applied sciences

MDPI

Article

Hash Based DNA Computing Algorithm for Image Encryption

Hongming Li [1], Lilai Zhang [2,3], Hao Cao [2,3] and Yirui Wu [2,3,4,5,*]

[1] Electronic Information Engineering, Ningbo Polytechnic, Ningbo 315000, China
[2] Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing 211100, China
[3] College of Computer and Information, Hohai Univerisity, Nanjing 211100, China
[4] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[5] Nanjing Hezehaichuan Technology Company, Nanjing 211100, China
* Correspondence: wuyirui@hhu.edu.cn

Abstract: Deoxyribonucleic Acid (DNA) computing has demonstrated great potential in data encryption due to its capability of parallel computation, minimal storage requirement, and unbreakable cryptography. Focusing on high-dimensional image data for encryption with DNA computing, we propose a novel hash encoding-based DNA computing algorithm, which consists of a DNA hash encoding module and content-aware encrypting module. Inspired by the significant properties of the hash function, we build a quantity of hash mappings from image pixels to DNA computing bases, properly integrating the advantages of the hash function and DNA computing to boost performance. Considering the correlation relationship of pixels and patches for modeling, a content-aware encrypting module is proposed to reorganize the image data structure, resisting the crack with non-linear and high dimensional complexity originating from the correlation relationship. The experimental results suggest that the proposed method performs better than most comparative methods in key space, histogram analysis, pixel correlation, information entropy, and sensitivity measurements.

Keywords: DNA computing; image encryption; context-aware DNA permutation; hash DNA encoding

Citation: Li, H.; Zhang, L.; Cao, H.; Wu, Y. Hash Based DNA Computing Algorithm for Image Encryption. *Appl. Sci.* **2023**, *13*, 8509. https://doi.org/10.3390/app13148509

Academic Editor: Krzysztof Koszela

Received: 3 March 2023
Revised: 21 June 2023
Accepted: 20 July 2023
Published: 23 July 2023

1. Introduction

With the breakthrough development of information technology, how to process data with secured and rapid characteristics has become a major concern [1]. Inspired by DNA structure, a novel computing paradigm using a biological molecule to carry genetic information is designed to ensure the feasibility of computing at a molecular level with impressive characterizations of programmability and high-throughput coding, which not only males it possible to replace traditional silicon-based facilities by biological tools [2], but also provides an alternative way to consider computing with more than 0 and 1. Therefore, DNA computing [3], as a popular computing model with considerable potential to meet the security requirement, has nowadays become a hot spot in the data processing domain.

In hardware-based data processing design, DNA logic gates are designed in combination with digital circuits, acting as basic components to form complex DNA circuits. For example, Zhang et al. [4] propose an entropy-driven incision-assisted recovery strategy for reactants in the DNA loop, which can recover reactants in the catalytic loop and improve their recoverability, creating more effective DNA circuits for molecular transformation and synthetic biology. Unlike most artificially catalyzed DNA loops, Yang et al. [5] develop a catalytic DNA logic circuit regulated by DNase, which is controlled by a covalent modification strategy and demonstrates a great potential in more complex computing by building complex cascade circuits.

In algorithm-based data computing design, molecular programming is built on the basis of DNA circuits with powerful modularity. For example, Zhang et al. [6] design a DNA molecular computing platform to analyze miRNA profiles in serum samples, achieving

an intelligent diagnosis of cancer. Later, Ma et al. [7] design and implement various types of a DNA computing system, which verifies the feasibility of using DNA computing for intelligent diagnoses in fields of biology and biomedicine.

Despite various designs of clinical applications, DNA computing is popular for building a encryption application, due to its reliable encryption performance and parallel processing capability. Specifically, we could roughly divide encryption methods into two types, i.e., diffusion encryption and permutation encryption. Diffusion encryption refers to the process of spreading information throughout the ciphertext, thus increasing unbreakability of the encryption algorithm. More precisely, diffusion encryption maps each bit of the plaintext to multiple bits in the ciphertext. Any change in a plaintext bit should correspond to variations in multiple bits of the ciphertext, successfully achieving a correlation between the plaintext and ciphertext. Permutation encryption encrypts the plaintext by replacing each bit with a different bit, which reorders the information of the plaintext to hide content, instead of altering values of each bit in the plaintext. The key process of the permutation encryption is to establish a one-to-one mapping between each bit of the plaintext and ciphertext, thus achieving a confusion effect of information.

In fact, the quantity of algorithms have been proposed for data encrypting based on DNA computing. For example, Khan et al. [8] propose DNAact Ran, a DNA computing-based sequence analysis engine, which could accurately detect a ransomware attack with their designed constraints and kmer frequency vector. Combining DNA computing with chaotic systems, Zou et al. [9] involve the DNA-based strand displacement strategy with the chaotic system, where the generated chaotic sequence greatly improves the unbreakability of their encryption system, owing to its capability to stand with statistical attacks. Most recently, Namasudra et al. [10] propose a DNA computing-based access control algorithm, where 1024 random keys are fed into the DNA computing system for the user's secret data encryption, significantly improving the security capability of the control model.

Inspired by the idea of utilizing DNA computing to encrypt, we aim to encrypt high-dimensional image data in this paper. Compared with text structure data, the image is characterized with high-dimensional and unstructured proptoses, which have two major difficulties in designing proper encryption algorithms. **First**, how to achieve an unbreakable capability for image data with less encoding complexity, since a more complex encoding strategy generally promotes encryption performance. **Second**, since the image is equipped with a natural property of high-dimensional complexity, how algorithms could involve such a property for for encoding. In other words, images are a natural source of randomness, where adjacent pixels marked with a considerable amount of redundant information demonstrate strong correlations. Therefore, the characteristic of smooth variation between pixels can be leveraged to speed up the image encryption process.

To address the above difficulties, we develop a novel hash encoding-based DNA computing algorithm to effectively encrypt high-dimensional image data, which consists of the DNA hash encoding module and content-aware encrypting module. The hash algorithm is unique in encrypting, which outputs a fixed-length output with data as complicated as possible. Moreover, it shares several impressive characteristics, such as a small computing burden with a simple function calculation; unidirectionality without possible ways to reverse the input data through hash results, tampering resistance where small modification would greatly vary the output results; and anti-collision capability, which guarantees that the unique output could be gained with different inputs. Inspired by these properties, we involve hash encoding into DNA computing inside the proposed DNA hash encoding module, thus boosting the encryption capability of high-dimensional data with a small computation burden. Considering high-dimensional image data as a natural source to encode with less complexity, a content-aware encrypting module is proposed to map between image content and encryption results with several simple but effective functions, thus offering new ways to encrypt images based on DNA computing.

To sum up, the contributions of this paper are three-fold:

- The proposed method combines the impressive characteristics of DNA computing and the hash function to realize an unbreakable image encryption with less computation burden.
- A novel DNA-based hash encoding module is proposed, which involves the hash function to construct mappings from high-dimensional image data to DNA bases.
- Considering correlations between adjacent pixels as natural sources of complexity, the proposed content-aware encrypting module sunccesfully generates random DNA sequences with chaos properties, which adopts non-linear functions originating from correlations of pixles as source of complexity.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 first presents an overview on algorithm structure, and then presents detailed steps of encryption and decryption. Section 4 conducts several experiments, including an analysis on the key space, histogram, pixel correlation, information entropy, sensitivity and computation cost. Finally, Section 5 concludes the paper and demonstrates the prospect.

