Image Encryption Scheme Based on Mixed Chaotic Bernoulli Measurement Matrix Block Compressive Sensing

Chen Yang, Ping Pan and Qun Ding *

Electronic Engineering College, Heilongjiang University, Harbin 150080, China; 2201658@s.hlju.edu.cn (C.Y.); 2201647@s.hlju.edu.cn (P.P.)
* Correspondence: 1984008@hlju.edu.cn

Abstract: Many image encryption schemes based on compressive sensing have poor reconstructed image quality when the compression ratio is low, as well as difficulty in hardware implementation. To address these problems, we propose an image encryption algorithm based on the mixed chaotic Bernoulli measurement matrix block compressive sensing. A new chaotic measurement matrix was designed using the Chebyshev map and logistic map; the image was compressed in blocks to obtain the measurement values. Still, using the Chebyshev map and logistic map to generate encrypted sequences, the measurement values were encrypted by no repetitive scrambling as well as a two-way diffusion algorithm based on GF(257) for the measurement value matrix. The security of the encryption system was further improved by generating the Secure Hash Algorithm-256 of the original image to calculate the initial values of the chaotic mappings for the encryption process. The scheme uses two one-dimensional maps and is easier to implement in hardware. Simulation and performance analysis showed that the proposed image compression–encryption scheme can improve the peak signal-to-noise ratio of the reconstructed image with a low compression ratio and has good encryption against various attacks.

Keywords: image encryption; block compressive sensing; Bernoulli measurement matrix; chaotic map; two-way diffusion

1. Introduction

With the rapid development of the internet and computer technology, more images, videos, and other multimedia information are being transmitted and stored through the internet. Image information security is receiving more attention. Image encryption is an effective method to protect digital images [1–6].

Image encryption methods are currently available using various techniques, such as chaotic systems [7–17], DNA coding [18–21], substitution boxes [22–24], and cellular automata [25–28]. Chaotic phenomena [29,30] are seemingly random irregular motions that occur in a deterministic system—a system described by a deterministic theory that behaves in an uncertain, irreducible, and unpredictable manner. Chaotic systems are the most widely used and continuously improved systems in the field of image encryption. Chaotic systems for image encryption are classified into two categories: one-dimensional (1D) chaotic systems [7–11] and multidimensional (MD) chaotic systems. One-dimensional chaotic systems have advantages, such as simple structures, easy implementation, and low computational complexity [11], but they also have disadvantages, such as small key spaces and low security [8,9]. To solve this problem, some MD chaotic systems have been applied to image encryption [12–17]. The encryption algorithms based on MD systems have larger key spaces and higher security levels; however, MD chaotic systems are relatively difficult to implement in hardware or software and the respective encryption systems have high computational complexity [11]. In this paper, we used Chebyshev mapping and logistic mapping to generate chaotic sequences for the encryption process and calculate the initial

Citation: Yang, C.; Pan, P.; Ding, Q. Image Encryption Scheme Based on Mixed Chaotic Bernoulli Measurement Matrix Block Compressive Sensing. *Entropy* 2022, 24, 273. https://doi.org/10.3390/e24020273
value of the one-dimensional chaotic system by the Secure Hash Algorithm-256 (SHA-256) function, which is not only easy to implement and increase the key space, but it also reduces the computational complexity of the cryptosystem and improves the security of the algorithm.

As a revolutionary data acquisition technique, compressive sensing (CS) [31–33] opens new perspectives for simultaneous compression and encryption of plaintexts, where the measurement matrix is shared between the sender and receiver as a key. To improve the confidentiality, linear feedback shift registers, pseudo-random number generators, and chaotic mappings were proposed to construct the measurement matrix [34–38]. The design of the measurement matrix is the key to ensuring the quality of signal sampling and to determine the hardware implementation of compressed sampling.

The methods used for constructing the measurement matrix are divided into two categories. One is the random measurement matrix, which mainly includes Gaussian measurement matrix, Bernoulli measurement matrix, local Fourier measurement matrix, etc. [32,39]. Although these matrices can reconstruct the original signals well, they are difficult to implement in hardware, in practical applications; storing the matrix elements requires a large storage space. Another type of matrix is the deterministic measurement matrix. In constructing such a matrix, once the system parameters and the parameters of the construction are determined, the elements of the matrix are also determined. The chaotic sequence generated by the chaotic system is a unification of determinism and randomness, and the measurement matrix constructed from the chaotic sequence is a deterministic matrix, which can effectively solve the problems of instability and the need for multiple measurements in the common random measurement matrix in experiments. Therefore, some scholars proposed to use chaotic sequences for the construction of compressive-aware measurement matrices [40–44]. L. Yu proposed using logistic chaotic sequences to construct measurement matrices in the literature [40] and proved that the matrices satisfied the RIP condition; Frunzete [41] et al. proved that the measurement matrices constructed by using tent chaotic sequences can satisfy the RIP condition. However, all of the above measurement matrix construction methods use chaotic sequences generated by a single chaotic mapping to construct the measurement matrix. The performance of the chaotic sequence generated by a single chaotic mapping is slightly worse than that of the hybrid chaotic sequence, so the performance of the measurement matrix is again influenced by the performance of the chaotic sequence [43]. In order to avoid affecting the reconstruction effect of the signal due to the lack of performance of a single chaotic sequence, a measurement matrix construction algorithm based on the XOR operation chaotic sequence is proposed to improve the reconstruction performance of the measurement matrix and the easy implementation feature of the hardware.

To reduce the computational complexity and storage space, the block compressed sensing (BCS) framework was introduced to process medium to large images. R. Huang et al. [45] designed a block cipher structure consisting of linear measurement, scrambling, S-box, and XOR operations. This scheme can work efficiently in the parallel computing environment. Zhu L et al. [46] conducted a novel BCS-based encryption scheme. The nonuniform sampling strategy was adopted in BCS stage to improve the compression efficiency, and the permutation–diffusion framework was utilized to enhance security. Experiments show that the algorithm has good reconstruction capability, but it has low correlation with plaintext images, and is vulnerable to known plaintext and selected plaintext attacks.

In this paper, we propose a simple and secure image encryption algorithm based on BCS and one-dimensional chaotic mapping. This scheme is made up of two primary stages. In the first stage, after generating chaotic sequences using two simple one-dimensional chaotic mappings—a Chebyshev map and logistic map for binary quantization and XOR operations—a Bernoulli measurement matrix was generated and the measurement value matrix was obtained by performing BCS on the original image. In the second stage, still using the Chebyshev map and logistic map to generate chaotic sequences, the measurement
value matrix was encrypted to obtain the ciphertext image. The simulation results verify the security, effectiveness, and good reconfiguration performance of the scheme. The main contributions of this paper are as follows.

1. A hybrid chaotic sequence measurement matrix satisfying Bernoulli’s property based on Chebyshev map and a logistic map, which is easy to implement in hardware, is designed, and the performance of the resulting measurement matrix is analyzed and applied to BCS.

2. The SHA-256 value of the plaintext image is used to calculate the initial value of the chaotic mapping for the encryption algorithm part, which increases the relevance of the algorithm to the plaintext image and improves the ability of the scheme to resist the selective plaintext attack.

3. A “no repetition” scrambling algorithm and a Galois domain-based two-way diffusion algorithm are proposed to encrypt the measurement value matrix and improve the security of the BCS framework.

4. In the reconstruction stage, the SPL reconstruction algorithm, based on DDWT, is used, which, combined with the hybrid chaotic sequence measurement matrix proposed in this paper, can achieve a much higher reconstruction quality of decrypted images with low compression ratio.

The rest of this paper is organized as follows. The preliminaries are discussed in Section 2. The proposed algorithm is described in Section 3. Simulation results and discussion are given in Section 4, and the statistical analysis of the scheme is discussed in Section 5. Conclusions are given in Section 6.

2. Preliminaries

In this section, BCS and the measurement matrix generation algorithm will be briefly introduced.

