REVERSIBLE HIDING METHOD USING SECRETS SHARING OF DNA-XNOR

Huy Cuong Do[1], *Thai Hung Pham*[2], *Minh Thanh Ta*[1]

https://doi.org/10.56651/lqdtu.jst.v11.n01.363.ict

Abstract

In this paper, we propose a reversible hiding solution based on secret sharing method using DNA exclusive-NOR (DNA-XNOR) for color images. The DNA-XNOR secret sharing method is used as a secret key matrix to hide and to extract information. We use the mean square error (MSE) to select the optimal embedding value for each pixel. These values will be used to embed the secret information into the image from which we can obtain the highest PSNR value of the embedded images. In our solution, the secret information will be embedded on all three color channels Red (R), Green (G), and Blue (B) of the original image. The proposed algorithm DNA-XNOR is a new method for sharing confidential information in the field of information hiding. Our experimental results show the efficiency of the algorithm and demonstrate its applicability in practice.

Index terms

Image watermarking, Copyright protection, Digital content, Secret sharing, DNA-XNOR.

1. Introduction

Nowadays, along with the development of technology, secret information hiding techniques have received attention. Such researches are widely applied in military, diplomatic, and security agencies, education and businesses when it comes to the exchange of important information. Confidential information hiding is a technique to hide information in multimedia files (images, audio, video, ...) to transmit to the recipient without the third party knowing the existence of information during transmission. This technique also changes the thinking in the field of information security because of the feasibility of hiding a large amount of secret information in an ordinary data that is difficult to detect by human perception.

Multimedia files (photos, audio, video, ...) are larger in size than normal text files. Therefore, images, audio, and videos are convenient to use as containers to hide information. To prevent hidden data from being attacked by third parties, hiding techniques need to satisfy a number of criteria such as: Hidden information is unlikely to be detected by normal perceptual systems; does not "change" the media containing the

[1]Faculty of Information Technology, Le Quy Don Technical University
[2]IT Center, Le Quy Don Technical University

information; cannot be checked without the appropriate key [1]. At the same time, consider the capacity of hiding information with other criteria such as image quality or sustainability, secrecy and resistance to forgery. Evaluating the hiding algorithm, people often consider some criteria as follows: (i) The optimization of the information embedding algorithm; (ii) The optimization of the information extraction algorithm; (iii) Information hiding capacity; (iv) Sustainability; (v) Transparency (invisibility of hidden information); and (vi) Reliability.

To ensure these criteria, many different hiding algorithms have been developed and published. Algorithms are generally classified into three groups:

- *Spatial domain*: The feature of the techniques for hiding information into the spatial domain is that the container will not or rarely be processed before hiding the information. Some algorithms and techniques are commonly used to hide information in the spatial domain such as: LSB (Least Significant Bit) [1], [2]; Pseudo-random Permutation [3]; Block hiding method; Brundox method; The Darmstadter-Dellegle-Quisquotter-McCa method.

- *Frequency domain*: Techniques to hide information into the frequency domain are usually process the image containing and then proceed to hide the information. Some commonly used algorithms are: Discrete Cosine Transforms (DCT) [1], [2], [3]; Discrete Wavelet Transform (DWT) [4], [5], [6]; Discrete Fourier Transform (DFT) [7], [8], [9]; and the associated frequency domain transform methods [10].

The approaches to hide the information in the space domain and the frequency domain have been mainly proposed recently. In addition, there are also a number of scientific articles and research works that have proposed a number of other approaches such as: Cox technique or direct sequence spreading technique [11], combination of spatial domain and frequency domain, and so on.

The sequence of the DNA sequence has complementary properties such as “0” and “1” in binary and has applications in information security and hiding. Therefore, some researchers have begun to apply the complementary properties of DNA sequences in security and information hiding. Some researchers have proposed some solutions presented in [6-12]. Although, the published solutions have solved many problems previously posed by the hiding algorithm based on DNA sequence encoding. However, these algorithms also have some limitations such as error conflicts or difficulty in implementation. Specifically, in the [12] solution, the Tuncer team has exploited the similarity of the XOR operator on DNA sequence encoding systems. The XOR mathematical combinations of nucleotide pairs in DNA are also thoroughly exploited by the authors to represent information embedding states of secret information. However, the assessment of pixel influence after the information embedded has not been completely resolved. Lee $et\ al.$ [13] uses DNA sequences to represent the histogram of pixels and uses frequency shifting of the histogram to hide information reversibly. However, because Lee uses the histogram of the pixels, the influence of image quality after embedding information will be a weakness if the amount of hidden information is large. The idea