2. Related Work

In this section, we provide a brief literature review of this paper, including an introduction to DNA computing and DNA computing-based image encryption.

2.1. Introduction to DNA Computing

With the development of DNA computing, the DNA strand displacement reaction has gradually become an important means to build complex digital circuits operating at room temperature. To solve the limitations of DNA logic circuits, i.e., slow computing speed and complex circuit design, Song et al. [11] propose a DNA logic circuit structure based on single strand logic gates, which improves the computing speed of DNA circuits and the number of DNA strands based on the biological performance of the strand replacement DNA polymerase. Combined with the commonly used cascading strategy, their proposed structure could aid to build large-scale logic circuits.

To build the DNA-based switch circuit, Wang et al. [12] developed a modular DNA molecular switch for the digital operation of any function, which solves the limitation of operation speed and signal-to-noise ratio comparing with the traditional logic gate circuit. In their experiments, their proposed DNA-based switch circuit not only controls the current signal transmission through the switch signal, but also controls the current signal transmission direction through the difference of molecular free energy in the reaction path. Essentially, the development of the DNA switch circuit has laid a solid foundation for molecular computers with decision-making abilities. Therefore, Thubagere et al. [13] successfully developed DNA-based robots, which could autonomously transport molecular-level goods to a designated location without any energy supply.

To improve the security of the physical layer, DNA computing is designed for data encryption during transmission. For example, Liu et al. [14] propose an image encryption scheme, where the DNA-based image compression are used to ensure the security of transmission. Later, Xiao et al. [15] propose a chaotic OFDM (Orthogonal Frequency Division Multiplexing) hybrid security method to ensure the security of the physical layer, which not only dynamically adjusts parameters of DNA-based encoding rules and scrambling methods, but also controls the positional and traversal direction of the helix through chaotic sequences.

2.2. DNA Computing-Based Image Encryption

DNA computing builds the encryption system based on modular arithmetic cryptography. Moreover, a chaotic strategy has been proved to be well matched with DNA ciphers, due to its characteristics of pseudo randomness, unpredictability and so on [16]. For example, Babaei et al. [17] realize a theoretically unbreakable data encryption algorithm, where they build a chaos strategy based on the difference between the original and optimized DNA-based encryption messages. Later, Wang et al. [18] first use piecewise linear chaotic

mapping (PWLCM) and logistic mapping to generate encryption parameters, and then use DNA computing for information encoding. Specifically, their PWLCM could make each small segment of the original data be paired with an element of the pseudo-random sequence, thereby increasing the complexity of the encryption algorithm. Afterwards, Samiullah et al. [19] use three chaotic systems including PWLCM, Lorenz and 4D Lorenz to build a multi-level security system, which could largely increase the non-repeatability of the key.

To reduce the computational cost of image encryption, Malik et al. [20] use a tent map to select key streams generated by their proposed confusing and diffusing channels. Later, Khan et al. [21] propose a DNA-based image encryption method to effectively reduce memory usage, which selects the most informativepart from the visual appearance analyzed by their proposed dependent chaotic system. Afterwards, Ravichandran et al. [22] propose a medical image encryption algorithm based on the combination of the integer wavelet transform (IWT), DNA and chaos system. Their strategy of utilizing the advantages of different encryption methods ensures the security of electronic health records during network transmission. Recently, Zhang et al. [23] proposed a multi-image encryption algorithm, which not only resists conventional attacks to protect image content, but also improves the transmission speed via Internet by reducing image size by encoding. Most recently, Wu et al. [24] designed a random encoding, sequencing and diffusion strategy based on content-aware DNA computing, which greatly improves the difficulty of cracking even if attackers obtain partial or all information of the transmitted image. Most relevant to our method, Chen et al. [25] propose an image encryption scheme that incorporates adaptive diffusion permutation and DNA random coding. We further improve their idea by involving the hash function and image content-based encoding, thus greatly improving the capability to resist different types of attacks.

3. Methods

We provide a detailed description of the proposed method with three parts, i.e., the overall framework, DNA hash encoding module and content-aware encryption module.

3.1. Overall Framework

We begin with an introduction to a basis of DNA computing in encryption. DNA bases are classified into four types, namely, adenine (A), thymine (T), cytosine (C) and guanine (G). It is worth noting that the former two types are complementary pairs, and so are the latter two types. We then describe rules for DNA-based sequence encoding, which uses a binary encoding idea to represent the input sequence under rules shown in Table 1 with four bases, i.e., A, C, G and T. With the aid of encoding rules of Table 1, we can convert binary sequence into a DNA-based encoding form for further computation. For example, under rule 2, A, C, G and T are represented as 00, 01, 10, and 11, respectively. If a pixel in the input image refers to 179 in gray levels and its binary representation is 10110011, its corresponding DNA-based encoding sequence should be G-T-A-T under rule 2. During decoding, if a DNA-based encoding sequence is G-A-T-C under rule 2, its corresponding binary value should be 10001101.

Table 1. DNA-based binary encoding rules, which uses different rules to represent binary values with DNA bases.

	Rule 1	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**	**Rule 6**	**Rule 7**	**Rule 8**
A	00	00	11	11	10	01	10	01
C	10	01	10	01	00	00	11	11
G	01	10	01	10	11	11	00	00
T	11	11	00	00	01	10	01	10

We then demonstrate the overall framework in Figure 1, which consists of sender and receiver structures. Specifically, the whole process of encryption inside the sender can be described as follows:

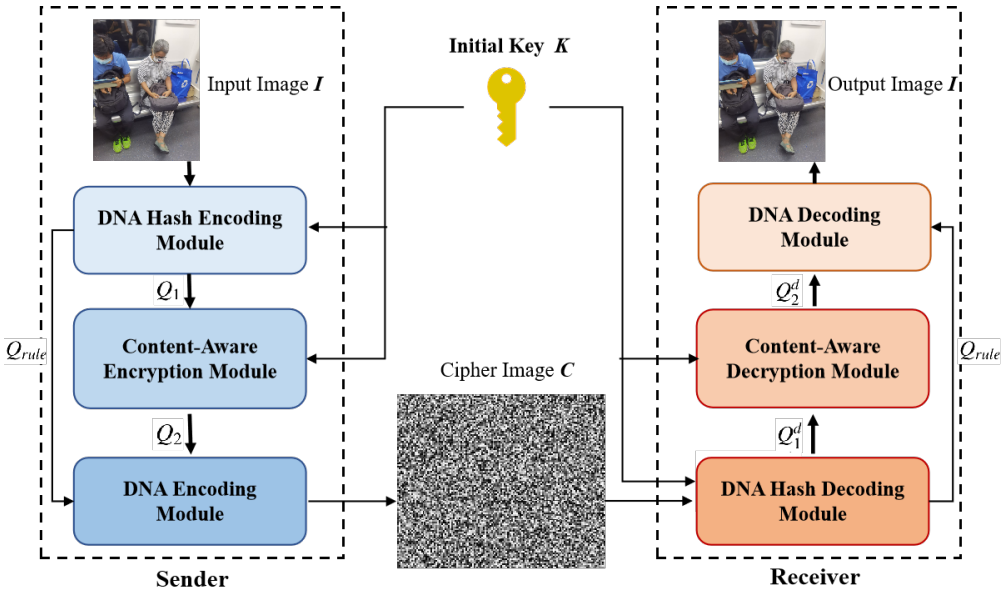

Figure 1. The overall structure of the proposed method, including sender and receiver. They both contain a DNA hash encoding module and a DNA decoding module. In fact, content-aware decryption module can be regarded as the inverse version of content-aware encryption module.

