered}
\boldsymbol{M} = \begin{pmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{11} & \cdots & P_{nn} \end{pmatrix} \\
\boldsymbol{M}_1 = \begin{pmatrix} P_{11} & \cdots & P_{1\frac{n}{2}} \\ \vdots & \ddots & \vdots \\ P_{\frac{n}{2}} & \cdots & P_{\frac{n}{2}\frac{1}{2}} \end{pmatrix} \boldsymbol{M}_2 = \begin{pmatrix} P_{1\frac{n}{2}} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{\frac{n}{2}\frac{n}{2}} & \cdots & P_{\frac{n}{2}n} \end{pmatrix} \\
\boldsymbol{M}_3 = \begin{pmatrix} P_{\frac{n}{2}1} & \cdots & P_{\frac{n}{2}\frac{n}{2}} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{n\frac{n}{2}} \end{pmatrix} \boldsymbol{M}_4 = \begin{pmatrix} P_{\frac{n}{2}} & \cdots & P_{\frac{n}{2}\frac{n}{2}} \\ \vdots & \ddots & \vdots \\ P_{\frac{n}{2}n} & \cdots & p_{nn} \end{pmatrix}
\end{gathered} \tag{15}$$

We set the walker to be at the centre of the Hilbert space $\hat{H}$ tensed by H_w and H_c, so the four submatrices $\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3, \boldsymbol{M}_4$ are centrosymmetric about the point $P_{\frac{n}{2}}$ in the final generation. To prevent the LSTM from learning the rule such that the statistical performance of the final generated key matrix is degraded, in this paper only $\hat{\boldsymbol{M}} = \boldsymbol{M}_1$ is chosen as the required initial pseudo-random number matrix to participate in the encryption.

3.1.2. Preparing the encryption key matrix

Step1:Ensure the reproducibility of the LSTM across devices. (i) Fix the random seeds of each dependency library so that each function is called with the same initial value and random value each time it is trained by the LSTM. (ii) Presetting the dropout function in the LSTM to 0, i.e. not dropping any nodes of the neural network, to ensure that the network model is fixed each time. (iii) Fixed platforms as well as devices, taking the current mainstream pytroch framework as an example, which still cannot guarantee the accuracy of model reproduction under different CPU and GPU pairings, and also requires CUDA environment variable configuration etc. in order to further reduce uncertainty.

Step2:Generate the LSTM input vector. Divide $\hat{\mathrm{M}}$ by column:

$$\begin{pmatrix} P_{11} & \cdots & P_{1\frac{n}{2}} \\ \vdots & \ddots & \vdots \\ P_{\frac{n}{2}1} & \cdots & P_{\frac{n}{2}\frac{1}{2}} \end{pmatrix} \rightarrow \left(\varphi_1, \varphi_2, \ldots \varphi_{\frac{n}{2}-1}, \varphi_{\frac{n}{2}}\right) \tag{16}$$

$\hat{\boldsymbol{M}}'$ is obtained by Min-Max normalization of $\hat{\boldsymbol{M}}$:

$$\left(\varphi_1, \varphi_2, \ldots \varphi_{\frac{n}{2}-1}, \varphi_{\frac{n}{2}}\right) \longrightarrow \left(\xi_1, \xi_2, \ldots \xi_j \ldots \xi_{\frac{n}{2}}\right) \tag{17}$$

ξ_i is the vector to be input.

Step3: Generate the key matrix required for encryption. Input the vectors ξ_{i} in matrix $\hat{\boldsymbol{M}}''$ into the LSTM in order for training, and set the LSTM prediction quantity as γ^2 to get the prediction matrix $\hat{\boldsymbol{M}}'''$:

$$\hat{\boldsymbol{M}}''' = \begin{pmatrix} \varpi_{11} & \cdots & \varpi_{1\gamma} \\ \vdots & \ddots & \vdots \\ \varpi_{\gamma 1} & \cdots & \varpi_{\gamma\gamma} \end{pmatrix} \tag{18}$$

Inverse normalization of $\hat{\boldsymbol{M}}'''$ yields $\boldsymbol{M}_E$:

$$\begin{pmatrix} \varpi_{11} & \cdots & \varpi_{1\gamma} \\ \vdots & \ddots & \vdots \\ \varpi_{\gamma 1} & \cdots & \varpi_{\gamma\gamma} \end{pmatrix} \longrightarrow \begin{pmatrix} \partial_{11} & \cdots & \partial_{1\gamma} \\ \vdots & \ddots & \vdots \\ \partial_{\gamma 1} & \cdots & \partial_{\gamma\gamma} \end{pmatrix} \tag{19}$$

In Figure 2, we show the comparison between the predicted data and the expected values formed from the accurate data after training the QW probability matrix as an LSTM training matrix. Where subplot a shows the trend in randomness between predicted and expected values; subplot b shows the distribution between specific predicted and expected values.

3.1.3. Image encryption

The R, G and B channels in our proposed encryption scheme are performed separately, and our encryption algorithm is described in terms of $\gamma \times \gamma$ pixels of RGB image I corresponding to a grey-scale map in the form of matrix $\boldsymbol{M}_I$.