of using DNA encoding to hide classified information and containers is also proved in Babatunde's paper [14]. The cipher data is converted to a binary 8-bit string and replaced by probability-based nucleotide states. However, the study by Babatunde [14] has not yet evaluated the effect of embedding information in the container. Therefore, a possible solution to solve the problem of increasing the hiding capacity and limiting the impact on the quality of the container is to use the theory of DNA modeling by exploiting the natural properties of the DNA sequence.

In this paper, we propose to use the color image hiding method using the DNA-XNOR secret key sharing scheme, which is first applied in reversible hiding. Each 2 bits of information are calculated, compared and selected in a three-component truth table based on the average mean square error (MSE) value for each case to embed into the corresponding R, G, B color channels of the original image. Our paper using the evaluation methods are MSEs, peak signal to noise ratio (PSNR), bit error ratio (BER), pixel distortion ratio (PD), structural similarity index (SSIM) to evaluate the proposed algorithm.

The structure of the paper is described as follows: In Section 2, we briefly describe the basic concept of DNA encoding and decoding methods for color images. Section 3 presents the details of the reversible hiding algorithm using the DNA-XNOR secret sharing method. For the results of the proposed method, we use PSNR, BER, PD and SSIM to evaluate and show the experimental results in Section 4. In Section 5, we present the conclusions of the paper.

2. DNA sequences

2.1. DNA sequence coding rules

A DNA sequence is made up of 4 types of nucleotides: A (adenine), C (cytosine), G (guanine), T (thymine). A and T, G and C have complementary properties like "0" and "1" in binary. The hiding algorithm uses this addition for 2 binary bits to enhance the security of the information and not change much of the image structure. The set of encoding and decoding rules using the DNA sequence uses the following table of 24 coding rules [6]:

2.2. XNOR operator and DNA-XNOR operator

The bit operation XNOR takes two bit sequences of the same length and performs a logical operation on each corresponding bit pair, which is the inverse of the XOR operation. That means the resulting in "1" if the two bits are the same and the result is "0" if the two bits are different. Applying the XNOR operator to the DNA sequence encoded by rule 1, Table 1, we get a DNA - XNOR calculation table like Table 2.

2.3. DNA encoding and decoding for color images

A normal color image is split into separate R, G, and B color channels, and the corresponding R, G, and B channels are converted to binary code. Finally, each pixel

Table 1. The 24 coding rules

	A	T	G	C		A	T	G	C
1	00	11	01	10	**13**	10	11	00	11
2	00	11	10	01	**14**	10	11	01	00
3	00	01	11	10	**15**	10	00	11	01
4	00	01	10	11	**16**	10	00	01	11
5	00	10	01	11	**17**	10	01	00	11
6	00	10	11	01	**18**	10	01	11	00
7	01	11	00	10	**19**	11	10	00	01
8	01	11	10	00	**20**	11	10	01	00
9	01	00	11	10	**21**	11	00	10	01
10	01	00	10	11	**22**	11	00	01	10
11	01	10	00	11	**23**	11	01	00	10
12	01	10	11	00	**24**	11	01	10	00

Table 2. XNOR operator for DNA strings

XNOR	A	T	G	C
A	T	A	C	G
T	A	T	G	C
G	C	G	T	A
C	G	C	A	T

of the R, G, and B channels is represented as a DNA sequence. For example, if a pixel on the Blue channel of an image has the binary code [11001001], then the DNA sequence is [TACG] according to rule 1 of Table 1. Assuming a 24-bit color image, size $m \times n$, convert to a 3-dimensional binary matrix R, G, B according to the formula: $A = [S_{i,j,k}]_{m \times n \times k}$ with $S_{i,j,k} \in \{0.1\}$ and $(i, j, k) \in \{0, 1, ..., m-1\} \times \{0.1, ..., n-1\} \times \{0.1, ..., k-1\}$. XNOR function for 3 channels R, G, B is determined according to the following formula (1):

$$\overline{R \oplus G \oplus B} = \overline{[S_{i,j,1} \oplus S_{i,j,2} \oplus S_{i,j,3}]_{mxnxk}} \quad (1)$$