Step 1. Feed the initial key K and the input image I into the DNA hash encoding module, which computes the rule selection sequence Q_{rule} and DNA-encoded Sequence Q_1.

Step 2. K and Q_1 are regarded as input of Content-aware Encryption module, which computes new sequence Q_2 with permutation and diffusion, thus encoding complexity for high unbreakability.

Step 3. Q_{rule} and Q_2 are fed into the DNA encoding module, which encodes Q_2 by utilizing Q_{rule} to generate a binary bitstream under DNA encoding rules. Afterwards, we transfer the generated bitstream as a ciphered image C with $H^{'} \times W^{'}$.

Step 4. Transmit C from the sender to the receiver via Internet.

The whole process of decryption inside the receiver can be described as follows:

Step 5. Regarding C as input, the receiver sends it to DNA hash decoding module with the initial key K, thus generating the rule selection sequence Q_{rule} and a DNA-based decoding sequence Q_1^d. It is noted that we keep the rule selection sequence exactly the same to ensure the consistency of decryption content.

Step 6. Feed K and Q_1^d into the content-aware decryption module for the reversed permutation and diffusion, thus generating a reversed DNA-based decoding sequence Q_2^d.

Step 7. Feed Q_2^d and Q_{rule} into the DNA decoding module, which uses Q_{rule} as the rule selector to decode Q_2^d into the binary bitstream I_{bit}. Finally, we transfer I_{bit} into the output image I for computation. Note that the input and output image should remain the same due to the general principles of encryption.

3.2. Design of DNA Hash Encoding Module

The DNA hash encoding module is designed to transform image data into a DNA-encoded form with different DNA bases, which could involve significant properties of the hash function, i.e., fixed-length output, small computing burden, tampering resistance and anti-collision ability. Such property help improve the unbreakability of the proposed method with low computing cost.

As shown in Table 1, there only exist eight encoding patterns that satisfy the "complementary" rule, where each two bits satisfys the condition that there XOR results should be true. For example, if 00 and 01 are encoded as A and C, respectively, 11 and 10 should be encoded as T and G, respectively. There are only eight encoding rules obtained through this combination, and they are all listed in Table 1. Based on the analysis of the "comple-

mentary" rule, we convert image data into a bit stream with the size of 3 bits to describe 8 different mapping rules, where the 2 bit is mapped as one DNA base for the encoding basis. Meanwhile, two more 8-bit streams are designed at the beginning of the stream to record the length and width of the original image, which could help restore the input image after encryption in a proper way.

Essentially, the proposed DNA hash encoding module plays a crucial role in encryption, which randomly employs various sets of complementary rules to encode each pixel. We demonstrate the overall encoding process of it during encryption in Figure 2. The detailed steps of the encoding and decoding process inside the DNA hash encoding module are also listed with pseudo codes in Algorithms 1 and 2. First, we adopt the SHA256 algorithm to generate two important parameters, i.e., initial key value and initializing factor. To effectively improve the security of the cryptosystem, we then feed the computing factor p into the PWLCM, generating the rule selection sequence with the following equation:

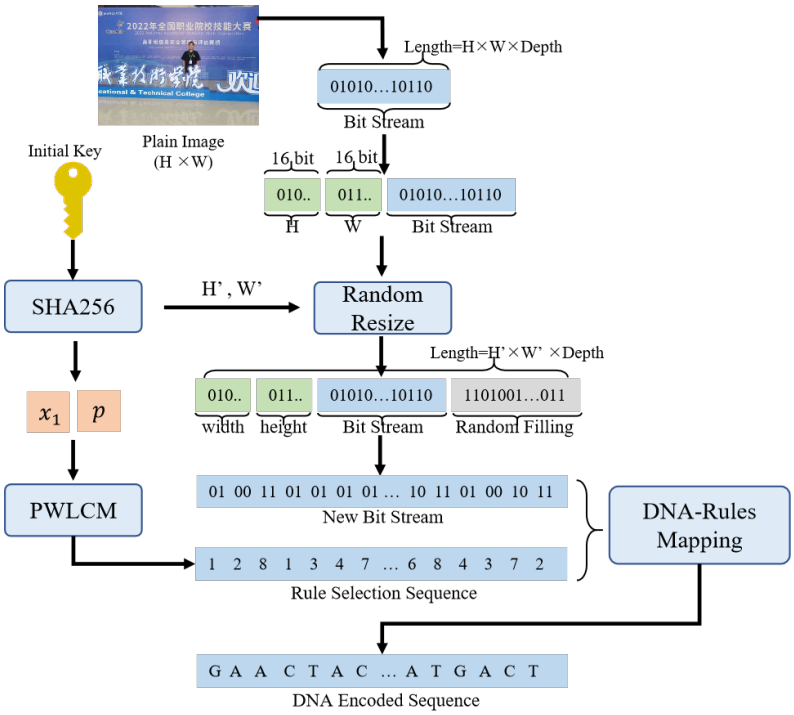

Figure 2. Encoding process inside DNA hash encoding module.

$$x_{n+1} = \begin{cases} \dfrac{x_n}{p}, & 0 \le x < p \\ \dfrac{x_n - p}{0.5 - p}, & p < x \le 0.5 \\ F(1 - x_n, p), & 0.5 < x \le 1 \end{cases} \tag{1}$$

It's noted that $p \in (0, 0.5]$, which enables variety in outcome with little modification in key values, whi ch is highly resistant to attack. Finally, we use the rule selection sequence to help encode the bitstream, which serves as DNA rules to encode pixels with DNA-based rules as shown in Table 1.

Algorithm 1: DNA hash encoding process

Input: Initial Key K, Plain Image I
Output: Rule Selection Sequence Q_{rule}, DNA encoded Sequence Q_1

$H, W =$ original height and original width of I;
Randomly generate $H^{'}$ and $W^{'}$, where $H < H^{'} < 2 \times H$ and $W < W^{'} < 2 \times W$;
Convert the integer numbers $H^{'}$ and $W^{'}$ into a l6 bit binary bitstream HW_{bit} ;
Convert the plain image I into a binary bitstream I_{bit};
Randomly generate a binary bitarray R_{bit} whose length is $H^{'} \times W^{'} \times 8 - H \times W \times 24 - 16$;
$I_{bit} = HW_{bit} \oplus I_{bit} \oplus R_{bit}$ (where $\oplus$ indicates concatenate operation);
$length =$ Length(I_{bit});
$HASH_K =$ SHA256(K);
$H_1 = HASH_K(0:63)$;
$H_2 = HASH_K(64:127)$;
$x_1 = modf\left(H_1/10^{15}\right)$;
$p = modf\left(H_2/10^{15}\right)/2$;
For (t = 0 **to** $length/2$);
    $x_{t+1} = PWLCM(x_t, p)$;
$X = [x_1, x_2, ..., x_n]$;
$Q_{rule} = mod\left(floor\left(\boldsymbol{X} \times 10^{15}\right), 8\right)$;
Introduce the DNA encoding table T ;
For(t = 0 **to** $length/2$);
    $Q_1(t) = T(Q_{rule}(t), I_{bit}(2t, 2t+1))$;
Output Rule Selection Sequence Q_{rule}, DNA encoded sequence Q_1;