Step1:Hide the pixel information in $\boldsymbol{M}_I$ by obfuscating the pixel values. Here, we borrow the heteroskedastic algorithm to implement the obfuscation operation:

$$\boldsymbol{M}_I' = \boldsymbol{M}_I \oplus \boldsymbol{M}_E \tag{20}$$

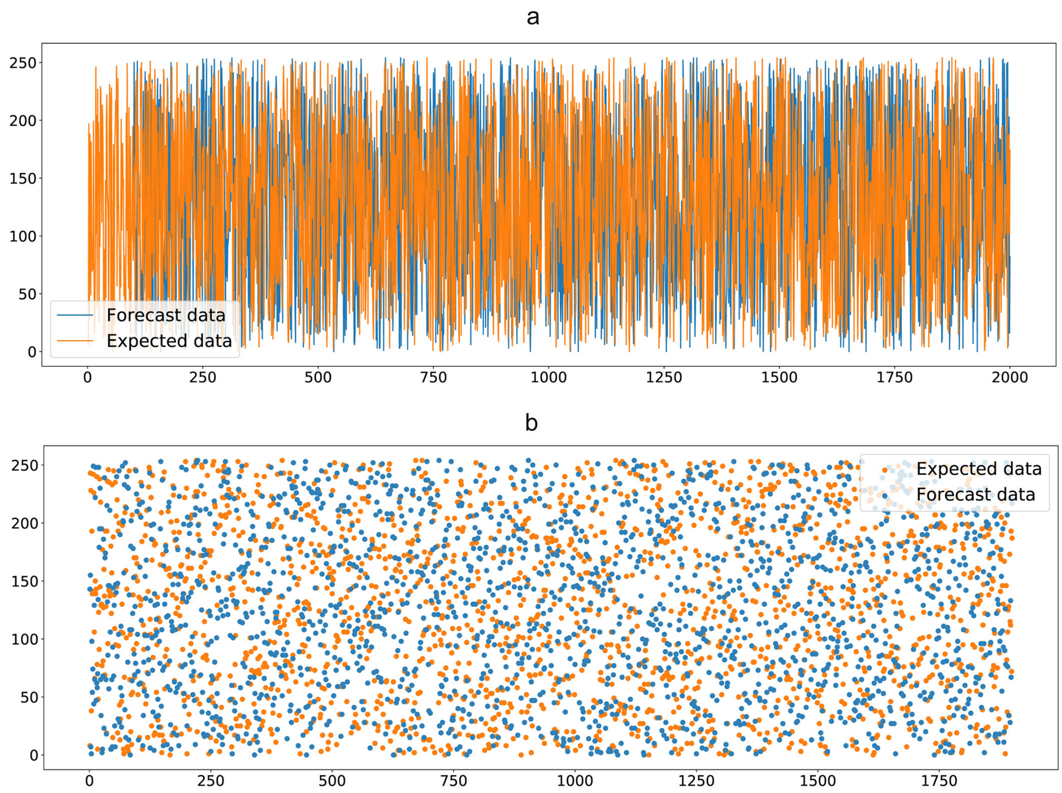

Figure 2. LSTM generation key matrix

Step2: Generate matrix $\boldsymbol{M}_E' = \boldsymbol{M}_E$, sort the index value matrix Ω of $\boldsymbol{M}_E'$ in order to get Ω', reorder the $\boldsymbol{M}_I'$ after the confusion operation according to the corresponding position in Ω', and achieve the dislocation of the image by destroying the relationship between adjacent pixel values to get $\boldsymbol{M}_I''$. The schematic diagram of the dislocation algorithm is shown in Figure 3. 162 163 164 165 166

3.2. *The decryption process* 167

3.2.1. Preparing the decryption key matrix 168

Since the QW algorithm generates pseudo-random numbers, i.e., the sequence of pseudorandom numbers generated any two times is the same if the start parameters α, β, N are unchanged, the new probability distribution matrix generated $\boldsymbol{M}' = \boldsymbol{M}$. Since we have removed the uncertainty and randomness from the LSTM, the $\boldsymbol{M}'$ is processed once according to the encryption process for $\boldsymbol{M}$, and finally The prediction matrix generated by the LSTM is processed to obtain $\boldsymbol{M}_D = \boldsymbol{M}_E$. 169 170 171 172 173 174

3.2.2. Decryption of encrypted image 175

Step1:The encrypted image $\boldsymbol{M}_I''$ is obtained using the inverse permutation $\boldsymbol{M}_I'$. This process is the inverse of the permutation operation and the algorithm is shown in Figure 3: 176 177

Step2: $\boldsymbol{M}_I{}'$ for obfuscation reduction to get $\boldsymbol{M}_I$. 178

3.3. *Encryption and decryption algorithm flow chart* 179

We show the key steps of our proposed image encryption scheme by means of a flowchart, including the generation of the QW probability density matrix, the process of generating the key matrix by LSTM, and the two key steps (scrambling, confusion) of the image encryption and decryption process using the key matrix, as shown in Figure 4. 180 181 182 183

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 December 2022 doi:10.20944/preprints202212.0270.v1