The binary code of the pixel $A_{i,j}$ at (i,j) can be converted to decimal by the formula (2) [7]:

$$A_{i,j} = a_{i,j,k-1}2^{k-1} + ... + a_{i,j,1}2^1 + a_{i,j,0}2^0 \quad (2)$$

3. Proposal of reversible data hiding

3.1. Secret key sharing scheme DNA-XNOR

The secret key sharing scheme was introduced to protect the reliability of encryption keys or data with the aim to provide a trusted and secure key. Many algorithms and schemes for sharing secret keys are proposed. To achieve an optimal steganography, a secret key sharing scheme is proposed in our paper. This secret key sharing scheme was proposed by Wang [8], the idea of the scheme is that from initial one component split into n components. Afterwards, when such n components are XNOR-ed together, the initial part is correctly generated.

Let A_0 be the binary secret:

Step 1: Generate random $n-1$ components as $B_1...B_{n-1}$;

Step 2: Make n into $A_1...A_n$ by the formula:

$A_1 = B_1$

$A_2 = \overline{B_1 \oplus B_2}$

...

$A_i = \overline{B_{i-1} \oplus B_i}$

$A_n = \overline{B_{n-1} \oplus} \mathrm{A}_0$

Example: We have secret information A_0 = 00, split A_0 into 3 components A_1, A_2, A_3; we have $n = 3$.

Step 1: Select random $n - 1 = 2$ any component $B_1 = 01, B_2 = 11$;

Step 2: Create 3 components A_1, A_2, A_3 according to the formula:

$A_1 = B_1 = 01$

$A_2 = \overline{B_1 \oplus B_2} = \overline{01 \oplus 11} = 01$

$A_3 = \overline{B_2 \oplus A} = \overline{01 \oplus 00} = 10$

$\overline{A_1 \oplus A_2 \oplus A_3} = \overline{01 \oplus 01 \oplus 10} = 00 = A_0$

Thus, given the secret key information B_i, where $i = 1, ..., n$ is randomly generated, then the secret information is distributed according to a rule preselected in Table 1.

3.2. Algorithm idea

Confidential information will be converted to an acid nucleotide sequence. Each nucleotide is divided into 3 parts using the DNA-XNOR operator, so that, these 3 components when XNOR together will generate the nucleotides initial. Each of those components, in turn, is 3 components R, G, B, which will be hidden in each R, G, and B channel of the color image. For the application to 3-channel color images, the value n is set to 3. The secret shared key will be chosen at random from the 24 encryption rules defined in table 1.

Starting from the secret key sharing scheme for 3 components and combined with the DNA-XNOR operator, we need to build a 3-component truth table for nucleotides A, T, G, C according to the principle of DNA sequences. Each nucleotide is initially separated into 3 components so that when XNOR 03 components, we get nucleotides initial. Table 3 presents a 3-way truth table XNOR part for nucleotide A. Similarly we also have 3-component truth tables XNOR for T, G and C.

Table 3. Truth Table 2 XNOR components for nucleotide A

1	A	A	A	**9**	G	A	G
2	A	T	T	**10**	G	G	A
3	A	G	G	**11**	G	C	T
4	A	C	C	**12**	G	T	C
5	C	A	C	**13**	T	A	T
6	C	C	A	**14**	T	T	A
7	C	G	T	**15**	T	C	G
8	C	T	G	**16**	T	G	C

3.3. Information hiding algorithm

The proposed hiding algorithm in the method is shown in Fig. 1. The hiding algorithm consists of the following steps:

Input: (1) CI color image size $m \times n$; (2) Secret message M.

Output: SI color image of size $m \times n$, carrying message.

Step 1: Take the 3 color channels R, G, B of the original CI color image ($m \times n$), calculate the maximum number of bits that can be hidden as $l_{CI} = m * n * 2 - 1$;

Step 2: Convert M to binary M_1, check the message string length; if $l_{M_1} > l_{CI}$, end the program.