Algorithm 2: DNA Decoding Process

Input: Rule Selection Sequence Q_{rule}, DNA Encoded Sequence Q_2
Output: Cipher Image C

$length =$ Length(Q_2);
$T =$ DNA Encoding Table;
For (t = 0 **to** $length$);
    $C_{bit}(2t, 2t+1) = T(S_{rule}(t), S_2(t))$;
Change cipher bit array C_{bit} into cipher image C;
Output Cipher Image C;

For readers' convenience, we further offer a step-by-step explanation of hash encoding in Figure 3, which performs an encryption process on a sample image, as follows:

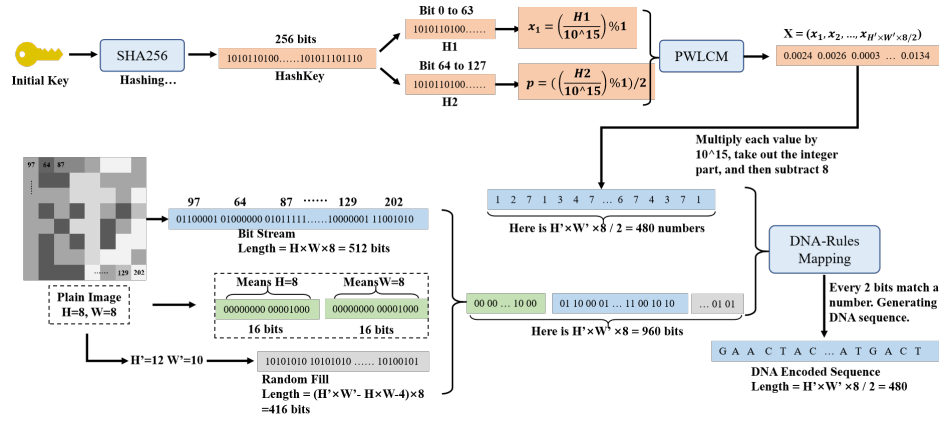

Figure 3. An example of DNA hash encoding by inputting an image with size 8 × 8.

Step 1. Obtain the height and width of the original image, where H and W equals 8 for the sample image. Convert the original image to a bitstream containing 512 bits of data.

Step 2. Save the original height and width as 16-bit data. For example, if H is 8, its corresponding bitstream is 00000000 00001000. Therefore, we use 32 bits of data to save the original height and width for further restoration.

Step 3. Randomly generate a new height and width, where both values should be larger but twice as small as the the original ones. For display, we set the new height H' and width W' to 12 and 10, respectively.

Step 4. It is noted that there exist $H' \times W' \times 8 = 960$ bits of data for encoding, where the first 32 bits represent the original height and width, and the other 512 bits refer to image content. It's noted that 416 bits remain unused, where they are randomly filled to make up a total 960-bit bitstream for further computation.

Step 5. We generate a 256-bit bit sequence HashKey through the SHA256 hash algorithm based on the input initial key. Note that $H1$ and $H2$ can be viewed as 64-bit numbers for computing.

Step 6. With $H1$ and $H2$, we calculate $x_1 = (H1/10^{15})\%1$ and $p = ((H1/10^{15})\%1)/2$.

Step 7. Then, x_1 and p are fed into the PWLCM function to generate a sequence, where $(H' \times W' \times 8)/2 = (12 \times 10 \times 8)/2 = 480$ numbers are generated. Each of them is multiplied by 10^{15}, modulared with 8, achieving results with a rule sequence containing 480 numbers.

Step 8. We further match 960 bits of original data with rule sequence, where we match every 2 bits of data with 1 rule number based on DNA mapping rules in Table 1. After converting, we could obtain results as a DNA hash-encoded sequence, which would be sent to the content-aware encryption module for permutation and diffusion.

3.3. Content-Aware Encryption Module

The content-aware encryption enhancing module is designed to reorganize the data structure by highly non-linear functions, originating from correlations between adjacent pixels and patches. It is noted that DNA operations could vary to form more diversified expressions, thus greatly improving the variability of encrypted sequences. Moreover, the inherent complexity of DNA operations can be enhanced by utilizing the relevance between neighboring pixels, borrowing non-linear equations from neighboring pixels and patches. Therefore, the proposed module is designed to naturally involve the complexity of the image, greatly boosting the difficulty of the cracking.

We demonstrate the structure design of the proposed module in Figure 4, where the DNA-encoded sequence and rule selection sequence are regarded as inputs for the module. Essentially, we design a reversible permutation and diffusion algorithm to combine both forms of complexity for better unbreakability. Both algorithms could be directly applied on the input image without additional transmission cost via Internet or LAN. Furthermore, we have designed several computation operations that satisfy the commutative law, namely ADD, SUB and XOR, which not only ensures the variance of DNA sequences, but also shares the same parameters to reduce unnecessary transmission costs in LAN.

Specifically, we describe the calculation process of the DNA Sequence Permutation algorithm in Algorithm 3, which is built on the following four parameters:

$$\begin{cases} \frac{dx_1}{dt} = x_{n+1} = a(y_n - x_n) + w_n \\ \frac{dy_1}{dt} = cx_n - y_n - x_n z_n \\ \frac{dz_1}{dt} = x_n y_n - b_n z_n \\ \frac{dw_1}{dt} = -y_n z_n + \gamma w_n \end{cases} \tag{2}$$

where a, b, c and γ are preset parameters equaling 10, 3/8, 28 and $[-1.52, -0.06]$, respectively. It is noted that such a calculation process ensures the chaos property of the

proposed module. Afterwards, four parameters are sent into the Hyper Chaotic Lorenz System (HCLS) to generate the permutation control sequence Q_p, where HCLS is a highly nonlinear dynamical system, being sensitive to the initial values for unpredictability.

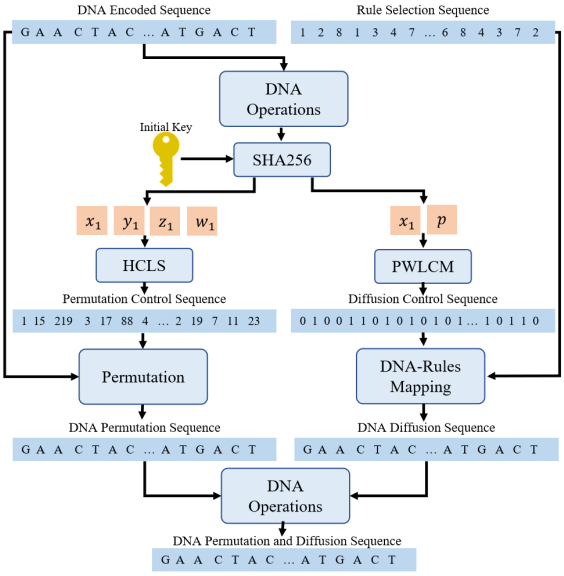

Figure 4. Structure design of the proposed content-aware encryption module.