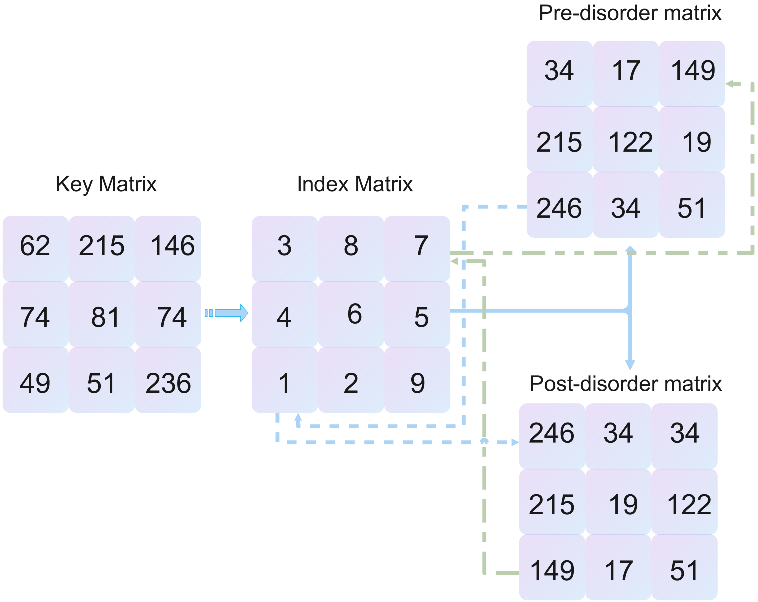

Figure 3. Encryption scheme - scrambling algorithm

4. Simulation and analysis

To verify the resistance of the proposed scheme, three RGB images with a pixel size of 2000×2000 were encrypted and decrypted according to the proposed encryption scheme, and various statistical analyses were carried out on the encrypted images and the keys used, including histogram analysis, correlation analysis and information entropy analysis for the encrypted images; sensitivity analysis and key space analysis for the key matrix, etc.

4.1. Experimental parameters and encryption and decryption results

We use $\varpi = 240, \alpha = \frac{\pi}{23}, \beta = \frac{\pi}{41}$ as the start parameters of the QW to prepare a QW probability matrix of size 100×2000, and set the prediction length of the LSTM to 2000 , i.e. to generate a key matrix of the same size as the RGB image to be encrypted. The encryption and decryption results are shown in Figure 5.

4.2. The statistical analysis

4.2.1. Correlation analysis

Adjacent pixel correlation R_{AB} is used to measure the degree of correlation of adjacent pixel values. Adjacent pixel values in RGB images often have strong correlations in horizontal, vertical and diagonal directions. Image encryption algorithms will destroy this correlation, and the degree of destruction can reflect the effect of encryption algorithms. The closer R_{AB} is to 0 , the better the destruction effect is, and the more difficult it is to obtain image information through the relationship between adjacent pixels[27].

$$R_{AB} = \frac{\operatorname{cov}(A,B)}{\sqrt{D(A)}\sqrt{D(B)}} \tag{21}$$

where $\operatorname{cov}(A,B)$ is the covariance of A, B, and $\sqrt{D(A)}$ and $\sqrt{D(B)}$ are the standard deviations of A and B respectively. In this paper, the horizontal, vertical and diagonal correlations of the three RGB images of Lena, Lemon and Sakur are compared before and after encryption. The correlation values for the three RGB images are shown in Table 1, and the specific pixel distribution information is shown in Figure 6 and 7.

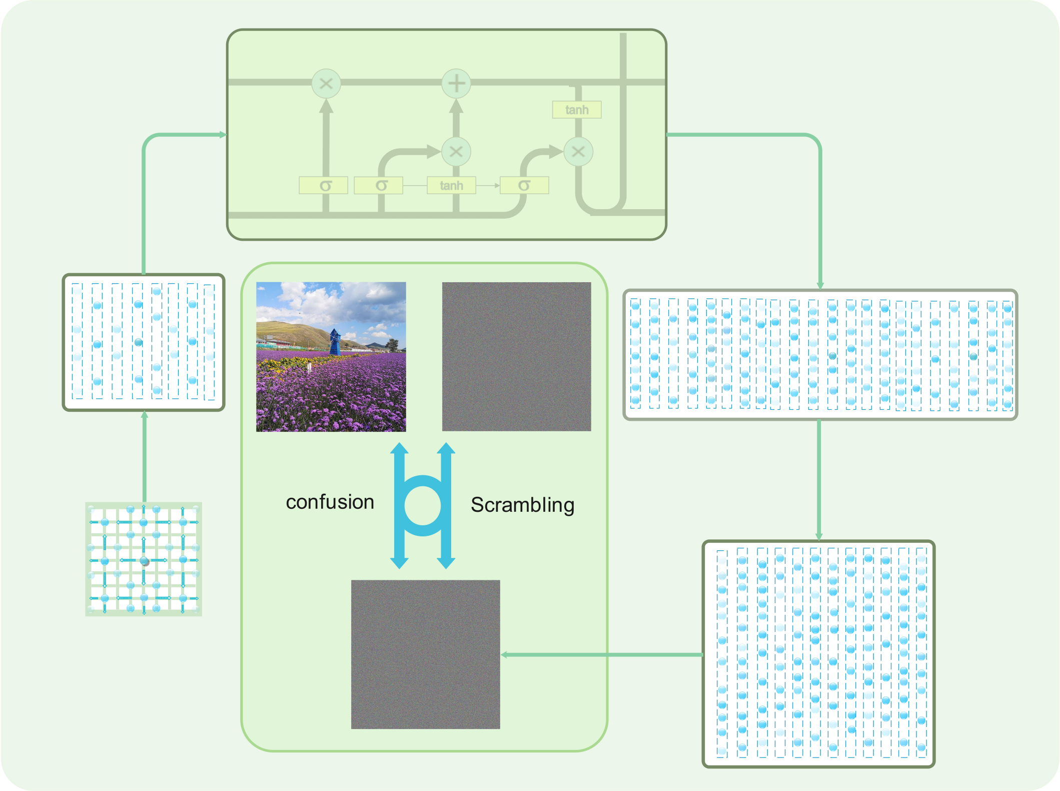

Figure 4. Encryption and decryption process

Table 1. Pixel correlation analysis data.