Step 3: Convert cryptographic data to binary M_1

According to Table 1, get the string M_2 in the form of DNA.

Step 4: Apply the DNA-XNOR operator to each element of the secret data, at the i-th nucleotide of M_2, obtain data table D whose values are derived from the the XNOR operation for A, T, G, C nucleotits.

Step 5: Get the value of the last 2 bits of each pixel in all 3 channels R, G, B by dividing the residual value of each color channel by 4.

Step 6: Calculate the local MSE of each case according to the formula:

$$MSE_{D_j} = \frac{1}{3}\left[(D_{R,j} - R)^2 + (D_{G,j} - G)^2 + (D_{B,j} - B)^2\right], \quad (3)$$

where MSE_{D_j} is the local MSE of pixel i for each case j; $D_{R,j}$, $D_{G,j}$, $D_{B,j}$ are the R, G, B values of truth table; R, G, B are the values calculated in step 5.

Step 7: Select the set of 3 numbers for the smallest MSE value.

Step 8: Change the last 2 bits of the old 3 pixel elements from R, G, B component with the values of the new triplet obtained from step 7.

Step 9: Check the end of the string information; if not finished, go back to step 4; if finished, mark the next pixel as the end point, outputting the hidden image.

Fig. 2 shows embedding 02 bits 00 (A - according to rule 1) into a pixel whose color channel value R, G, B is 132, 180, 229, respectively. Our method calculates values of

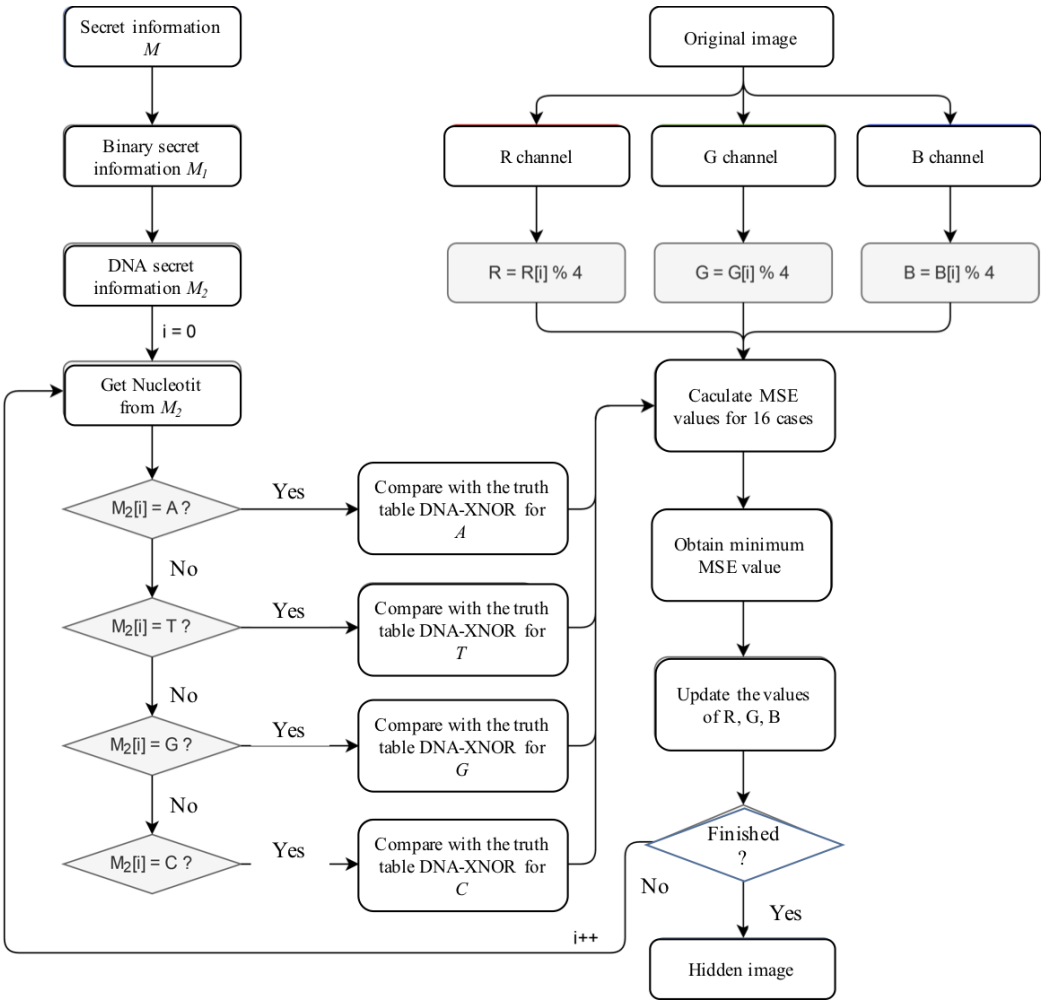
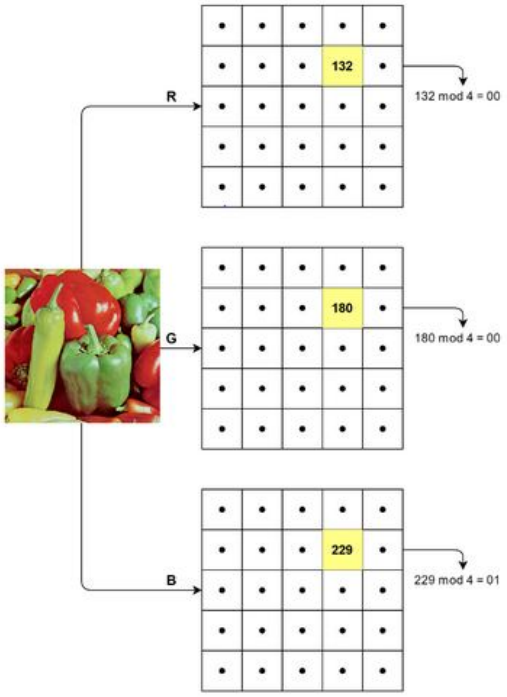