Algorithm 3: DNA Sequence Permutation Process

Input: Initial Key K, DNA Encoded Sequence Q_1
Output: DNA Permutation Sequence Q_1^p

$TA =$ DNA ADD Table;
$TX =$ DNA XOR Table;
$length =$ Length(Q_1);
$AR = Q_1(0); XR = Q_1(0)$;
For (i = 1 **to** $length$);
    $AR = TA(AR, Q_1(i))$;
    $XR = TX(XR, Q_1(i))$;
$HASH_D =$ SHA256$([AR, XR])$;
$HASH_K =$ SHA256(K);
$HASH_{DK} = HASH_D \oplus HASH_K$;
$A_1 = HASH_{DK}(0:63); A_2 = HASH_{DK}(64:127); A_3 = HASH_{DK}(128:191)$;
$A_4 = HASH_{DK}(192:255)$;
$x_1 = \left(\mathrm{mod}\left(\mathrm{fix}\left(A_1/10^8\right), 80\right) - 40\right) + \left(A_1/10^{14} - fix\left(A_1/10^{14}\right)\right)$;
$y_1 = \left(\mathrm{mod}\left(\mathrm{fix}\left(A_2/10^8\right), 80\right) - 40\right) + \left(A_2/10^{14} - fix\left(A_2/10^{14}\right)\right)$;
$z_1 = \left(\mathrm{mod}\left(\mathrm{fix}\left(A_3/10^8\right), 80\right) + 1\right) + \left(A_3/10^{14} - fix\left(A_3/10^{14}\right)\right)$;
$w_1 = \left(\mathrm{mod}\left(\mathrm{fix}\left(A_4/10^8\right), 500\right) - 250\right) + \left(A_4/10^{14} - fix\left(A_4/10^{14}\right)\right)$;
Put x_1, y_1, z_1, w_1 into HCLS to generate a sequence $X = [x_1, x_2, ..., x_n]$ by iterating;
$Q_p = mod\left(floor\left(\boldsymbol{X} \times 10^{15}\right), length\right)$;
If (This is the encryption process);
    $i = 0, j = length/2$;
Else;
    $i = length/2, j = 0$;
For ($k = i$ **to** j);
    $Q_1(Q_p(k)) \leftrightarrow Q_1(Q_p(length - k))$;
$Q_1^p = Q_1$;
Output DNA Permutation Sequence Q_1^p;

Similarly, we describe the calculation process of the DNA Sequence diffusion algorithm in Algorithm 4. Specifically, we first calculate key parameters X_1 and p based on values of H_1 and H_2 transmitted from the last module. Then, we input these parameters into the PWLCM algorithm to generate a diffusion control sequence. With the input rule selection sequence, we adopt DNA rules to map from numbers to DNA bases, thus generating DNA diffusion sequence. Finally, we apply DNA-based calculations on the generated permutation sequence and diffusion sequence, thus obtaining the merged sequence as the final output of the DNA permutation and diffusion sequence.

Algorithm 4: DNA Sequence Diffusion Process

Input: Initial Key K, Rule Selection Sequence Q_{rule}, Permutated Sequence Q_1^p
Output: Permutated and Diffused Sequence S_2

Convert the plain image I into a binary bitstream I_{bit};
length = Length(S_p);
$HASH_K$ = SHA256(K);
$H_1 = HASH_K(128:191)$; $H_2 = HASH_K(192:255)$;
$x_1 = modf\left(H_1/10^{15}\right)$;
$p = modf\left(H_2/10^{15}\right)$;
For (t = 0 **to** $length/2$);
    $x_{t+1} = PWLCM(x_t, p)$;
$X = [x_0, x_1, ..., x_{length/2}]$;
$Y = mod\left(floor\left(\boldsymbol{X} \times 10^{15}\right), 256\right)$;
T =DNA Encoding Table;
For (i = 0 **to** $length/2$);
    $S_{key}(i) = T(S_{rule}(i), Y(2i, 2i+1))$;
TA = DNA ADD Operation Table;
TS = DNA SUB Operation Table;
TX = DNA XOR Operation Table;
If (in encryption process);
    $D(0) = TA(Q_p(0), S_{key}(0))$;
    $D(0) = TX(D(0), S_{key}(0))$;
    For (i = 0 **to** $length$);
        If mod$(i, 2) == 1$;
            $D(i) = TX(Q_p(i), S_{key}(i))$; $D(i) = TX(D(i), D(i-1))$;
        Else;
            $D(i) = TA(Q_p(i), S_{key}(i))$; $D(i) = TX(D(i), D(i-1))$;
Elif (in decryption process);
    For (i = $length$ **to** 0);
        If mod$(i, 2) == 1$;
            $D(i) = TX(Q_p(i), S_{key}(i))$; $D(i) = TX(D(i), Q_p(i-1))$;
        Else;
            $D(i) = TA(Q_p(i), Q_p(i))$; $D(i) = TX(D(i), S_{key}(i-1))$;
    $D(0) = TA(Q_p(0), S_{key}(0))$;
    $D(0) = TX(D(0), S_{key}(0))$;
$Q_2 = D(0 : length - 1)$ Output Permutated and Diffused Sequence Q_2;

4. Experiment and Analysis

We firstly introduce the experiment settings and implementation details. Then, we perform variants of the analysis to verify the resistance against different attacks. Afterwards, we perform ablation and computation analysis to evaluate performance. Finally, the comparison experiments are conducted.

4.1. Experiment Settings and Implementation Details

We deploy the proposed encryption system in TCP/IP environment. We connected two computers through the Tenda-AC71200M router, thus forming a simple wireless LAN environment. Specifically, the original image is provided by a laptop (sender) and encrypted into a cipher image by the proposed method. Then, the cipher image is transmitted to another laptop (receiver) via IP Messenger protocol and decrypted into a plain image.

It is noted that we use two computers with the same configurations to simulate the sender and receiver. Both computers are equipped with an Intel Core i5-12400F CPU, NVIDIA RTX 3060Ti GPU, and 16GB memory. Additionally, the encryption method is implemented based on Python 3.8, and the initial key is set to 'GOOD-LUCK'.

4.2. Key Space Analysis

Key space is defined as the size of a key range, i.e., the number of keys, where different encryption algorithms have their own specific key space. Essentially, we should ensure that the key space is large enough to withstand brute force attacks. Specifically, we can calculate that the size of the key space of the proposed method is $S = (0.5 \times 10^{15})^2 \times (1 \times 10^{15})^2 \times (80 \times 10^{14})^3 \times (500 \times 10^{14}) = 6.410^{127} \approx 2^{418}$. Obviously, such a large key space ensures the capability to resist brute force attacks.

4.3. Histogram Analysis

The statistical analysis attack refers to the fact that attackers can obtain a quantity of information by analyzing the distribution of image pixel values. To verify the effectiveness of the proposed method against the statistical analysis attack, we compare the pixel value distribution of the original and cipher image. As shown in Figure 5a–e, the histogram of the original image fluctuates greatly, where attackers can easily construct effective attack strategies. Figure 5f–j shows the corresponding cipher images and pixel histograms, which proves that the encrypted image can effectively hide image information to resist statistical analysis attacks.