2.1. Block Compressive Sensing

In traditional compressive sensing (CS) image reconstruction methods, the observations are obtained by randomly projecting the image as a whole through the observation matrix. However, two-dimensional image signals often contain a large amount of information, which make the overall projection of the data very large, requiring large storage space and leading to high computational complexity in the image reconstruction process, causing a lot of problems for the specific implementation. Therefore, the compressed perception theory usually encounters two major obstacles when applied to video and images [47]: first, the observation matrix occupies a large amount of storage; second, the computational complexity is too high in the image reconstruction process. To solve this problem, the block compressive sensing (BCS) framework [48] is proposed, which is described in detail as follows.

The image $x$, with pixel $N = I_r \times I_c$, is divided into several $B \times B$ size image blocks. Denoting the $i$-th image block by $x_i$, $i = 1, 2, \cdots, N/B^2$. The same measurement matrix $\Phi_B$ is applied to each sub-block $x_i$ for the chunked image to obtain the measurement values:

$$y_i = \Phi_B x_i$$

(1)

where $\Phi_B \in R^{M_B \times B^2}$, and $M_B = \lfloor CR \times B^2 / N \rfloor$ is the number of samples of each sub-block and $CR$ is the compression ratio. The image is chunked; equivalently, the sampling matrix of the entire image is:

$$\Phi = \begin{bmatrix} \Phi_B & \Phi_B & \cdots & \Phi_B \end{bmatrix}$$

(2)
The observed values $y$ for the whole image are:

$$y = \Phi x = \begin{bmatrix} \Phi B x \\ \vdots \\ \Phi B x \end{bmatrix}$$  \hspace{1cm} (3)

Compared to the traditional image compressive sensing algorithm, the BCS algorithm solves the problem of large memory consumption required for the observation matrix in the image reconstruction process and greatly improves the speed of the image compressive sensing signal recovery. At the same time, the receiver does not have to wait for all of the sampling data transmissions to be completed, which is convenient for real-time processing of the signal.

Reconstruction algorithms for block compressive sensing can use orthogonal matching pursuit (OMP) \cite{49}, subspace pursuit (SP) \cite{50}, basis pursuit (BP) \cite{51}, etc., and can effectively recover the signal from the measured values.

In this paper, we use a smoothed PL (SPL) based on a dual-tree discrete wavelet transform (DDWT) \cite{52} to reconstruct the image. A block-based random image sampling was coupled with a projection-driven compressed-sensing recovery that encouraged sparsity in the domain of directional transforms simultaneously with a smooth reconstructed image. Both contourlets, as well as complex-valued dual-tree wavelets, were considered for their highly directional representations, while bivariate shrinkage was adapted to their multiscale decomposition structures to provide the requisite sparsity constraint. Smoothing was achieved via a Wiener filter incorporated into the iterative projected Landweber compressed-sensing recovery, yielding fast reconstruction. The proposed approach yields images with qualities that match or exceed that produced by a popular, yet computationally expensive, technique, which minimizes total variation. Additionally, reconstruction quality is substantially superior to that from several prominent pursuits-based algorithms that do not include any smoothing.

2.2. Measurement Matrix Generation Algorithm

2.2.1. Hybrid Chebyshev–Logistic Bernoulli Sequence

The Chebyshev chaotic map is defined as:

$$x_{n+1} = \cos(\omega \cdot \arccos x_n) \quad x_n \in [-1, 1]$$  \hspace{1cm} (4)

where $\omega$ is the order of the system. When the order $\omega > 2$, the map enters the chaotic state and the chaotic sequence can be generated by Equation (4). When $\omega > 4$, the Chebyshev map is in the full-rank mapping state, and the chaotic sequence generated by the system iteration is traversed on the interval $[-1, 1]$. The mean value of the chaotic sequence generated by the Chebyshev map is 0, and its probability density function is:

$$\rho_X(x) = \begin{cases} \frac{1}{\pi \sqrt{1-x^2}}, & x \in (-1, 1) \\ 0, & \text{else} \end{cases}$$  \hspace{1cm} (5)

The logistic chaotic map is defined as:

$$y_{n+1} = 1 - \lambda y_n^2, y_n \in [-1, 1]$$  \hspace{1cm} (6)

where $\lambda$ is the system parameters, when $\lambda \in (1.40015, 2]$, the Lyapunov exponent of the system are greater than 0, then the mapping is in the chaotic state. When $\lambda = 2$, the logistic map is a full mapping, the chaotic sequence generated by the iteration of Equation (6)
has good pseudo-randomness, and the generated chaotic sequences are distributed in the interval $[-1, 1]$. The probability density function of the logistic chaotic sequence is:

$$
\rho_X(y) = \begin{cases} 
\frac{1}{\pi \sqrt{1-y^2}}, & y \in (-1, 1) \\
0, & \text{else}
\end{cases}
$$

(7)

The chaotic sequence generated by the above chaotic system is a real-valued sequence, which is not suitable for transmission and processing in digital communication systems, so it is necessary to convert the real-valued chaotic sequence into a binary chaotic sequence. Here, the signum can be used to convert the chaotic sequence into a binary chaotic sequence. The signum formula is shown in Equation (8).

$$
\text{sgn}(x) = \begin{cases} 
0, & x < c \\
1, & x \geq c
\end{cases}
$$

(8)

where $c$ is the threshold value and $x$ is the value of the real-valued sequence.

**Theorem 1.** The chaotic sequence $\{x_n\}$ generated by the Chebyshev chaotic map, is transformed by the signum function to produce a spread spectrum sequence $\{x'_n\}$ satisfying the Bernoulli distribution, which is called the Chebyshev–Bernoulli (C–B) sequence.

**Theorem 2.** The chaotic sequence $\{y_n\}$ generated by the iteration of the logistic chaotic map, is transformed by the signum function to generate the spread spectrum sequence $\{y'_n\}$, which is a Bernoulli random sequence; that is, logistic–Bernoulli (L–B) sequence.

**Theorem 3.** The sequence generated by XOR operation of two Bernoulli random sequences is still the Bernoulli random sequence.

Equation (5) shows that the Chebyshev chaotic sequence $\{x_n\}$ is uniformly distributed in $[-1, 1]$. From Equations (5) and (8), the distribution of $x_n$ on the interval $[-1, 1]$ corresponding to a value of 0 or 1 for $x'_{n+1}$ can be found as follows:

$$
\text{sgn}(x'_{n+1}) = \begin{cases} 
0, & x_n \in (-1, 0) \\
1, & x_n \in (0, 1)
\end{cases}
$$

(9)

Similarly, the logistic chaotic sequence $\{y_n\}$ is uniformly distributed on $[-1, 1]$. The distribution of $y_n$ on $[-1, 1]$ corresponding to $y'_{n+1}$ when it is 0 and 1, respectively, can be derived from Equations (12) and (13).

$$
\text{sgn}(y'_{n+1}) = \begin{cases} 
0, & y_n \in \left(-1, -\frac{\sqrt{2}}{2}\right) \cup \left(\frac{\sqrt{2}}{2}, 1\right) \\
1, & y_n \in \left(-\frac{\sqrt{2}}{2}, 0\right) \cup \left(\frac{\sqrt{2}}{2}, 1\right)
\end{cases}
$$

(10)

Finally, a hybrid chaotic spread spectrum sequence is generated, as shown in Equation (11).

$$
z_n = x'_n \oplus y'_n
$$

(11)

XOR operation does not change the state of the sequence distribution. Therefore, the sequence $\{z_n\}$ is also uniformly distributed over $[-1, 1]$. From Equations (9) and (10), the distribution of $z_n$ on $[-1, 1]$ corresponding to $z_{n+1}$ taking values of 0 and 1 can be obtained as:

$$
\text{sgn}(z_{n+1}) = \begin{cases} 
0, & z_n \in \left(-1, -\frac{\sqrt{2}}{2}\right) \cup \left(0, \frac{\sqrt{2}}{2}\right) \\
1, & z_n \in \left(-\frac{\sqrt{2}}{2}, 0\right) \cup \left(\frac{\sqrt{2}}{2}, 1\right)
\end{cases}
$$

(12)
From Equation (12), the probability that \( z_{n+1} \) takes the value of 0 or 1 is 1/2. Treating \( \{z_n\} \) as a Markov chain, its one-step transfer probability matrix is:

\[
P = \begin{bmatrix}
0.5 & 0.5 \\
0.5 & 0.5 
\end{bmatrix}
\] (13)

The i-step transfer probability matrix \( P^i = P(i \geq 1) \) can be obtained from Equation (13). According to the properties of the Markov chain, it can be proved that the sequence \( \{z_n\} \) is random and independent statistics, so the sequence \( \{z_n\} \) is the Bernoulli sequence, and it is called the mixed Chebyshev and logistic Bernoulli (MCLB) sequence.