Image	Channel	Horizontal	Vertical	Diagonal
Unencrypted-a	Red	0.8846	0.8924	0.8297
	Green	0.9062	0.9146	0.8568
	Blue	0.9269	0.9272	0.8905
Encrypted-a	Red	0.0006	0.0011	0.0032
	Green	0.0032	0.0027	0.0021
	Blue	0.0041	0.0016	0.0022
Unencrypted-b	Red	0.9930	0.9944	0.9869
	Green	0.9940	0.9949	0.9897
	Blue	0.9927	0.9939	0.9876
Encrypted-b	Red	0.0022	0.0011	0.0023
	Green	0.0021	0.0025	0.0014
	Blue	0.0009	0.0041	0.0013

4.2.2. Histogram analysis

The histogram provides a visual representation of the statistical data of the pixel values in an RGB image. The histogram of a normal image usually has a distinct statistical pattern, and to resist statistical attacks[25], the histogram of an encrypted image must be as uniform and smooth as possible. The more such criteria are met, the more uniform the pixel distribution is, the less statistical information the image displays, the less information can be accurately predicted, and the more secure the image encryption scheme is [15]. In this paper, the histograms of the RGB three channels of Lena, Lemon, and Sakura images are analysed separately, and the specific histograms are shown in Figure 7 and 8.

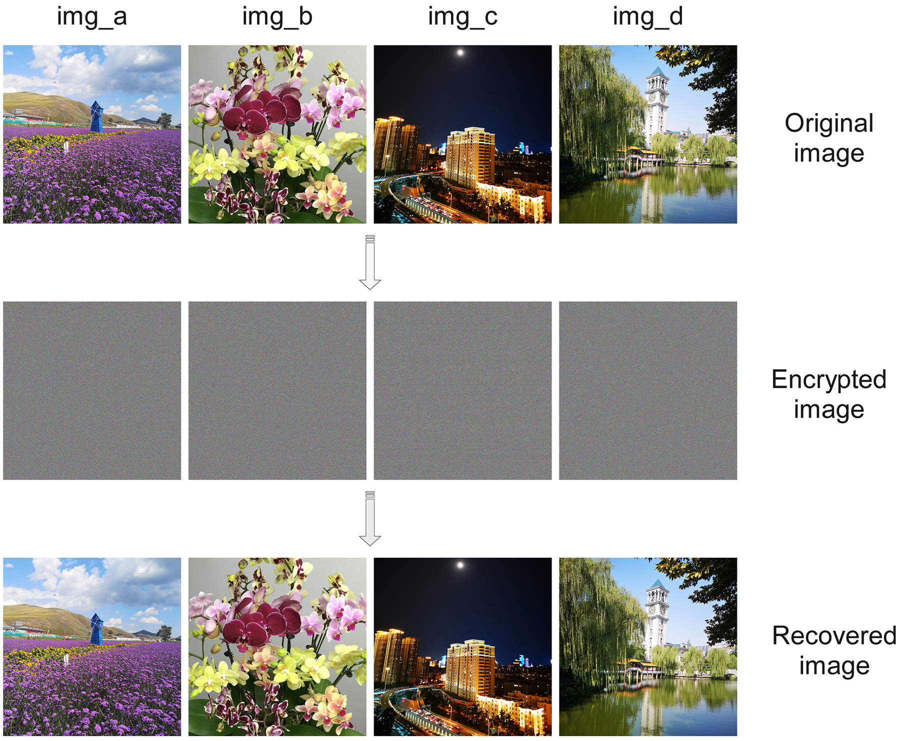

Figure 5. Comparison of correlation before & after img b encryption

4.2.3. Information entropy analysis 211

Information entropy H was proposed by Shannon, the father of information theory, to describe the uncertainty of the occurrence of each possible event of the information source. The pixel values of RGB images range from 0 to 255 , so the information entropy $H \leq 8$. The closer the entropy value is to 8 , the more information it carries and the more resistant it is to statistical attacks[11]. The formula for this is as follows:

$$H(m) = -\sum_{i=0}^{N-1} P(m_i) \log_2 P(m_i) \tag{22}$$

where m_i is the grey scale value and $P(x_i)$ is the probability of m_i occurrence. This 212
paper analyzes the information entropy of the R, G, and B channels of the three different 213
RGB images of Lena, Lemon, and Sakura. The relevant data are shown in Table 2. 214

Table 2. Entropy analysis.