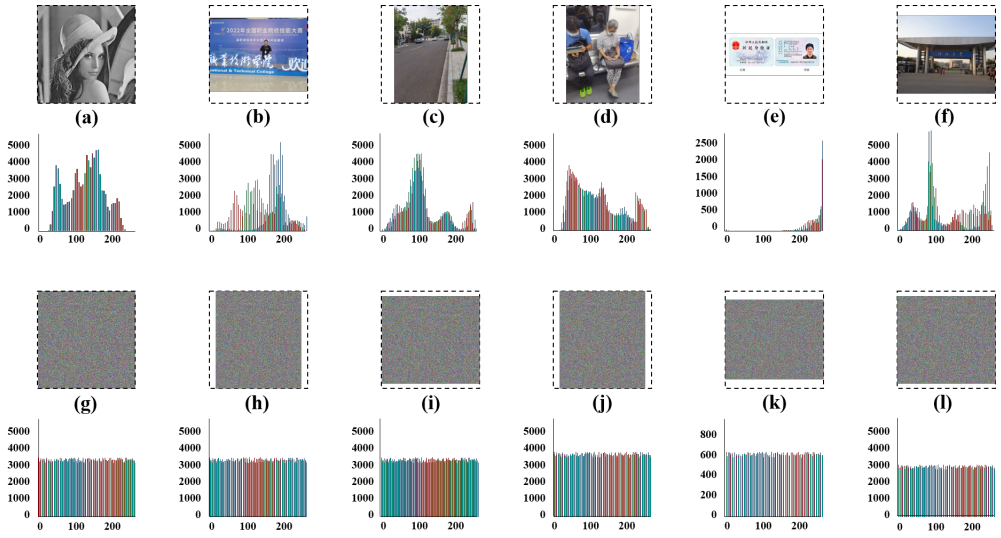

Figure 5. The corresponding histograms for different plain and cipher images, where (**a**–**f**) are input plain images of 'Lena', 'Poster', 'Street', 'Subway', 'ID Card' , 'Hohai University', and (**g**–**l**) are their corresponding cipher images.

4.4. Pixel Correlation Analysis

Permutation and diffusion operations are used to scramble adjacent pixels between images, thus reducing the correlation between adjacent pixels and improving the security

after image encryption. We can calculate the correlation coefficient between a and b to evaluate such correlation effects

$$\begin{cases} r_{ab} = \frac{\mathrm{cov}(a,b)}{\sqrt{D(a)}\sqrt{D(b)}} \\ \mathrm{cov}(a,b) = \frac{1}{N}\sum_{i=1}^{N}(a_i - E(a))(b_i - E(b)) \\ D(a) = \frac{1}{N}\sum_{i=1}^{N}(a_i - E(a))^2 \\ E(a) = \frac{1}{N}\sum_{i=1}^{N} a_i \end{cases} \tag{3}$$

where $\mathrm{cov}(a,b)$, $D(a)$ and $E(a)$ are defined as the covariance, mean square and expected error, respectively.

Table 2 shows the correlation coefficients of the plain and cipher image, where we can observe that the pixel correlation has been significantly reduced by comparing between both images. Correspondingly, Figure 6 shows the comparison of image correlation before and after encryption, which proves that the proposed method can effectively reduce the correlation in three directions.

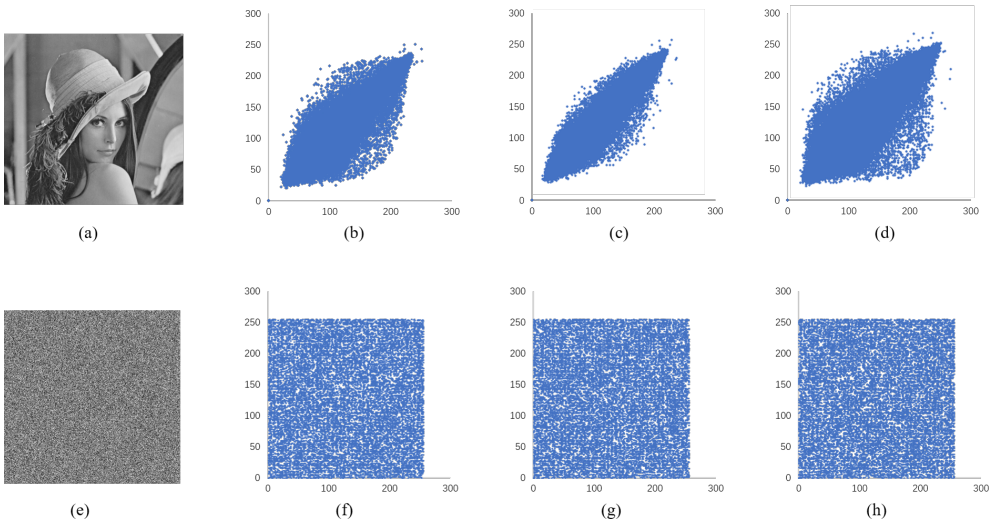

Figure 6. Correlation coefficients distributions of 'Lena' (**a**) and its corresponding cipher image (**e**), where (**b**,**f**) are distributions in horizontal direction, (**c**,**g**) are distributions in vertical direction, and (**d**,**h**) are distributions in diagonal direction.

Table 2. Results of correlation coefficients analysis on several testing images.

Images	Plain Image			Cipher Image		
	V	H	D	V	H	D
Baboon	0.8813	0.7524	0.7638	0.0039	0.0037	0.0048
Bridge	0.9344	0.9531	0.9812	−0.0089	0.0049	−0.0017
Chest	0.9674	0.9892	0.9816	−0.0037	0.0045	−0.0081
Lena	0.9832	0.9729	0.9631	0.0017	0.0026	0.0033
Hohai University	0.8681	0.8724	0.9076	−0.0018	−0.0089	0.0094

4.5. Information Entropy Analysis

Information entropy is used to measure the uncertainty of data, where higher information entropy refers to larger complexity for cracking. Supposing the number of keys of an encryption algorithm is K-bit, the ideal information entropy should be K. To evaluate the uncertainty of the data, i.e., the diffusion performance, information entropy can be calculated as follows:

$$H = -\sum_{i=1}^{256} p(e_i)\log_2 p(e_i) \tag{4}$$

where $p(e_i)$ represents the probability of the pixel being i.

The information entropy of five test images is 7.987421 (Poster), 7.996523 (Street), 7.989923 (Subway), 7.998615 (ID Card) and 7.988991 (Hohai University), where all the information entropy is close to 8, implying that the proposed method performs a pixel diffusion with an optimized performance.

4.6. Sensitivity Analysis

By modifying values of a few pixels and comparing results, differential attacks could help attackers to easily obtain the rules of encryption. Therefore, the cryptosystem is required to have a certain sensitivity to resist pixel modifications. We adopt a unified average changing intensity (UACI) and number of pixels change rate (NPCR) to measure such sensitivity:

$$\begin{cases} NPCR = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} D_{ij} \times 100\% \\ UACI = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(\frac{I(i,j)-I'(i,j)}{255}\right) \times 100\% \end{cases} \tag{5}$$

where $D_{ij} = \begin{cases} 1, I(i,j) \neq I'(i,j) \\ 0, I(i,j) = I'(i,j) \end{cases}$. I and I' are the original and the modified image, respectively.

As shown in Figure 7, we choose an 8-bit image as the plain image and use the initial key 'GOOD-LUCK' to encrypt the image, obtaining the 'pseudo initialize factor' $(x_1 = 0.401894441844344, p = 0.33533929244635197)$. Afterwards, we change the pseudo-initialized factor to generate a new cipher image E_0. Moreover, we define $x_1 = x_1 + 10^{-14}$ and $p = p + 10^{-14}$ to generate another two cipher images E_2 and E_3.