2.2.2. Construction of a Measurement Matrix Based on MCLB Sequence

The Bernoulli random observation matrix, as a common random measurement matrix, can satisfy the RIP property with great probability [53]. The mixed chaotic sequence proposed in the previous subsection is a random and statistically independent Bernoulli random sequence, so the measurement matrix constructed from the hybrid chaotic sequence satisfies the RIP property. Therefore, this paper proposes an algorithm to construct the measurement matrix based on the mixed chaotic sequence, and its flowchart is shown in Figure 1.

The specific construction steps are as follows:

- **Initial value** \( x_0 \)
  - Parameters \( \omega \)
  - Chebyshev map
  - Chebyshev chaotic sequence \( \{x_n\} \)
  - Binary quantization sequence \( \{x'_n\} \)
  - Mixed chaotic sequence \( \{z_n\} \)
  - Measurement matrix
- **Initial value** \( y_0 \)
  - Parameters \( \mu \)
  - Logistic map
  - Logistic chaotic sequence \( \{y_n\} \)
  - Binary quantization sequence \( \{y'_n\} \)

*Figure 1. Flowchart of the constructing the measurement matrix based on MCLB sequences.*
1. Set the parameters $\omega = 4$, $\lambda = 2$ of Equations (4) and (6), and given the initial values $x_0$, $y_0$, generate Chebyshev chaotic sequences $\{x_n\}$ and logistic chaotic sequences $\{y_n\}$ of the desired length iteratively from Equations (4) and (6).

2. Binary quantization of the chaotic sequences $\{x_n\}$ and $\{y_n\}$ using Equations (9) and (10), respectively, to obtain the chaotic spread spectrum sequence $\{x'_n\}$ and $\{y'_n\}$, and XOR operation to generate a hybrid chaotic sequence $\{z_n\}$.

3. The mixed chaotic sequence is cyclically shifted by columns to generate a measurement matrix of desired size, which is called the mixed Chebyshev and logistic Bernoulli matrix (MCLBM). Finally, the MCLBM is orthogonalized and normalized to obtain the final measurement matrix.

2.2.3. Simulation Test Based on MCLBM

This section verifies the reliability and validity of the MCLBM through image simulation experiments and compares the MCLBM with the Gaussian matrix (GM), Bernoulli matrix (BM), logistic–Bernoulli matrix (LBM), and Chebyshev–Bernoulli matrix (CBM) are compared.

BCS was used to test the “Lena” image with a size of $512 \times 512$. First, the image was divided into $32 \times 32$ blocks, the compression ratio was set to 0.3, different measurement matrices were used for compression measurements, and the DDWT-SPL algorithm was used in the reconstruction stage. MCLBM, GM, BM, LBM, and CBM were selected as measurement matrices for compressive reconstruction of “Lena” to compare the performance. In order to reduce the instability of GM and BM, they were averaged over 100 experiments. Given the parameters of the chaotic system, the constructed measurement matrices were intrinsically deterministic and were definite measurement matrices, so there was no need to repeat the experiments. We used the peak signal-to-noise ratio (PSNR) as an evaluation index for reconstructed images. The PSNR was the most widely used objective evaluation index for images. The PSNR is usually used to judge the quality of the decrypted image [54]. The larger the PSNR value, the smaller the difference between the two images and the higher the reconstruction accuracy.

PSNR is defined as follows:

$$\text{PSNR} = 10 \log \frac{255 \times 255}{\left( \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (X(i,j) - Y(i,j))^2 \right)}$$

(14)

where $X(i,j)$ is the value of the corresponding pixel point in the original image and $Y(i,j)$ is the value of the corresponding pixel point in the reconstructed image.

Figure 2 shows the effect plots of the reconstruction of the “Lena” image with different measurement matrices for a compression ratio of 0.3 compared with the original image. It can be seen from the figure that MCLBM has the best reconstruction effect and is closer to the original image.

For a more intuitive understanding of the performance of MCLBM, the experimental data of PSNR and reconstruction time of the five measurement matrices in reconstructing images at a compression ratio of 0.3 are given in Table 1. From the data in the table, it can be obtained that MCLBM outperforms several other measurement matrices in terms of PSNR at a compression ratio of 0.3. The reconstruction times of different measurement matrices are approximately the same, and the reconstruction time of MCLBM is slightly better than the other measurement matrices.

Figure 3 reflects the variation of the PSNR of the “Lena” reconstructed images for different compression ratios for the five measurement matrices GM, BM, LBM, CBM, and MCLBM. As can be seen in Figure 3, the higher the compression ratio, the higher the PSNR of the reconstructed image when using different measurement matrices to compress the reconstructed image. Before the compression ratio is less than 0.4, the PSNR of the reconstructed image with MCLBM as the measurement matrix is greater than that of other measurement matrices by about 1 dB. When the compression ratio is greater than
0.4, the effect of the compression ratio on the PSNR exceeds that of the measurement matrix, so the PSNR with MCLBM as the measurement matrix is not greater than that of other measurement matrices by 1 dB, but is still greater than that of other measurement matrices. In summary, the peak signal-to-noise ratio of the reconstructed image using MCLBM is always the largest, and the reconstruction effect is better than the other four measurement matrices.

Figure 2. Reconstructed images with different measurement matrices at a compression ratio of 0.3. (a) Original image; (b) GM reconstructed image; (c) BM reconstructed image; (d) LBM reconstructed image; (e) CBM reconstructed image; (f) MCLBM reconstructed image.

Figure 3. PSNR using different measurement matrices at different compression ratios.
3. The Proposed Algorithm

The main objective of the image compression/encryption algorithm proposed in this paper was to improve the performance of image compression, reconstruction, and encryption at low compression ratios; that is, with a reduced amount of more data. For this purpose, the plaintext images were blocked and measured using the MCLBM in the compressive measurement phase. To improve the resistance to selective plaintext attacks, the chaotic encryption sequence was generated for the plaintext image using the SHA-256 function to generate a key as the initial value of the chaotic map, and a new image encryption scheme was designed accordingly.

The proposed compression/encryption algorithm consists of two main processes, BCS compression sampling and quantization encryption of the measured values, as shown in Figure 4. The specific steps are discussed in detail in the following sections.

![Image Encryption Process Diagram](image)

**Figure 4.** The proposed image encryption and decryption process.

### 3.1. Block Compression Sampling and Quantification Process

Assuming the BCS compression ratio is CR, the size of the plaintext image P is $M \times N$, and it is divided into non-overlapping blocks of size $B \times B$, which can be divided into $n_B = \frac{M \times N}{B^2}$ blocks, and then these blocks are expanded into a one-dimensional vector in the scanning order from left to right and then from top to bottom, and noted as $S_i$, $i = 1, 2, \cdots, n_B$. The BCS compression sampling process is as follows:

1. Generate the MCLBM sequence $\{z_{ij}, i = 1, 2, \cdots, L\}$, from the steps in Section 2.2.2, and $L = m \times n, n = B \times B, m = \text{floor}(CR \times n)$.