Fig. 1. A scheme of hiding information with the DNA-XNOR operator.

local MSE of each case for A with value 00 00 01 according to Table 3, the choose the smallest MSE value. In case of multiple equal values, randomly select a set of numbers. In the example of Fig. 2, we choose k = 1 (A A A) transformed pixels have values of 132, 180, and 228, respectively. The information embedding algorithm selects the position with the lowest pixel influence to optimize the embedding position and replace 2 bits of secret information equals 3 states of 2 bits according to the nucleotide property demonstrated in section 3.1. Computational complexity is the change of pixel values of each pixel on 3 image planes. Therefore, the computational complexity is simply the total number of substitutions $O(n)$.

3.4. Information extraction algorithm

The extraction algorithm is explained in Fig. 3. The information extraction algorithm includes the following steps:



XNOR truth table for A and calculate MSE with the value of 00 00 01

k	R	G	B	MSE_k	k	R	G	B	MSE_k
1	0 0 / A	0 0 / A	0 0 / A	**0.33**	9	0 1 / G	0 0 / A	0 1 / G	**0.33**
2	0 0 / A	1 1 / T	1 1 / T	**4.33**	10	0 1 / G	0 1 / G	0 0 / A	**1**
3	0 0 / A	0 1 / G	0 1 / G	**0.33**	11	0 1 / G	1 0 / C	1 1 / T	**3**
4	0 0 / A	1 0 / C	1 0 / C	**1.67**	12	0 1 / G	1 1 / T	1 0 / C	**3.67**
5	1 0 / C	0 0 / A	1 0 / C	**1.67**	13	1 1 / T	0 0 / A	1 1 / T	**4.33**
6	1 0 / C	1 0 / C	0 0 / A	**3**	14	1 1 / T	1 1 / T	0 0 / A	**6.33**
7	1 0 / C	0 1 / G	1 1 / T	**3**	15	1 1 / T	1 0 / C	0 1 / G	**4.33**
8	1 0 / C	1 1 / T	0 1 / G	**4.33**	16	1 1 / T	0 1 / G	1 0 / C	**3.67**

Fig. 2. An example of embedding A (00) according to our proposed method.

Input: Color image SI with size of $m \times n$.

Output: Secret message M.

Step 1: Get the R, G, B channels of the SI color image.

Step 2: Check the end of the message string marker.

Step 3: Take each color channel of SI, divide it by 4, get the last 2 bits of the 3 channels R, G, and B.