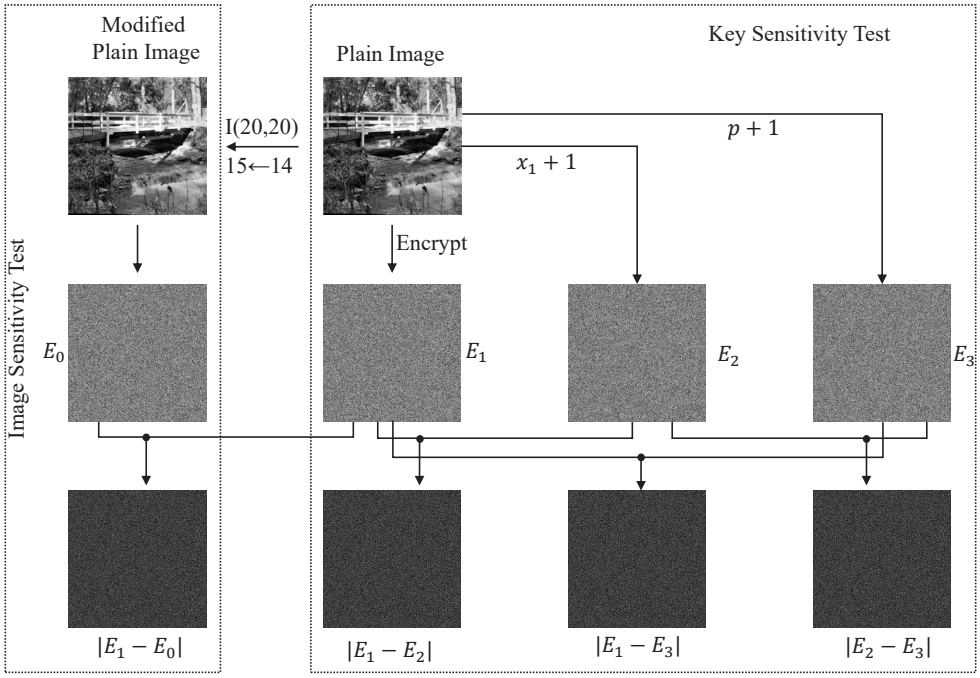

Figure 7. We demonstrate differential cipher images achieved by the proposed method, offering samples on sensitivity and key sensitivity analysis.

The average NPCR and UACI values for different images are shown in Table 3. In fact, Wu et al. [26] state that the expected UACI value should be 33.4635% for an image with ranging values from 0 to 255. It is noted that all results are close to the expected UACI values, which proves that the proposed is capable of resisting differential attacks.

Table 3. Results of NPCR and UACI achieved by the proposed method to evaluate sensitivity against differential attacks.

Images	NPCR	UACI
Poster	99.6732	33.4836
Street	99.5890	33.4821
Subway	99.6016	33.5104
ID Card	99.6232	33.4302
Hohai University	99.6278	33.4886
Key: $E_1 \leftrightarrow E_2$	99.5982	33.3821
Key: $E_1 \leftrightarrow E_3$	99.6107	33.4442
Key: $E_2 \leftrightarrow E_3$	99.5198	33.4250

4.7. Ablation Analysis

To verify the effectiveness of the proposed content-aware encryption module, we make an ablation analysis with "Lena" image as input, which is considered as the most commonly used test image [27]. As shown in Table 4, the performance without the proposed module is relatively poor, since simply adopting DNA encoding without content modeling fails in encoding as muchinformation of the input image. In fact, the proposed content-aware encryption module uses a permutation and a diffusion process to describe informative information remaining in the encoded data, greatly improving the complexity for breaking.

Table 4. Results of ablation analysis on "Lena" image, where CAEM indicates content-aware encryption module.

Ablation Methods	Correlation			Entropy	Sensitivity	
	V	H	D		NPCR	UACI
Ours (without CAEM)	0.1425	0.0986	0.0483	7.0145	93.2143	32.4218
Ours (Full)	0.0032	0.0019	0.0004	7.9982	99.6442	33.4732

4.8. Comparison Experiments

For comparative experiments, we follow the rules in Wu et al. [27], which are currently used by the vast majority of image encryption articles. Table 5 shows the experimental results with the comparing methods. It's noted that results achieved by other methods are directly obtained from their papers.

Table 5. Results achieved by the proposed method and several comparison methods.

Cryptosystems	Correlation			Entropy	Sensitivity	
	V	H	D		NPCR	UACI
Ours	0.0032	0.0019	0.0004	7.9982	99.6442	33.4732
Chen et al. [25]	−0.0064	0.0003	0.0110	7.9993	99.6218	33.5084
Zhang et al. [23]	0.0000016	−0.000003	−0.0000001	-	99.5009	33.4408
Aouissaoui et al. [28]	0.0240	0.0014	−0.0014	7.9978	99.6552	33.5871
Yan et al. [29]	−0.0056	−0.0012	−0.0020	7.9994	99.62	33.55

Essentially, correlation refers to the capability of resistance to statistical attack. Information entropy represents the capability of resistance to entropy attack; meanwhile, sensitivity represents the capability of resistance to the differential attack. It can be observed from Table 5 that the proposed method cannot achieve the best performance on all measurements. For example, Zhang et al. [23] has the best performance in correlation, but the worst in NPCR. Such a phenomenon means their method cannot bear a differential attack. Aouissaoui et al. [28] has the best performance in NPCR, but the worst in entropy and correlation. Such an observation means it would fail under an entropy attack and statistical attack. The proposed method has achieved a balanced performance in three measurements, which represents it does not have as much weaknesses as the other two

comparison methods by obtaining enough capability of resistance to any type of attacks. Therefore, it can be concluded that our method is competitive among these methods.

4.9. Computation Cost Analysis

We compare the computing cost of the proposed method and other DNA computing-based methods in Table 6, thus proving its relatively low computing burden. It's noted that we use color image with 1024×1024 as input. All programs are deployed with C language and tested on the same devices (two laptops with i7-9750H CPU and 8GB DDR3 RAM), thus ensuring the fairness of comparisons. It is noted that the encryption and decryption process would cost 0.491 s and 0.507 s with the proposed method, which has certain advantages when comparing to other DNA computing-based methods.

Table 6. Computation cost comparisons among the proposed method and several DNA computing-based methods.

Cryptosystems	Encryption Time	Decryption Time
Ours	0.491 s	0.507 s
Chen et al. [25]	0.525 s	0.534 s
Zhang et al. [23]	0.572 s	0.568 s
Aouissaoui et al. [28]	0.554 s	0.549 s
Yan et al. [29]	0.598 s	0.602 s

5. Conclusions

The proposed method consists of two modules, i.e., DNA hash encoding and content-aware encrypting module. Specifically, the former one builds the quantity of hash mappings from image pixels to DNA computing bases, thus integrating the advantages of the hash function and DNA computing. The latter one considers strong correlations between adjacent pixels as sources of complexity, thus reorganizing the data structure by highly non-linear functions extracted from neighboring pixels and patches. Experimental results demonstrate that the proposed method could achieve a superior performance in various security measurements, compared with several of the latest comparison methods. Our future work includes DNA computing design on medical images a.