Image	Channel	image entropy (bit)
Encrypted-a	Red	7.9991
	Green	7.9996
	Blue	7.9989
Encrypted-b	Red	7.9992
	Green	7.9992
	Blue	7.9994

4.2.4. Key sensitivity analysis 215

An effective key sensitivity means that a slight change in the key information will result in a significant change in the encrypted image. The ideal values of NPCR and UACI

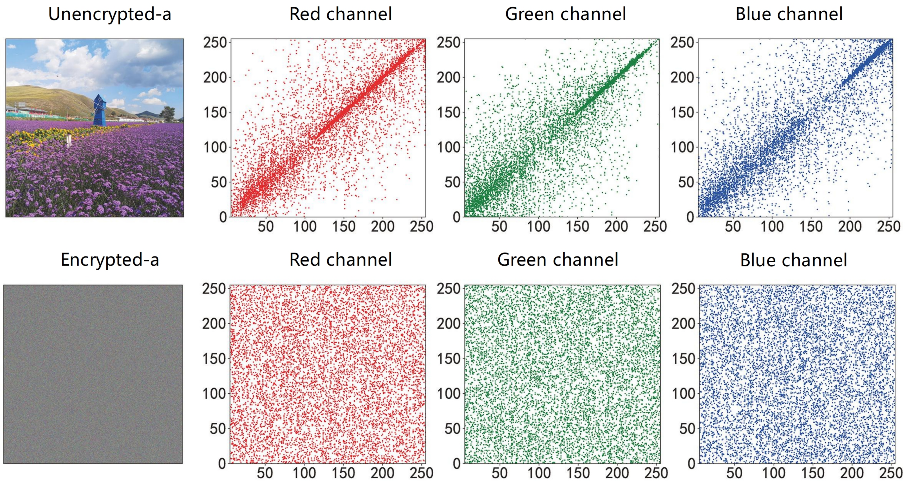

Figure 6. Comparison of correlation before & after img a encryption

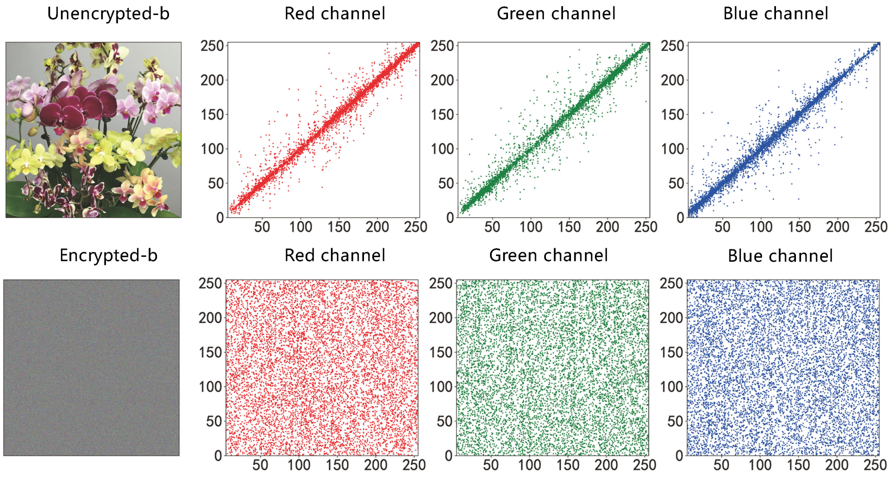

Figure 7. Comparison of correlation before & after img b encryption

are 99.61% and 33.46% respectively[29]. Higher calculated values of NPCI and UACI of an encryption scheme indicate that the encryption scheme is more resistant to differential attacks.

$$\Gamma(i,j) = f(x) = \begin{cases} 1, & \text{if } C_1(i,j) \neq C_2(i,j) \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

$$NPCR = \frac{\sum_{i,j} \Gamma(i,j)}{\mathfrak{J} \times \mathfrak{R}} \times 100\% \tag{24}$$

$$UACI = \frac{1}{\mathfrak{J} \times \mathfrak{R}} \left[\sum_{\mathrm{i,j}} \frac{\mathrm{C_1(i,j) - C_2(i,j)}}{255} \right] \times 100\% \tag{25}$$

where , $\Im, \Re$ are the length and width of the encrypted image, $\Gamma(i,j)$ is the above equation, and C1, C2 are the images after encryption with different keys.

In this paper, the key sensitivity of the R, G and B channels of the RGB images of Lena, Lemon and Sakura were analysed separately and the relevant data are shown in Table 3.

4.2.5. The key space

The key space refers to the set of all possible keys used to generate the key and determines whether the encryption scheme can resist a brute-force attack. Cryptosystems with a

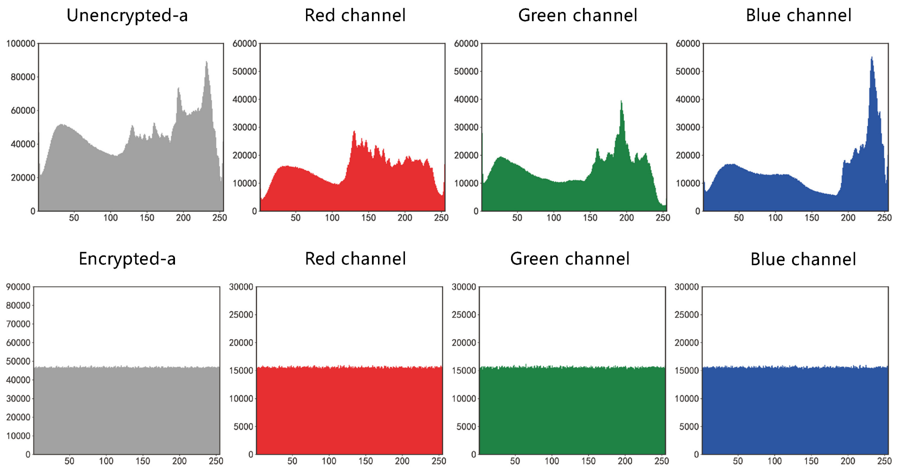

Figure 8. Comparison of histogram before & after img a encryption.