Step 4: Convert the 3 values of step 3 to the form of the DNA code according to rule 1.

Step 5: XNOR of the 3 components obtained from step 4, obtains a single component M_2. Convert the element to the form of binary.

Step 6: Go back to step 2, apply to next pixels until fit the string marker.

Step 7: Converting the binary message string to characters, we obtain the secret message string M.

An example of extraction can be shown in Fig. 3. We extract information at pixels with values 132, 180, 228; first check if it's the end of the message chain, then mod each value for 4 to get 00, 00, 00 (converting to DNA form according to rule 1, we get a set of 03 values A A A); perform XNOR with 3 values A A A obtained A corresponding

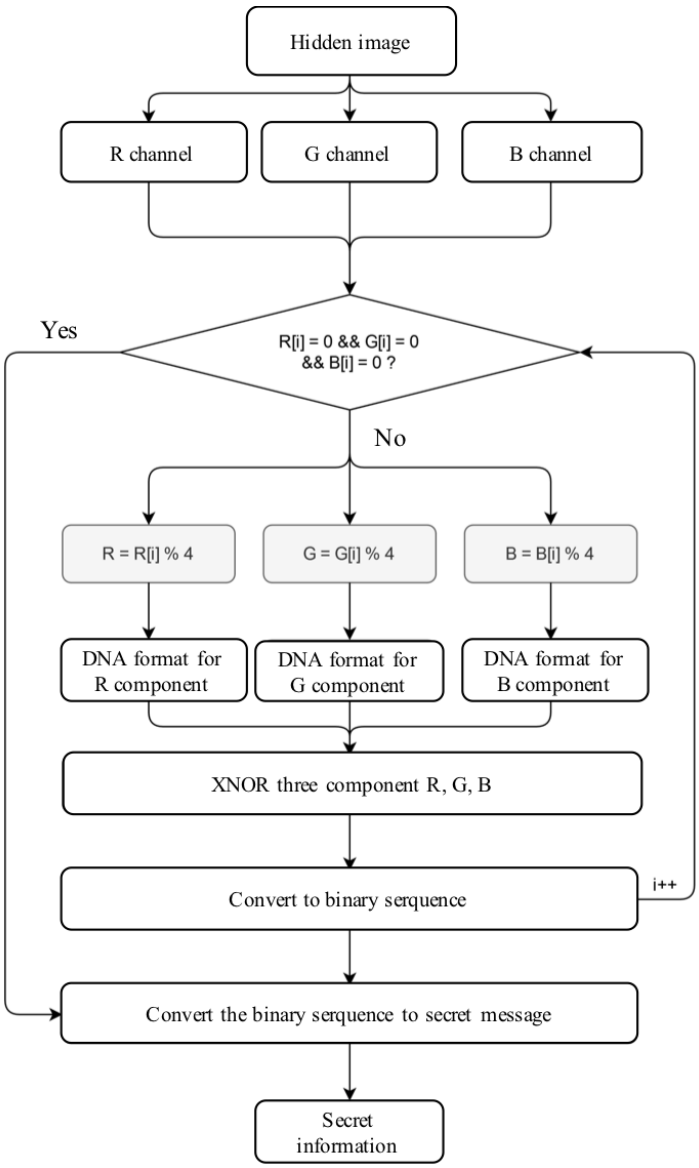


Fig. 3. Diagram of information extraction with DNA-XNOR operator.

to the binary code 00. The operation to recover the hidden secret information can be calculated based on the operation proved in section 3.1.

4. Experiment and evaluation

Experiments are conducted on Windows 10 Pro operating system and the develop language used is C#. To evaluate the experimental results, we use the MSE, PSNR,

Table 4. The PSNR results of our experimental

Image name	**16K**	**80K**	**160K**	**480K**
SanDiego	56.14	54.81	53.85	51.34
San Francisco	57.04	55.67	54.28	51.01
Oakland	56.26	54.77	54.01	51.09
North Island	57.15	56.23	54.24	51.23
Point Loma	56.53	55.02	53.51	50.84
Splash	58.75	56.38	54.46	50.98
Mandrill	57.32	56.58	54.45	50.70
Peppers	57.65	54.96	54.15	51.19
Average	**57.11**	**55.55**	**54.12**	**51.05**