#### Table 1. Performance comparison of different measurement matrices at a compression ratio of 0.3.

| Measurement Matrix | PSNR/dB | Time/s |
|--------------------|---------|--------|
| GM                 | 32.82   | 3.50   |
| BM                 | 32.25   | 3.70   |
| LBM                | 32.45   | 3.21   |
| CBM                | 32.57   | 3.56   |
| MCLBM              | 33.82   | 3.18   |
2. Generate a measurement matrix \( \Phi_0 \) of size \( m \times n \) by cyclically shifting \( \{ z_j \} \) by columns:

\[
\Phi_0 = \begin{pmatrix}
        z_1 & z_{m+1} & \cdots & z_{mn-m+1} \\
        z_2 & z_{m+2} & \cdots & z_{mn-m+2} \\
        \vdots & \vdots & \ddots & \vdots \\
        z_m & z_2m & \cdots & z_{mn}
    \end{pmatrix}
\]  

(15)

3. Express \( \Phi_0 \) as a row vector as \( \Phi_0 = (\alpha^T_1, \alpha^T_2, \ldots, \alpha^T_M) \). Let \( \beta_1 = \alpha_1 \) and use Equations (16) and (17) to normalize the row vectors.

\[
\beta_r = \alpha_r - \sum_{i=1}^{r-1} \frac{\langle \beta_i, \alpha_r \rangle}{\| \beta_i \|} \beta_i \quad r = 2, 3, \ldots, M
\]

(16)

\[
\eta_j = \frac{\beta_i}{\| \beta_j \|} \quad j = 1, 2, \ldots, M
\]

(17)

where \( \langle \alpha, \beta \rangle \) denotes the inner product of vector \( \alpha \) and \( \beta \), and \( \| \beta \| \) denotes the 2-norm of vector \( \beta \). Finally, measurement matrix \( \Phi_i = (\eta^T_1, \eta^T_2, \ldots, \eta^T_M) \), \( i = 1, 2, \ldots, n_B \) after orthogonal normalization is obtained.

4. Compress the plaintext image \( P \) after blocked and obtain the measurement value \( C_i \) of each block.

\[
C_i = \Phi_i S_i \quad i = 1, 2, \ldots, n_B
\]

(18)

Finally, \( C_i \) is combined in blocks to obtain the measurement value matrix \( C \), at which point the measurement value matrix is

\[
C = (C_1, C_2, \ldots, C_{n_B})
\]

(19)

At this time, the size of \( C \) is \( m \times n_B \), and the data are obviously smaller than that of the original image. If the compression ratio is \( CR = 0.25 \), then the size of \( C \) is only \( 1/4 \) of the size of the plaintext image, which is convenient for the subsequent encryption processing.

The output matrix values after CS measurements are usually outside the grayscale range of \([0, 255]\). Therefore, it must be quantized before encryption. The quantization process is expressed as follows:

\[
P_0 = \text{floor} \left( \frac{255 \times (Y - \text{min})}{\text{max} - \text{min}} \right)
\]

(20)

where floor(\( x \)) represents the maximum integer value that is not greater than \( x \) and min and max are the minimum and maximum elements of the measurement matrix \( Y \), respectively.

3.2. Cipher Sequence Generation Algorithm

In order to make the proposed encryption algorithm resistant to known plaintext and selected plaintext attacks, this paper uses the SHA-256 value of the plaintext image to calculate the parameters and initial values of the variable parameter chaotic mapping, so that the parameters and initial values are closely related to the plaintext image. The specific steps are as follows.

The 256-bit hash of the plaintext image is first obtained, written in 64-bit hexadecimal form, and then each bit is converted to the corresponding decimal number, which is used as the key \( K \) and is denoted as:

\[
K = k_1, k_2, \ldots, k_{64}
\]

(21)

where \( k_i \in (0, 16), i = 1, 2, \ldots, 64 \). The encryption algorithm requires the use of Chebyshev chaotic map and the logistic chaotic map to generate two cipher sequences, so generate
their initial values $x'_0$, $\omega'$, $y'_0$, and $\lambda'$ according to the key $K$, generated by the following equation.

\[
\begin{align*}
  x'_0 &= -1 \times \frac{1}{256} \sum_{i=1}^{16} k_i + 1 \\
  \omega' &= 2 \times \frac{1}{256} \sum_{i=17}^{32} k_i \\
  y'_0 &= -2 \times \frac{1}{256} \sum_{i=33}^{48} k_i + 1 \\
  \lambda' &= -0.6 \times \frac{1}{256} \sum_{i=49}^{64} k_i + 2
\end{align*}
\]

(22)

Then the initial values $x'_0$, $\omega'$, $y'_0$, and $\lambda'$ generated by SHA-256 are substituted into Equations (4) and (6) to generate a chaotic sequence $S = \{s_1, s_2, \cdots, s_{mn_B}\}$ and a chaotic sequence $Z = \{z_1, z_2, \cdots, z_{2mn_B}\}$, respectively, after which the elemental values of the chaotic sequences $S$ and $Z$ are quantized using the following equation:

\[
\begin{align*}
  X &= \text{mod} (\text{floor} ((S + 100) \times 10^{10}), m \times n_B) \\
  Y &= \text{mod} (\text{floor} (Z \times 2^{16}), 256)
\end{align*}
\]

(23)

Sequence $X$ is used for scrambling and $Y$ for diffusion.

3.3. Image Encryption Algorithm

The encryption process begins with a scrambling of the matrix of measurement values, followed by a diffusion operation, as follows.

1. First, keep only one of the recurring pseudo-random numbers in the pseudo-random sequence $X$ (i.e., the first occurrence), then add the values in $\{1, 2, \cdots, mn_B\}$ that do not appear in $X$ to the end of $X$ in descending order, and note $X_1$. $X_1$ is then used to displace $C$ without repetition. $C$ is then expanded column-wise into a one-dimensional vector, denoted $C_1$. Finally, $C_1(X_1(i))$ is swapped with $C_1(X_1(mn_B - i + 1))$ to give the disordered matrix $C_2$.

2. Select the first $mn_B$ values of the sequence $Y$ to form the sequence $Y_1$. $C_3$ is obtained by forward diffusion of $C_2$ using $Y_1$ as follows.

\[
\begin{align*}
  C_3(1) &= C_3(0) \times Y_1(1) \times C_2(1) \\
  C_3(i) &= C_3(i - 1) \times Y_1(i) \times C_2(i) \quad i = 2, 3, \cdots, mn_B
\end{align*}
\]

(24)

where “$\times$” represents the multiplication operation of GF(257) domain and $C_3(0) = 0$.

3. Select the values from $mn_B + 1$ to $2mn_B$ of the sequence $Y$ to form the sequence $Y_2$. Use $Y_2$ to perform backward diffusion on $C_3$ to obtain $C_4$, as follows.

\[
\begin{align*}
  C_4(mn_B) &= C_4(0) \times Y_2(mn_B) \times C_3(mn_B) \\
  C_4(i) &= C_4(i + 1) \times Y_2(i) \times C_3(i) \quad i = mn_B - 1, mn_B - 2, \cdots, 1
\end{align*}
\]

(25)

where “$\times$” represents the multiplication operation of GF(257) domain and $C_4(0) = 0$.

Finally, $C_4$ is rearranged into a matrix of $m \times n_B$ by columns, and after the above operation, the ciphertext image $C_5$ is obtained.

3.4. Image Decryption and Reconstruction Process

The decompression and decryption process is the opposite of the compression and encryption process. The steps are summarized as follows:

1. Use the key $K$ to generate the chaotic mapping initial values $x'_0$, $\omega'$, $y'_0$, and $\lambda'$, and generate $X$ and $Y$ by Equation (23).

2. The received ciphertext image $C_5$ is sequentially subjected to backward diffusion, forward diffusion, and scrambling algorithms to obtain the quantized measurement value matrix $P_0$. 
3. The measurement matrix $C$ is recovered by inverse quantization of the quantized measurement matrix $P_0$ by the following equation:

$$C = \frac{P_0 \times (\max - \min)}{255} + \min$$  \hspace{1cm} (26)

4. The parameters $\omega$, $\lambda$ and the initial value $x_0$, $y_0$ of the MCLB transmitted by the sender are used to iterate and generate the MCLBM, that is, the measurement matrix $\Phi_i$.

5. The DDWT-SPL algorithm is used to reconstruct the recovered measurement matrix $C$ to obtain a reconstructed original image $P$ of size $M \times N$.