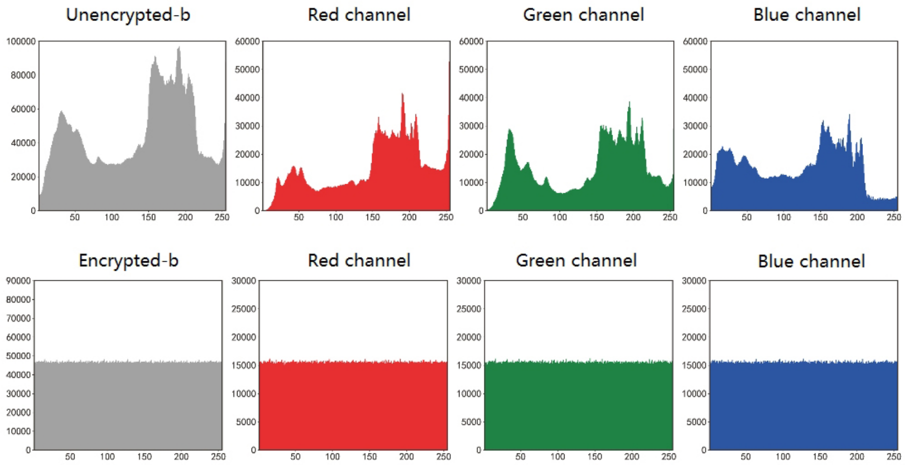

Figure 9. Comparison of histogram before & after img b encryption

key space size of $2^{1}28$ are effective in resisting brute force attacks. The key space calculation for the scheme proposed in this paper is based on quantum effects. Since in quantum theory the position of a particle in a defined space is not deterministic, each position has its probability of existence, only with different probabilities, and this probability can be changed by specifying the size of the space for a QW and the initial walking direction and forward direction. As the walk direction takes values from 0 to 2π and the QW is extremely sensitive to accuracy, the change in probability is infinite as the accuracy of the computer increases, i.e. the key space established based on the QW is infinite.

4.2.6. Explicit Attack

- Known plaintext attack: The attacker can recover the key by obtaining the decrypted image and comparing it with the ciphertext image. Since the algorithm in this paper has good diffusion effect, the difficulty of obtaining the key by this method is close to that of a direct brute force attack, so the encryption scheme in this paper can effectively resist known plaintext attacks.
- Selective plaintext attack:Assuming that the attacker has gained access to the encrypted machine, he can select an arbitrary number of plaintexts for the encryption algorithm under attack to encrypt and obtain the corresponding ciphertexts. The attacker’s goal is to gain some information about the encryption algorithm through this process that will allow the attacker to more effectively crack messages encrypted by the same

Table 3. Key Sensitivity Analysis

Image	Channel	NPCR	UACI
img_a	Red	99.6124%	33.4216%
	Green	99.6088%	33.3657%
	Blue	99.6003%	34.2157%
img_b	Red	99.6419%	33.6114%
	Green	99.5986%	33.4268%
	Blue	99.6036%	33.5762%

encryption algorithm (and associated key) in the future. In the worst case, the attacker can simply obtain the key used for decryption. This scheme is commonly used against public key encryption schemes. The keys in this scheme are not universal, i.e. they are changed periodically, even differently each time, making it impossible for an attacker to obtain valid information.

4.2.7. Time complexity analysis

The time complexity analysis of an encryption scheme is an important indicator to evaluate the excellence of an encryption scheme, which will directly affect the encryption efficiency. The time consumption of our proposed scheme consists of two parts, one is the time required to generate the key matrix, and the other is the completion of the image encryption by the key matrix. Although the efficiency of generating the pseudo-random number matrix is important, it is not part of the time complexity of the encryption scheme as it is decoupled from the image encryption process. The encryption time complexity of our proposed scheme consists of a combination of pixel obfuscation and scrambling. The time complexity of this process is $O(n^2 + n\log n)$, as the time consumed by matrix permutation is $O(n^2)$. In summary, the encryption time complexity of our proposed scheme is $O(2n^2 + n\log n)$.

4.2.8. Noise Robustness Testing

During the transmission of image information over the network, information may be lost or misplaced due to packet loss, malicious attacks, and so on. We simulate the continuous loss of image information due to network fluctuations using Gaussian noise and pretzel noise. A malicious attack was simulated using partial block replacement of the encrypted image. Figure 9 shows the decrypted image of the Lena encrypted image with the addition of Gaussian noise, pretzel noise and a clipping attack.

4.3. Comparison of encryption schemes

In this section, we analyse and compare the use of QW alone, the encryption scheme proposed in this paper and similar work in recent years in terms of the important measures of average relevance, information entropy, average NPCR, average UACI and key space size to resist brute-force cracking, the data of which are presented in Table 4.