BER, PD and SSIM parameters defined in the formulas (4), (5), (6), (7), (8) [9-13].

$$MSE = \frac{1}{m*n}\sum_{1}^{m}\sum_{1}^{n}\left(CI_{i,j} - SI_{i,j}\right)^2 \tag{4}$$

$$PSNR = 10*log\frac{Max\left(CI_{i,j}^2\right)}{MSE} \tag{5}$$

$$PD = \frac{Sum_{bitthaydoi}}{Sum_{bitgiautin}} \tag{6}$$

$$BER = \frac{Sum_{bitthaydoi}}{Sum_{bit}} \tag{7}$$

$$SSIM_{(CI,SI)} = \frac{\left(2\mu_1\mu_2 + C_1\right)\left(2\delta_{CISI} + C_2\right)}{\left(\mu_1^2 + \mu_2^2 + C_1\right)\left(\delta_{CI}^2 + \delta_{SI}^2 + C_2\right)}, \tag{8}$$

where CI is the original image; SI is the image containing the information; μ_1 is the average of the CI image; μ_2 is the average of the SI images; δ_{CI}^2 is the variance of the original image; δ_{SI}^2 is the variance of the news image; L is range of values of pixels (one pixel encoded in 8 bits $L = 2^8 - 1$); k_1 = 0.01; k_2 = 0.03.

- *Experimental scenario*: Experimental data is taken from the website of USC-SIPI [1] in .tiff color image format (24 bits/pixel) with size of 512×512 pixels, 768 Kb. We test on 08 with size of 512×512 images with named "SanDiego", "San Francisco", “Oakland”, “North Island”, “Point Loma”, “Splash”, “Mandrill”, “Peppers”, respectively. Number of bits hidden in the image is 16K (3.05%), 80K (15.25%), 160K (30.5%), 480K (91.5%), respectively. Specific experimental results in Table 4, Table 5 and Fig. 8 compare the original image and the hidden image according to the algorithm's experimental scenario.

DNA-XNOR algorithm recommends using sequential embedding and using 2 bpp. Therefore, to compare the efficiency of the DNA-XNOR algorithm, we compare the efficiency of the DNA-XNOR algorithm, in this experimental result compare it with

[1]http://sipi.usc.edu/database/

Table 5. The SSIM results of our experimental

Image name	**16K**	**80K**	**160K**	**480K**
SanDiego	0.99997	0.99997	0.99996	0.99993
San Francisco	0.99999	0.99999	0.99998	0.99996
Oakland	0.99998	0.99997	0.99997	0.99994
North Island	0.99999	0.99998	0.99997	0.99995
Point Loma	0.99997	0.99996	0.99995	0.99991
Splash	0.99999	0.99999	0.99999	0.99997
Mandrill	0.99999	0.99999	0.99998	0.99995
Peppers	0.99999	0.99999	0.99998	0.99997
Average	**0.99999**	**0.99998**	**0.99997**	**0.99995**

Table 6. Comparison of DNA–XNOR and LSB method

	PSNR		**SSIM**	
Image name	**DNA-XNOR**	**LSB**	**DNA-XNOR**	**LSB**
SanDiego	52.99	49.03	0.99995	0.99987
San Francisco	53.25	48.96	0.99998	0.99993
Oakland	52.90	49.08	0.99996	0.99990
North Island	53.11	49.11	0.99997	0.99991
Point Loma	52.39	49.04	0.9999	0.99986
Splash	53.33	48.95	0.99998	0.99995
Mandrill	52.47	49.02	0.99997	0.99993
Peppers	52.87	48.87	0.99998	0.99994
Average	**52.91**	**49.01**	**0.99996**	**0.99991**

the classical LSB hiding method with 2 bpp. The scenario is briefly described as follows: Each image pixel is represented by a set of 3 integers (R, G, B), each value pixel is from 0 - 255 (8 bits); changing 2 bits if the last bit of each channel is equal to 2 bits of the information to be hidden, the pixel value on each channel will change by an amount from -3 to 3.

- Table 6 effectively compares two algorithms with embedding rate of 45.8% on 512×512 size images and compares PSNR and SSIM values of 8 experimental images applied in the experiment for the proposed method and the classical LSB method. The results show that the proposed method is superior in terms of quality of embedded images by considering the influence on the difference of each pixel before and after embedding information.