### 4. Simulation and Performance Analysis

#### 4.1. Image Encryption Algorithm

To verify the effectiveness of the proposed scheme, simulation results and performance analyses are given in this section. The experiments were performed on a computer with 2.50 GHz CPU and 8 GB RAM using MATLAB R2017a. Different grayscale images (size $512 \times 512$) are selected for simulation tests, such as “Lenna”, “Peppers”, “Cell”, and “X-ray”. The initial value of MCLBM was set to $\omega = 2$, $\lambda = 4$, $x_0 = 0.169877$, $y_0 = 0.2555$. The compression ratio of the compression measurement was $CR = 0.25$, and the size of the plain image blocks were all $32 \times 32$. In the decryption stage, DDWT-SPL was used as the reconstruction algorithm.

The simulation results are shown in Figure 5. The first to third columns represent the plaintext image, the encrypted image, and the decrypted reconstructed image, respectively. From the final encryption results, no meaningful information can be obtained from the ciphertext image in the third column. The compression and reconstruction results show that the size of the ciphertext image is only the size of the original image, which greatly reduces the image file size, storage space, and transmission bandwidth required for transmission. In addition, as shown in the third column, the decrypted reconstructed image is visually very similar to the corresponding plaintext image.

#### 4.2. Impact of Important Parameters on Encryption and Decryption Performance

In this scheme, two important parameters, block size and compression ratio, both affect the encryption and decryption results. Next, we will evaluate their effects on the simulation results for better performance.

Firstly, the influence of block size on the reconstructed image PSNR is discussed when the compression ratio is the same. The “Lenna” image of $512 \times 512$ size was used for compressed encryption as well as decrypted reconstruction, with the compression ratio set to 0.25. Table 2 lists the PSNR values and the consumption time for different block sizes. It can be seen from Table 2 that when the compression ratio is the same, the PSNR value and the time consumed increase as the block size increases, but the PSNR value increases insignificantly and the time consumed increases too much, which has a side effect instead, so a high PSNR value can be achieved by choosing a block size of $32 \times 32$.

Then, we discuss the PSNR of the reconstructed image with a block size of $32 \times 32$ and different image compression–encryption and decryption at different compression ratios, as shown in Figure 6. It can be seen that the PSNR values of the different reconstructed images can still reach more than 30 dB at a compression ratio of 0.2, which indicates that the compression–encryption scheme proposed in this paper can achieve a high PSNR value at a low compression ratio.

Finally, the compression–aware compression ratio in the algorithm of this paper is set to 0.25 and the block size is $32 \times 32$. Table 3 compares the scheme of this paper with other compression–encryption schemes, and it can be seen that the algorithm proposed in this paper has higher PSNR values than other algorithms under the same compression ratio, which indicates that it has good compression reconstruction performance.
Figure 5. Simulation results. (a1–a4) Plaintext images; (b1–b4) ciphertext images; (c1–c4) decrypted images.

Table 2. Comparison of PSNR and consumption time for different block sizes with the same compression ratio.

| Block Size | PSNR/dB   | Time/s     |
|------------|-----------|------------|
| 32 × 32    | 32.7265   | 8.955823   |
| 64 × 64    | 32.9089   | 12.1209232 |
| 128 × 128  | 33.1612   | 153.747594 |
Finally, the compression–aware compression ratio in the algorithm of this paper is set to 0.25 and the block size is 32 × 32.

Table 3 compares the scheme of this paper with other compression–encryption schemes, and it can be seen that the algorithm proposed in this paper has higher PSNR values than other algorithms under the same compression ratio, which indicates that it has good compression reconstruction performance.

Table 3. PSNR/dB of different algorithms.

| Image  | CR | Ours       | Reference [55] | Reference [56] |
|--------|----|------------|----------------|----------------|
| Lena   | 0.25 | 32.7265  | 31.4240        |                |
| Peppers|     | 32.9463   | 30.6809        |                |
| Lena   | 0.5  | 36.7805   | 33.2299        | 23.3608        |
| Peppers|     | 36.4142   | 32.1889        | 27.3366        |
| Lena   | 0.75 | 41.0781   | 34.1313        | 34.7149        |
| Peppers|     | 40.1133   | 33.1721        | 35.8463        |

5. Statistical Analysis

5.1. Key Space

For a secure image encryption algorithm, the key space is at least $2^{100}$ to resist brute force attacks. In this scheme, the key consists of two parts: (1) the parameters $\omega$, $\lambda$ and the initial value $x_0$, $y_0$ of the MCLB; and (2) the SHA-256 hash function of the plaintext image. According to the IEEE floating point standard, the precision of the double precision values is approximated as $10^{-15}$, SHA-256 key space can be up to $2^{128}$. Therefore, the key space of the proposed scheme in this paper is $2^{300}$, greater than $2^{100}$. In other words, the key space of the algorithm meets the security criteria and can resist exhaustive attacks. Table 4 compares the key space of this paper with that of other compression–encryption schemes, and the results in Table 4 show that the proposed image encryption algorithm is secure enough to resist brute-force attacks.

Table 4. Comparison of key space.

| Algorithm | Ours       | Reference [55] | Reference [57] | Reference [58] |
|-----------|------------|----------------|----------------|----------------|
| Key space | $2^{300}$  | $6.561 \times 10^{67}$ | $10^{80}$     | $10^{75}$      |

5.2. Key Sensitivity

Key sensitivity analysis aims to analyze the difference between two ciphertext images obtained by encrypting the same plaintext image when there is a small change in the key. If the algorithm is sensitive to the key, it means that if the input decryption key is slightly
different from the correct one when decrypting, no valuable information about the plain image can be obtained from the decrypted result. This section uses the plain image “Lena” as an example to test the key sensitivity of the algorithm. The sensitivity of the chaotic map to the initial conditions determines the sensitivity of the encryption system to the key. Therefore, the 256-bit hash value of the plaintext is changed from K to K0, K1, and K2 with bit modifications, which are shown below.

\( K = \text{eb1197ae40dd90f096474ebf6a25494802027a}0a8b8f81c9d9a2e623c0b0a \)
\( K_0 = \text{fb1197ae40dd90f096474ebf6a25494802027a}0a8b8f81c9d9a2e623c0b0a \)
\( K_1 = \text{eb1197ae40dd90f006474ebf6a25494802027a}0a8b8f81c9d9a2e623c0b0a \)
\( K_2 = \text{eb1197ae40dd90f006474ebf6a25494802027a}0a8b8f81c9d9a2e623c0b0a \)

The “Lena” image is encrypted with K to K0, K1, and K2 to obtain different cipher images, as shown in Figure 7b–e. Figure 7f–g shows the difference between two ciphertexts; as seen from Figure 7f–g, the “difference image” of two ciphertexts obtained by encrypting the same plaintext image shows a noise style, intuitively, it reflects that the two ciphertexts differ significantly; that is, the image encryption system has key sensitivity. Figure 7l shows the images decrypted by different “wrong” keys; it can be seen that the information of the original image cannot be obtained in the decrypted images, which indicates that the image decryption system is key sensitive.

![Figure 7](image-url)

**Figure 7.** Results of wrong key encryption and decryption. (a) The plaintext image; (b–e) encrypted plaintext images using K, K0, K1, and K2, respectively; (f) Difference image of encrypted image ciphertext using K and K0; (g) Difference image of encrypted image ciphertext using K and K1; (h) Difference image of encrypted image ciphertext using K and K2; (i) Decrypt the (b) ciphertext image using K; (j–l) Decrypt the (b) ciphertext image using K0, K1, and K2, respectively.
To quantify the differences between ciphertext images obtained from the same plaintext image, the pixel change range (NPCR) and the uniform average change intensity (UACI) were calculated using the correct and incorrect keys for encrypting the images, and the results are shown in Table 5. NPCR and UACI are calculated as follows.

\[
\begin{align*}
\text{NPCR} &= \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i, j) \\
\text{UACI} &= \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| \frac{C_1(i, j) - C_2(i, j)}{255} \right|
\end{align*}
\]

(27)

where, \(D(i,j)\) is the Heaviside function, \(D(i,j) = 0\) when \(C_1(i,j) = C_2(i,j)\), and \(D(i,j) = 1\) when \(C_1(i,j) \neq C_2(i,j)\). The theoretical expectation of NPCR is 996.6034% and the theoretical expectation of UACI is 33.4635%.