5. Conclusions

We propose a more efficient encryption scheme for the current lack of encryption schemes for high pixel images in the field of image encryption. The probability density matrix generated by the quantum random walk is trained by exploiting the memory learning capability of the LSTM and the non-linear nature of the quantum random walk. It can take advantage of the nearly infinite key space brought by the quantum random walk algorithm, and also solve the shortcoming of the low generation efficiency of the quantum random walk itself. At the same time, both the permutation and obfuscation processes of

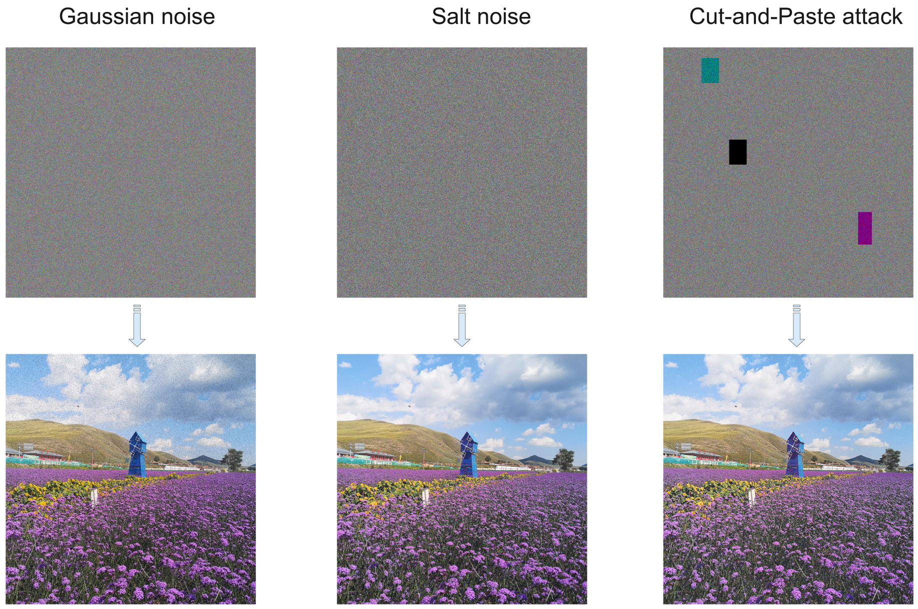

Figure 10. Comparison of histogram before & after img b encryption

Table 4. This is a table caption.

Scheme	NPCR(%)	UACI(%)	correlation	Entropy(bit)	KeySpace
QW	93.14	32.36	0.0149	7.9947	$>2^{128}$
our	99.6109	33.6024	0.0032	7.9992	$>2^{128}$
[4]	99.6127	33.4471	0.0013		$>2^{128}$
[30]	99.6336	33.4636	0.0026	7.9937	$>2^{128}$
[6]	99.6326	33.4022	0041	7.9973	$>2^{128}$

The comparison in this article is for reference only as the images used in the different solutions are different and have different pixels.
As the pixel sizes vary in each scenario, we have used the largest pixel images from their scenarios for comparison and selected their average values as a reference.

our scheme make use of the key space of the quantum random walk, avoiding the shortage of key space in a particular process.

Author Contributions: Conceptualization, J.Q.L. and Z.Y.S.; Methodology, J.Q.L. and Z.Y.S.; Software, Z.Y.S. and Z.W.S.; Validation, M.L and H.Y.M; Formal analysis, Z.W.S.; Investigation, Z.Y.S.; Resources, J.Q.L.; Data Curation, J.Q.L.; Writing-Original Draft Preparation, J.Q.L.; Writing-Review and Editing, S.Z.Y. and M.L.; Visualization, Z.W.S.; Supervision, M.L and H.Y.M.; Project Administration, M.L and H.Y.M.; Funding acquisition, H.Y.M.; Image Encryption, J.Q.L. and Z.Y.S.

Funding: This research was funded by Natural Science Foundation of Shandong Province, China (Grant Nos. ZR2021MF049),Joint Fund of Natural Science Foundation of Shandong Province (Grant Nos.ZR2022LLZ012),Joint Fund of Natural Science Foundation of Shandong Province (Grant Nos.ZR2021LLZ001),Project supported by the National Natural Science Foundation of China (Grant Nos. 11975132),National Natural Science Foundation of China (Grant Nos.12005110),Natural Science Foundation of Shandong Province, China (Grant Nos. ZR2022JQ04).

Conflicts of Interest: The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Coppersmith, Don. The Data Encryption Standard (DES) and its strength against attacks. *IBM journal of research and development* **1994**, *38*, 243–250.
2. Heron, Simon. Advanced encryption standard (AES). *Network Security* **2009**, *12*, 8–12.