Author Contributions: Conceptualization, H.L. and L.Z.; methodology, H.L. and L.Z.; software, H.C.; validation, H.L., L.Z. and L.Z.; formal analysis, Y.W.; investigation, Y.W.; resources, H.L.; data curation, L.Z.; writing—original draft preparation, H.L. and L.Z.; writing—review and editing, H.L. and Y.W.; visualization, H.L. and L.Z.; supervision, Y.W.; project administration, Y.W.; funding acquisition, H.L. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Provincial Ideological and Political Education Research Projects in the Education Department of Zhejiang Province, the Fundamental Research Funds for the Central Universities under Grant B220202074, the Fundamental Research Funds for the Central Universities, JLU.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

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

References

1. Wu, Y.; Cao, H.; Yang, G.; Lu, T.; Wan, S. Digital Twin of Intelligent Small Surface Defect Detection with Cyber-Manufacturing Systems. *ACM Trans. Internet Technol.* **2022**. [CrossRef]
2. Shu, J.J.; Wang, Q.W.; Yong, K.Y. DNA-based computing of strategic assignment problems. *Phys. Rev. Lett.* **2011**, *106*, 188702. [CrossRef]
3. Adleman, L.M. Molecular computation of solutions to combinatorial problems. *Science* **1994**, *266*, 1021–1024. [CrossRef] [PubMed]

4. Zhang, C.; Wang, Z.; Liu, Y.; Yang, J.; Zhang, X.; Li, Y.; Pan, L.; Ke, Y.; Yan, H. Nicking-assisted reactant recycle to implement entropy-driven DNA circuit. *J. Am. Chem. Soc.* **2019**, *141*, 17189–17197. [CrossRef] [PubMed]
5. Yang, J.; Wu, R.; Li, Y.; Wang, Z.; Pan, L.; Zhang, Q.; Lu, Z.; Zhang, C. Entropy-driven DNA logic circuits regulated by DNAzyme. *Nucleic Acids Res.* **2018**, *46*, 8532–8541. [CrossRef] [PubMed]
6. Zhang, C.; Zhao, Y.; Xu, X.; Xu, R.; Li, H.; Teng, X.; Du, Y.; Miao, Y.; Lin, H.C.; Han, D. Cancer diagnosis with DNA molecular computation. *Nat. Nanotechnol.* **2020**, *15*, 709–715. [CrossRef]
7. Ma, Q.; Zhang, C.; Zhang, M.; Han, D.; Tan, W. DNA Computing: Principle, Construction, and Applications in Intelligent Diagnostics. *Small Struct.* **2021**, *2*, 2100051. [CrossRef]
8. Khan, F.; Ncube, C.; Ramasamy, L.K.; Kadry, S.N.; Nam, Y. A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning. *IEEE Access* **2020**, *8*, 119710–119719. [CrossRef]
9. Zou, C.; Wei, X.; Zhang, Q.; Zhou, C.; Zhou, S. Encryption algorithm based on DNA strand displacement and DNA sequence operation. *IEEE Trans. Nanobiosci.* **2021**, *20*, 223–234. [CrossRef]
10. Namasudra, S. Fast and Secure Data Accessing by Using DNA Computing for the Cloud Environment. *IEEE Trans. Serv. Comput.* **2022**, *15*, 2289–2300. [CrossRef]
11. Song, T.; Eshra, A.; Shah, S.; Bui, H.; Fu, D.; Yang, M.; Mokhtar, R.; Reif, J. Fast and compact DNA logic circuits based on single-stranded gates using strand-displacing polymerase. *Nat. Nanotechnol.* **2019**, *14*, 1075–1081. [CrossRef]
12. Wang, F.; Lv, H.; Li, Q.; Li, J.; Zhang, X.; Shi, J.; Wang, L.; Fan, C. Implementing digital computing with DNA-based switching circuits. *Nat. Commun.* **2020**, *11*, 121. [CrossRef]
13. Thubagere, A.J.; Li, W.; Johnson, R.F.; Chen, Z.; Doroudi, S.; Lee, Y.L.; Izatt, G.; Wittman, S.; Srinivas, N.; Woods, D.; et al. A cargo-sorting DNA robot. *Science* **2017**, *357*, eaan6558. [CrossRef] [PubMed]
14. Liu, L.; Jiang, D.; Wang, X.; Zhang, L.; Rong, X. A Dynamic Triple-Image Encryption Scheme Based on Chaos, S-Box and Image Compressing. *IEEE Access* **2020**, *8*, 210382–210399.
15. Xiao, Y.; Chen, Y.; Long, C.; Shi, J.; Ma, J.; He, J. A novel hybrid secure method based on DNA encoding encryption and spiral scrambling in chaotic OFDM-PON. *IEEE Photonics J.* **2020**, *12*, 1–15. [CrossRef]
16. Wu, Y.; Guo, H.; Chakraborty, C.; Khosravi, M.; Berretti, S.; Wan, S. Edge Computing Driven Low-Light Image Dynamic Enhancement for Object Detection. *IEEE Trans. Netw. Sci. Eng.* **2022**, 1. [CrossRef]
17. Babaei, M. A novel text and image encryption method based on chaos theory and DNA computing. *Nat. Comput.* **2013**, *12*, 101–107. [CrossRef]
18. Wang, X.; Liu, C. A novel and effective image encryption algorithm based on chaos and DNA encoding. *Multim. Tools Appl.* **2017**, *76*, 6229–6245. [CrossRef]
19. Samiullah, M.; Aslam, W.; Nazir, H.; Lali, M.I.U.; Shahzad, B.; Mufti, M.R.; Afzal, H. An Image Encryption Scheme Based on DNA Computing and Multiple Chaotic Systems. *IEEE Access* **2020**, *8*, 25650–25663. [CrossRef]
20. Malik, M.G.A.; Bashir, Z.; Iqbal, N.; Imtiaz, M.A. Color Image Encryption Algorithm Based on Hyper-Chaos and DNA Computing. *IEEE Access* **2020**, *8*, 88093–88107. [CrossRef]
21. Khan, J.S.; Boulila, W.; Ahmad, J.; Rubaiee, S.; Rehman, A.U.; Alroobaea, R.; Buchanan, W.J. DNA and Plaintext Dependent Chaotic Visual Selective Image Encryption. *IEEE Access* **2020**, *8*, 159732–159744. [CrossRef]
22. Ravichandran, D.; Banu, S.A.; Murthy, B.K.; Balasubramanian, V.; Fathima, S.; Amirtharajan, R. An efficient medical image encryption using hybrid DNA computing and chaos in transform domain. *Med. Biol. Eng. Comput.* **2021**, *59*, 589–605. [CrossRef] [PubMed]
23. Zhang, Q.; Han, J.; Ye, Y. Multi-image encryption algorithm based on image hash, bit-plane decomposition and dynamic DNA coding. *IET Image Process.* **2021**, *15*, 885–896. [CrossRef]
24. Wu, Y.; Zhang, L.; Berretti, S.; Wan, S. Medical Image Encryption by Content-Aware DNA Computing for Secure Healthcare. *IEEE Trans. Ind. Inform.* **2023**, *19*, 2089–2098. [CrossRef]
25. Chen, J.; Zhu, Z.; Zhang, L.; Zhang, Y.; Yang, B. Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption. *Signal Process.* **2018**, *142*, 340–353. [CrossRef]
26. Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI randomness tests for image encryption. *Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun. (JSAT)* **2011**, *1*, 31–38.
27. Wu, Y.; Zhou, Y.; Noonan, J.P.; Agaian, S. Design of image cipher using latin squares. *Inf. Sci.* **2014**, *264*, 317–339. [CrossRef]
28. Aouissaoui, I.; Bakir, T.; Sakly, A. Robustly correlated key-medical image for DNA-chaos based encryption. *IET Image Process.* **2021**, *15*, 2770–2786. [CrossRef]
29. Yan, X.; Wang, X.; Xian, Y. Chaotic image encryption algorithm based on arithmetic sequence scrambling model and DNA encoding operation. *Multim. Tools Appl.* **2021**, *80*, 10949–10983. [CrossRef]