Table 5. NPCR and UACI for cryptographic image using the correct key and different incorrect keys.

| Secret Keys | K and K_0 | K and K_1 | K and K_2 |
|-------------|-----------|-----------|-----------|
| NPCR (%)    | 99.6223   | 99.6074   | 99.6013   |
| UACI (%)    | 33.4190   | 33.4347   | 33.3847   |

Table 5 shows that both NPCR and UACI are close to the theoretical values and more than 99.6% of the pixels are modified, which means that the encryption process is highly sensitive to the key.

Table 6 lists the NPCR and UACI values in Figure 7i–l, where more than 99.5% of the pixels are changed when the key is slightly changed, which indicates that the decryption process is highly sensitive to the key.

Table 6. NPCR for decrypting images using the correct key and different incorrect keys.

| Secret Keys | K and K_0 | K and K_1 | K and K_2 |
|-------------|-----------|-----------|-----------|
| NPCR (%)    | 99.5671   | 99.5725   | 99.6372   |

5.3. Histogram

The histogram is a fundamental property of digital images. It reflects the distribution of pixel grayscale values in the image. The more uniform the histogram distribution of the ciphertext image, the better the encryption effect [9]. As can be seen from Figure 8, the pixel value distributions of different plaintext images have their own characteristics, and their ciphertext image histograms are all approximately uniformly distributed, so that the attacker cannot obtain any statistical information from the ciphertext image. Therefore, the image compression and encryption algorithm proposed in this paper is able to resist statistical analysis attacks.

5.4. Information Entropy Analysis

The information entropy reflects the uncertainty of the image information, and it is generally believed that the greater the entropy, the greater the uncertainty, and the greater the amount of information, the smaller the visible information. The formula for calculating information entropy is shown below:

\[
H = -\sum_{i=0}^{L} p(i) \log_2 p(i)
\]

(28)

where \(L\) is the gray level of the image and \(p(i)\) denotes the probability of occurrence of the gray value \(i\).

For a grayscale random image with \(L = 256\), the theoretical value of information entropy is 8. Table 7 presents the information entropy of different cipher images for
different compression ratios. From Table 7, it can be seen that the information entropy of each cipher image is very close to the theoretical value, even under different compression ratios. Therefore, the compression–encryption scheme proposed in this paper can achieve both image compression and encryption, and the encrypted images have good randomness and sufficient security.

![Histograms of the original and encrypted images.](image)

**Figure 8.** Histograms of the original and encrypted images. (a) Plaintext histogram of Lena; (b) plaintext histogram of Peppers; (c) plaintext histogram of Cell; (d) plaintext histogram of X-ray; (e) ciphertext histogram of Lena; (f) ciphertext histogram of Peppers; (g) ciphertext histogram of Cell; (h) ciphertext histograms of X-ray.

**Table 7.** Information entropy of different cipher.

| CR   | Lena | Peppers | Cell | X-ray |
|------|------|---------|------|-------|
| 0.25 | 7.9972 | 7.9971 | 7.9975 | 7.9975 |
| 0.5  | 7.9986 | 7.9988 | 7.9986 | 7.9983 |
| 0.75 | 7.9990 | 7.9990 | 7.9991 | 7.9989 |

5.5. **Correlation Analysis**

The correlation coefficient of adjacent pixels can be used to evaluate the image encryption effectiveness. Generally, a relatively good digital image encryption scheme can make the correlation between adjacent pixels of the ciphertext image relatively low [58,59]. The closer the correlation coefficient of adjacent pixels to 0, the better the encryption effect. In this paper, for 10,000 pairs of pixels selected in the image, the correlation coefficient of adjacent pixels is calculated by the formula:

\[
\begin{align*}
    r_{uv} &= \frac{\text{cov}(u,v)}{\sqrt{D(u)}\sqrt{D(v)}} \\
    \text{cov}(u,v) &= \frac{1}{N} \sum_{i=1}^{N} (x_i - E(u))(y_i - E(v)) \\
    D(u) &= \frac{1}{N} \sum_{i=1}^{N} (u_i - E(u))^2 \\
    E(u) &= \frac{1}{N} \sum_{i=1}^{N} u_i
\end{align*}
\]  

(29)

where \(N\) denotes any \(N\) pair of adjacent pixel points in the image, \((u_i, v_i), i = 1, 2, \ldots, N\) denotes the grayscale value of \(N\) pairs of adjacent pixel points, \(E(u)\) and \(E(v)\) are the
averages of all \( u \) and \( v \). In addition, \( D(u) \) is the variance, \( \text{cov}(u, v) \) is the covariance, and \( r_{uv} \) is the correlation coefficient. Lenna, Peppers, Cell, and X-ray of size 512 × 512 were used as plaintext images in this experiment. To calculate the correlation coefficients, 10,000 pairs of pixels were randomly selected from the plaintext and ciphertext images in horizontal, vertical, and diagonal directions. The calculation results are shown in Table 8.

### Table 8. Correlation coefficients of neighboring pixels in different images.

| Direction | Lenna | Peppers | Cell | X-ray |
|-----------|-------|---------|------|-------|
|           | Plaintext | Ciphertext | Plaintext | Ciphertext | Plaintext | Ciphertext | Plaintext | Ciphertext |
| Horizontal | 0.9845 | −0.0032 | 0.9751 | 0.0075 | 0.9893 | 0.0163 | 0.9985 | 0.0410 |
| Vertical  | 0.9746 | 0.0123 | 0.9766 | 0.0129 | 0.9904 | −0.0059 | 0.9976 | −0.0148 |
| Diagonal  | 0.9703 | −0.0071 | 0.9649 | −0.0051 | 0.9792 | 0.0215 | 0.9964 | −0.0082 |

As can be seen from Table 8, the correlation coefficients of neighboring pixels in the plain images in all three directions are greater than 0.9, indicating that the correlation between neighboring pixels is high. The correlation of adjacent pixels in the images encrypted with the algorithm proposed in this paper is close to 0. Using Lena as an example, 10,000 pairs of pixels were selected from the original and encrypted images in the horizontal, vertical, and diagonal directions. Their adjacent pixel correlations in horizontal, vertical, and diagonal directions are plotted in Figure 9, which shows the pixel distribution before and after encryption of the Lena image; it can be seen that the pixel distribution follows a certain pattern, and the grayscale values from the ciphertext image are randomly distributed between 0 and 255.

![Figure 9](image)

**Figure 9.** Distribution of adjacent pixels. (a) Plaintext horizontal adjacent pixels; (b) Plaintext vertical adjacent pixels; (c) Plaintext diagonal adjacent pixels; (d) Ciphertext horizontal adjacent pixels; (e) Ciphertext vertical adjacent pixels; (f) Ciphertext diagonal adjacent pixels.

### 5.6. Choose Plaintext Attack

According to Kerckhoffs’s principle, the security of an encryption system depends on the key and not on the encryption algorithm itself. The choose plaintext attack [10,54,60], implies that an attacker temporarily gains access to use the encryption machine, so that
he can encrypt any plaintext and obtain the corresponding ciphertext to decrypt all (or part) of the plaintext and the key. The chosen plaintext attack is the strongest attack, and if a cryptosystem can resist this attack, then it must be able to resist other attacks. In this paper, the SHA-256 hash function of the original image was calculated as the initial value of the chaotic system. Therefore, when different plaintext images are encrypted, the corresponding encryption sequences change, so that a hacker cannot obtain useful information by encrypting some specific images. Therefore, the scheme can resist well to selective plaintext attacks.

5.7. Ciphertext Sensitivity

The purpose of ciphertext sensitivity analysis is to analyze the difference between the restored image and the original plaintext image after a small change in the ciphertext image by the decryption system. If the restored image is very different from the original plaintext image, the image decryption system is said to have strong ciphertext sensitivity; if the restored image is not very different from the original plaintext image, the image cryptosystem is said to have weak ciphertext sensitivity, and then this type of image cryptosystem is deficient in fighting against selected ciphertext attacks or known ciphertext attacks. Moreover, a small change in the ciphertext image is a small change in the value of one or more pixel points of a given ciphertext image.