3. Wang, Xing-Yuan and Li, Zhi-Ming. A color image encryption algorithm based on Hopfield chaotic neural network. *Optics and Lasers in Engineering* **2019**, *115*, 107–118.
4. Wang, Xingyuan and Yang, Jingjing. A privacy image encryption algorithm based on piecewise coupled map lattice with multi dynamic coupling coefficient. *Information Sciences* **2021**, *569*, 217–240.
5. Hua, Zhongyun and Zhu, Zhihua and Chen, Yongyong and Li, Yuanman. Color image encryption using orthogonal Latin squares and a new 2D chaotic system. *Nonlinear Dynamics* **2019**, *115*, 107–118.
6. Chai, Xiuli and Zhi, Xiangcheng and Gan, Zhihua and Zhang, Yushu and Chen, Yiran and Fu, Jiangyu. Combining improved genetic algorithm and matrix semi-tensor product (STP) in color image encryption. *Signal Processing* **2021**, *183*, 108041.
7. Zhou, Nanrun and Pan, Shumin and Cheng, Shan and Zhou, Zhihong. Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing. *Optics & Laser Technology* **2016**, *82*, 121–133.
8. Duan, Chen-Feng and Zhou, Jie and Gong, Li-Hua and Wu, Jun-Yun and Zhou, Nan-Run. New color image encryption scheme based on multi-parameter fractional discrete Tchebyshev moments and nonlinear fractal permutation method. *Optics and Lasers in Engineering* **2022**, *150*, 106881.
9. Chuman, Tatsuya and Sirichotedumrong, Warit and Kiya, Hitoshi. Encryption-then-compression systems using grayscale-based image encryption for JPEG images. *IEEE Transactions on Information Forensics and Security* **2018**, *14*, 1515–1525.
10. Chen, Liping and Yin, Hao and Huang, Tingwen and Yuan, Liguo and Zheng, Song and Yin, Lisheng. Chaos in fractional-order discrete neural networks with application to image encryption. *Neural Networks* **2020**, *125*, 174–184.
11. Mani, Prakash and Rajan, Rakkiyappan and Shanmugam, Lakshmanan and Joo, Young Hoon. Adaptive control for fractional order induced chaotic fuzzy cellular neural networks and its application to image encryption. *Information Sciences* **2019**, *491*, 74–89.
12. Hochreiter, Sepp and Schmidhuber, Jürgen. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735-1780.
13. Schmidhuber, Jürgen. A color image encryption algorithm based on Hopfield chaotic neural network. *Neural networks* **2015**, *61*, 85–117.
14. Steane, Andrew. Quantum computing. *Signal Processing: Image Communication* **2022**, *61*, 116891.
15. NanRun Zhou and Tian Feng Zhang and Xin Wen Xie and Jun-Yun Wu. Hybrid quantum–classical generative adversarial networks for image generation via learning discrete distribution. *IBM journal of research and development* **2019**, *115*, 107–118.
16. Wang, Haowen and Xue, Yunjia and Qu, Yingjie and Mu, Xiaoyi and Ma, Hongyang. Multidimensional Bose quantum error correction based on neural network decoder. *npj Quantum Information* **2022**, *8*,1–11.
17. Long, Gui-Lu. Grover algorithm with zero theoretical failure rate. *Physical Review A* **2001**, *64*, 107–118.
18. Weinstein, Y. S. and Pravia, M. A. and Fortunato, E. M. and Lloyd, S. and Cory, D. G. Implementation of the Quantum Fourier Transform. *Phys. Rev. Lett.* **2001**, *86*, 1889–1891.
19. Harrow, Aram W and Hassidim, Avinatan and Lloyd, Seth. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502.
20. Aharonov, Y. and Davidovich, L. and Zagury, N. Quantum random walks. *Phys. Rev. A* **1993**, *48*, 107–118.
21. Farhi, Edward and Gutmann, Sam. Quantum computation and decision trees. *Phys. Rev. A* **1998**, *58*, 915–928.
22. Watrous, John. Quantum simulations of classical random walks and undirected graph connectivity. *Journal of computer and system sciences* **2001**, *62*, 376–391.
23. Baryshnikov, Yuliy and Brady, Wil and Bressler, Andrew and Pemantle, Robin. Two-dimensional quantum random walk. *Journal of Statistical Physics* **2011**, *142*, 78–107.
24. Zhao, Zhi-Peng and Zhou, Shuang and Wang, Xing-Yuan. A new chaotic signal based on deep learning and its application in image encryption. *Acta Phys. Sin* **2021**, *70*, 23.
25. Yang, Yu-Guang and Pan, Qing-Xiang and Sun, Si-Jia and Xu, Peng. Novel image encryption based on quantum walks. *Scientific reports* **2015**, *5*, 107–118.
26. Abd EL-Latif, Ahmed A and Abd-El-Atty, Bassem and Venegas-Andraca, Salvador E.Controlled alternate quantum walk-based pseudo-random number generator and its application to quantum color image encryption. *Physica A: Statistical Mechanics and its Applications* **2020**, *547*, 123869.
27. Ma, Yulin and Li, Nachuan and Zhang, Wenbin and Wang, Shumei and Ma, Hongyang. Image encryption scheme based on alternate quantum walks and discrete cosine transform. *Optics Express* **2021**, *29*, 28338–28351.
28. Lam, Edmund Y and Goodman, Joseph W. A mathematical analysis of the DCT coefficient distributions for images. *IEEE transactions on image processing* **2000**, *9*, 1661–1666.
29. Wu, Yue and Noonan, Joseph P and Agaian, Sos and others. NPCR and UACI randomness tests for image encryption. *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)* **2011**, *1*, 31–38.
30. Hua, Zhongyun and Zhu, Zhihua and Chen, Yongyong and Li, Yuanman. Color image encryption using orthogonal Latin squares and a new 2D chaotic system. *Nonlinear Dynamics* **2019**, *104*, 4505–4522.