In this paper, different plaintext images \( P \) were selected, and the corresponding ciphertext image \( C_1 \) was obtained by encrypting \( P \) using this encryption system. A pixel point was randomly selected from \( C_1 \) and the value of this pixel point was changed by the amount of 1, the changed image was noted as \( C_2 \), and the restored image \( P_1 \) was obtained by decrypting \( C_2 \). The NPCR and UACI values of \( P \) and \( P_1 \) were calculated and listed in Table 9.

| Images  | NPCR (%) | UACI (%) |
|---------|----------|----------|
| Lena    | 99.6246  | 33.5234  |
| Peppers | 99.6109  | 33.4788  |
| Cell    | 99.6265  | 33.4929  |
| X-ray   | 99.6147  | 33.4934  |

From Table 9, we can see that the calculated results of both NPCR and UACI are close to their respective theoretical values, indicating that the encryption system has strong ciphertext sensitivity.

5.8. Time Complexity Analysis

The algorithm differs from the traditional compressive sensing algorithm in the use of block compressive sensing, and in the image recovery process, the DDWT-SPL algorithm is used. The encryption and decryption times of Lena and Peppers images of size 512 × 512, with different compression ratios \( CR \), were analyzed; the results are shown in Table 10.

| CR  | Encryption Time (seconds) |
|-----|---------------------------|
| 0.25| 0.34                      |

Table 11 shows the comparison of the encryption time of the algorithm in this paper with other algorithms for the compression ratio \( CR = 0.25 \) (encrypt “Lena” image). It can be seen from Table 11 that the algorithm is faster than other BCS image encryption algorithms [61] when encrypting images of the same size. Compared to the image encryption algorithms based on the chaotic measurement matrix and SHA-256 [16], the image compression–encryption algorithm in this paper is slightly slower.

Next, the time complexity of the encryption scheme proposed in Section 3 was analyzed in detail, assuming that the size of the plaintext image was \( M \times N \), the size of the block compression–aware measurement matrix was \( m \times n \), and the compression ratio was \( CR \). Section 3.1, involved generating the measurement value matrix, and the time complexity of generating the MCLB is \( \Theta(2 \times m \times n) \), the time complexity of generating the measurement matrix is \( \Theta(m \times n) \), and the time complexity of the final quantized mea-
Measurement value matrix is \( \Theta(CR \times M \times N) \). Section 3.2 focused on generating the cipher sequence, where the time complexity of generating the key value was \( \Theta(M \times N) \), the time complexity of generating two cipher sequences was \( \Theta(2 \times CR \times M \times N) \). Section 3.3 involved the encryption of the measurement value matrix, where the time complexity of the permutation of the measurement values was \( \Theta(CR \times M \times N) \), and then we diffused it, with a time complexity of \( \Theta(2 \times CR \times M \times N) \). Then the total time complexity was \( \Theta(2 \times M \times N) \). Compared with the [55,61] encryption algorithms listed in Table 12, our scheme has a smaller time complexity.

### Table 10. Image encryption and decryption time(s) at different compression ratios.

| Images | Lena | Peppers |
|--------|------|---------|
| Process | Encryption | Decryption | Encryption | Decryption |
| CR = 0.25 | 1.007266 | 7.948557 | 0.903141 | 7.736676 |
| CR = 0.5 | 1.344250 | 5.933346 | 1.190280 | 6.051189 |
| CR = 0.75 | 1.640416 | 3.167031 | 1.956602 | 3.195703 |

### Table 11. Image encryption and decryption speed of different schemes at CR = 0.25.

| Algorithm | Time/s |
|-----------|--------|
| Process | Encryption | Decryption |
| Ours | 1.007266 | 7.948557 |
| Reference [61] | 5.380744 | 12.175169 |
| Reference [16] | 0.4115 | 3.5099 |

### Table 12. Comparison results of time complexity.

| Algorithm | Time Complexity |
|-----------|----------------|
| Ours | \( \Theta(2 \times M \times N) \) |
| Reference [55] | \( \Theta(5 \times M \times N) \) |
| Reference [61] | \( \Theta(5 \times M \times N) \) |

### 6. Conclusions

A block compressive sensing image encryption algorithm based on the mixed chaotic Bernoulli measurement matrix is proposed. The core of this encryption algorithm is the combination of the mixed chaotic sequence measurement matrix and BCS, which is practical and can improve the reconstruction quality of images with low compression ratios under the premise of ensuring system security and easy hardware implementation. The compression performance of the mixed chaotic sequence measurement matrix was analyzed, and the performance analysis shows that the measurement matrix has better measurement reconstruction performance than the ordinary measurement matrix. In the encryption process, the SHA-256 function of the plaintext image was used to generate the initial values of the desired chaotic sequence, which improved the ability of the scheme to resist the selective plaintext attack. Finally, the results of the encryption scheme were analyzed and evaluated. First, the effect of reconstructing the image was analyzed; the method has better reconstruction capabilities at a low compression ratio compared with other methods, which could greatly reduce the amount of data to be processed. Secondly, the security of the encryption algorithm was analyzed. The encryption algorithm could resist various attacks from key space, statistical analysis, and information entropy. Finally, the proposed encryption scheme has good encryption effects and image compression capabilities.

**Author Contributions:** C.Y. conceived and wrote the paper. P.P. and Q.D. analyzed the data. Q.D. provided theoretical guidance. P.P. provide guidance for the paper structure. All authors have read and agreed to the published version of the manuscript.
Funding: This research was funded by the National Natural Science Foundation of China under grant number 61471158.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, M.; Xu, M.; Luo, J.; Fan, H. Cryptanalysis of an Image Encryption Using 2D Henon-Sine Map and DNA Approach. *IEEE Access* 2019, 7, 63336–63345. [CrossRef]

2. Chai, X.; Gan, Z.; Yuan, K.; Chen, Y.; Liu, X. A novel image encryption scheme based on DNA sequence operations and chaotic systems. *Neural Comput. Appl.* 2019, 31, 219–237. [CrossRef]

3. Niyat, A.Y.; Moattar, M.H.; Torshiz, M.N. Color image encryption based on hybrid hyper-chaotic system and cellular automata. *Opt. Lasers Eng.* 2017, 90, 225–237. [CrossRef]

4. Hua, Z.; Yi, S.; Zhou, Y. Medical image encryption using high-speed scrambling and pixel adaptive diffusion. *Signal Process.* 2018, 144, 134–144. [CrossRef]

5. Wang, X.; Wang, Q. A novel image encryption algorithm based on dynamic S-boxes constructed by chaos. *Nonlinear Dyn.* 2013, 75, 567–576. [CrossRef]

6. Liu, H.; Kadir, A. Asymmetric color image encryption scheme using 2D discrete-time map. *Signal Process.* 2015, 113, 104–112. [CrossRef]

7. Hu, G.; Li, B. Coupling chaotic system based on unit transform and its applications in image encryption. *Signal Process.* 2021, 178, 107790. [CrossRef]

8. Mansouri, A.; Wang, X. A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. *Inf. Sci.* 2020, 520, 46–62. [CrossRef]

9. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* 2017, 138, 129–137. [CrossRef]

10. Zhu, C.; Sun, K. Cryptanalyzing and Improving a Novel Color Image Encryption Algorithm Using RT-Enhanced Chaotic Tent Maps. *IEEE Access* 2018, 6, 18759–18770. [CrossRef]

11. Wang, X.Y.; Teng, L.; Qin, X. A novel colour image encryption algorithm based on chaos. *Signal Process.* 2012, 92, 1101–1108. [CrossRef]

12. Tsafack, N.; Kengne, J.; Abd-El-Atty, B.; Boulila, W.; Ahmad, J.; Jan, S.U.; Qayyum, A.; Buchanan, W.J. A Lightweight Chaos-Based Medical Image Encryption Scheme Using Random Shuffling and XOR Operations. *Wirel. Pers. Commun.* 2021, 1–28. [CrossRef]

13. Zhu, H.; Wang, X.; Kadir, A. Image Encryption using DNA complementary rule and chaotic maps. *Appl. Soft Comput.* 2012, 12, 1457–1466. [CrossRef]

14. Wei, X.; Guo, L.; Zhang, Q.; Jiang, J.; Lian, S. A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system. *J. Syst. Softw.* 2014, 85, 290–299. [CrossRef]

15. Farah, M.; Farah, A.; Farah, T. An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. *Nonlinear Dyn.* 2020, 99, 3041–3064. [CrossRef]

16. Zhang, Y. The fast image encryption algorithm based on lifting scheme and chaos. *Inf. Sci.* 2020, 520, 177–194. [CrossRef]

17. Naveed, A.A.; Umar, H.; Maria, A.; Hayat, U.; Ayub, M. A Substitution Box Generator, its Analysis, and Applications in Image Encryption. *Signal Process.* 2021, 187, 108144. [CrossRef]

18. Zhang, W.; Zhu, Z.; Yu, H. A symmetric image encryption algorithm based on a coupled logistic-bernoulli map and cellular automata diffusion strategy. *Entropy* 2019, 21, 504. [CrossRef]

19. Ping, P.; Xu, F.; Wang, Z.J. Image encryption based on non-affine and balanced cellular automata. *Signal Process.* 2014, 105, 419–429. [CrossRef]
27. Wang, Y.; Zhao, Y.; Zhou, Q.; Lin, Z. Image encryption using partitioned cellular automata. Neurocomputing 2018, 275, 1318–1332. [CrossRef]
28. Su, Y.; Wo, Y.; Han, G. Reversible cellular automata image encryption for similarity search. Signal Process. Image Commun. 2019, 72, 134–147. [CrossRef]
29. Fan, C.L.; Ding, Q.; Tse, C.K. Counteracting the dynamical degradation of digital chaos by applying stochastic jump of chaotic orbits. Int. J. Bifurc. Chaos 2019, 29, 1930023. [CrossRef]
30. Fan, C.L.; Ding, Q. Analysing the dynamics of digital chaotic maps via a new period search algorithm. Nonlinear Dyn. 2019, 97, 831–841. [CrossRef]
31. Matthews, R. On the derivation of a “Chaotic” encryption algorithm. Cryptologia 1989, 8, 29–41. [CrossRef]
32. Candès, E.J.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory 2006, 52, 489–509. [CrossRef]
33. Donoho, D.L. Compressed sensing. IEEE Trans. Inf. Theory 2006, 52, 1289–1306. [CrossRef]
34. Rachlin, Y.; Baron, D. The Secrecy of Compressed Sensing Measurements. In Proceedings of the 2008 46th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 23–26 September 2008.
35. George, S.N.; Pattathil, D.P. A Secure LFSR Based Random Measurement Matrix for Compressive Sensing. Sens. Imaging 2014, 15, 85.1–85.29. [CrossRef]
36. Gan, H.; Li, Z.; Li, J.; Wang, X.; Cheng, Z. Compressive sensing using chaotic sequence based on Chebyshev map. Nonlinear Dyn. 2014, 78, 2429–2438. [CrossRef]
37. Cambareri, V.; Mangia, M.; Fareschi, F.; Rovatti, R.; Setti, G. Low-Complexity Multiclass Encryption by Compressed Sensing, IEEE Trans. Signal Process. 2015, 63, 2183–2195. [CrossRef]
38. Ahmad, J.; Khan, M.A.; Hwang, S.O.; Khan, J.S. A compression sensing and noise-tolerant image encryption scheme based on chaotic maps and orthogonal matrices. Neural Comput. Appl. 2017, 28, 953–967. [CrossRef]
39. Liu, H.; Liu, Y.B.; Xu, G.X. Securely Compressive Sensing Using Double Random Phase Encoding. Adv. Mater. Res. 2014, 926–930, 3554–3558. [CrossRef]
40. Gilbert, A.; Indyk, P. Sparse Recovery Using Sparse Matrices. Proc. IEEE 2010, 98, 937–947. [CrossRef]
41. Lei, Y.; Barbot, J.P.; Gang, Z. Compressive Sensing with Chaotic Sequence. IEEE Signal Process. Lett. 2010, 17, 731–734. [CrossRef]
42. Ke-Hui, S.; Shao-Bo, H.; Lin-Zi, Y.; Li-Kun, A.; Duo, D.L. Application of FuzzyEn algorithm to the analysis of complexity of chaotic sequence. Acta Phys. Sin. 2012, 61, 130507–379. [CrossRef]
43. Zhou, W.; Jing, B.; Zhang, H.; Huang, Y.F.; Li, J. Construction of measurement matrix in compressive sensing based on composite chaotic mapping. Acta Electron. Sin. 2017, 45, 2177–2183.
44. Luo, Y.; Dang, J.J.; Song, Z.X.; Wang, B.P. Construction and performance research of discrete chaotic measurement matrix. Syst. Eng. Electron. 2020, 42, 749–755.
45. Huang, R.; Rhee, K.H.; Uchida, S. A parallel image encryption method based on compressive sensing. Multimed. Tools Appl. 2014, 72, 71–93. [CrossRef]
46. Zhu, L.; Song, H.; Zhang, X.; Yan, M.; Zhang, L.; Yan, T. A Novel Image Encryption Scheme Based on Nonuniform Sampling in Block Compressive Sensing. IEEE Access 2019, 7, 22161–22174. [CrossRef]
47. Sun, L.; Wen, X.; Lei, M.; Xu, H.; Zhu, J.; Wei, Y. Signal Reconstruction Based on Block Compressed Sensing. In International Conference on Artificial Intelligence and Computational Intelligence; Springer: Berlin/Heidelberg, Germany, 2011.
48. Lin, L. Block Compressed Sensing of Natural Images. In Proceedings of the International Conference on Digital Signal Processing, Cardiff, UK, 1–4 July 2007.
49. Rezaifar, Y.C.P.; Krishnaprasad, P.S. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1–3 November 1993.
50. Dai, W.; Milenkovic, O. Subspace Pursuit for Compressive Sensing Signal Reconstruction. IEEE Trans. Inf. Theory 2009, 55, 2230–2249. [CrossRef]
51. Chen, S.S.; Saunders, D. Atomic Decomposition by Basis Pursuit. Siam Rev. 2001, 43, 129–159. [CrossRef]
52. Mun, S.; Fowler, J.E. Block compressed sensing of images using directional transforms. In Proceedings of the IEEE International Conference on Image Processing, Cairo, Egypt, 7–10 November 2009.
53. Zhang, Y.; Zhang, G.; Zhu, Z. A Fast Algorithm of Compressed Sensing for 2D Signals. IETE Tech. Rev. 2016, 33, 455–465. [CrossRef]
54. Alain, H.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In Proceedings of the 20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23–26 August 2010.
55. Gan, Z.; Chai, X.; Zhang, J.; Zhang, Y.; Chen, Y. An effective image compression–encryption scheme based on compressive sensing (CS) and game of life (GOL). Neural Comput. Appl. 2020, 32, 14113–14141. [CrossRef]
56. Wang, X.; Su, Y. Image encryption based on compressed sensing and DNA encoding. Signal Process. Image Commun. 2021, 12, 116246. [CrossRef]
57. Wen, W.; Hong, Y.; Fang, Y.; Li, M.; Li, M. A Visually Secure Image Encryption Scheme based on Semi-Tensor Product Compressed Sensing. Signal Process. 2020, 173, 107580. [CrossRef]
58. Wang, H.; Xiao, D.; Li, M.; Xiang, Y.; Li, X. A visually secure image encryption scheme based on parallel compressive sensing. *Signal Process.* 2019, *155*, 218–232. [CrossRef]
59. Mirzaei, O.; Yaghoobi, M.; Irani, H. A new image encryption method: Parallel sub-image encryption with hyper chaos. *Nonlinear Dyn.* 2012, *67*, 557–566. [CrossRef]
60. Zhu, C.; Wang, G.; Sun, K. Cryptanalysis and Improvement on an Image Encryption Algorithm Design Using a Novel Chaos Based S-Box. *Symmetry* 2018, *10*, 9. [CrossRef]
61. Shi, Y.; Hu, Y.; Wang, B. Image Encryption Scheme Based on Multiscale Block Compressed Sensing and Markov Model. *Entropy* 2021, *23*, 1297. [CrossRef]