To compare the affected image quality with the increase of the hidden message capacity, we gradually increase the number of hidden bit tests to evaluate. The higher the number of hidden bits, the better without affecting the quality of the hidden image. Fig. 4 is a comparison of the value of the proposed method with the classical LSB method when increasing the number of embedded bits on the same container image. The results show that, when the number of hidden bits in the image increases, the proposed method has the image quality (PSNR value) not affected much, so the algorithm can be applied in practice.

- *Algorithm efficiency analysis*: Conduct experiments on the proposed algorithm and

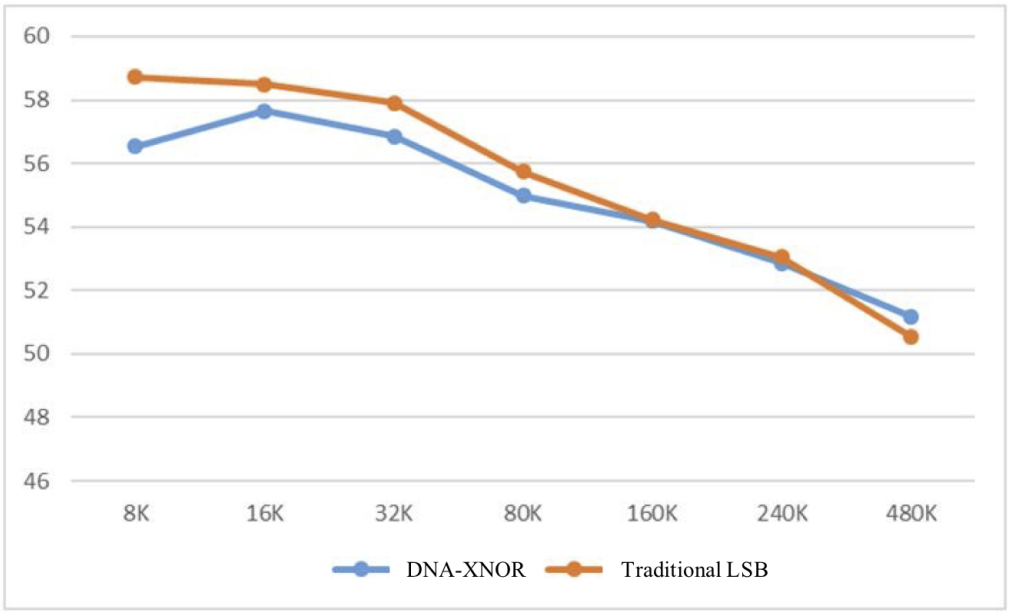


Fig. 4. Comparison of PSNR values of two DNA-XNOR and LSB methods by number of embedded bits.

the classical LSB algorithm to test and compare the effectiveness of information hiding and against image detection methods. Currently popular hiding information, the results are as follows:

- Apply advanced LSB method [15] to check for any changes between the original image and the hidden image. Fig. 5 shows that when the number of hidden bits is 45.8% (240K bits), it is difficult to distinguish the hidden image of the DNA-XNOR algorithm with the human eyes. Fig. 5 (d), (e), (f) give a different result when representing the LSB plane by applying pixel - by - pixel comparison, we can say that, at embedding ratio less than 50% there is little variation in the LSB planes of the original and hidden images.

- Applying hiding information detection technique based on Histogram shift [15], gives us results when comparing the difference histogram correlation of 03 images (original image, hidden image by DNA-XNOR and hidden image by LSB dictionary) with an embedding rate of 60%, it is difficult to detect hidden images by DNA-XNOR method. In addition, Fig. 6 gives the comparison results when analyzing the Pixel Difference Histogram [16] on image 8 between the original image and the 320000 bit hidden image (embedding rate 61%).

- With statistical analysis technique, applying [15] shows the effect of LSB embedding for each neighboring pixel. An inefficient hiding algorithm increases or decreases the LSB by embedding up to 26 neighboring pixels. From Fig. 7, it can be seen that the neighboring pixels do not change too much in the case of embedding rate below 50%.

The new feature of the DNA-XNOR method is the security of the information hidden

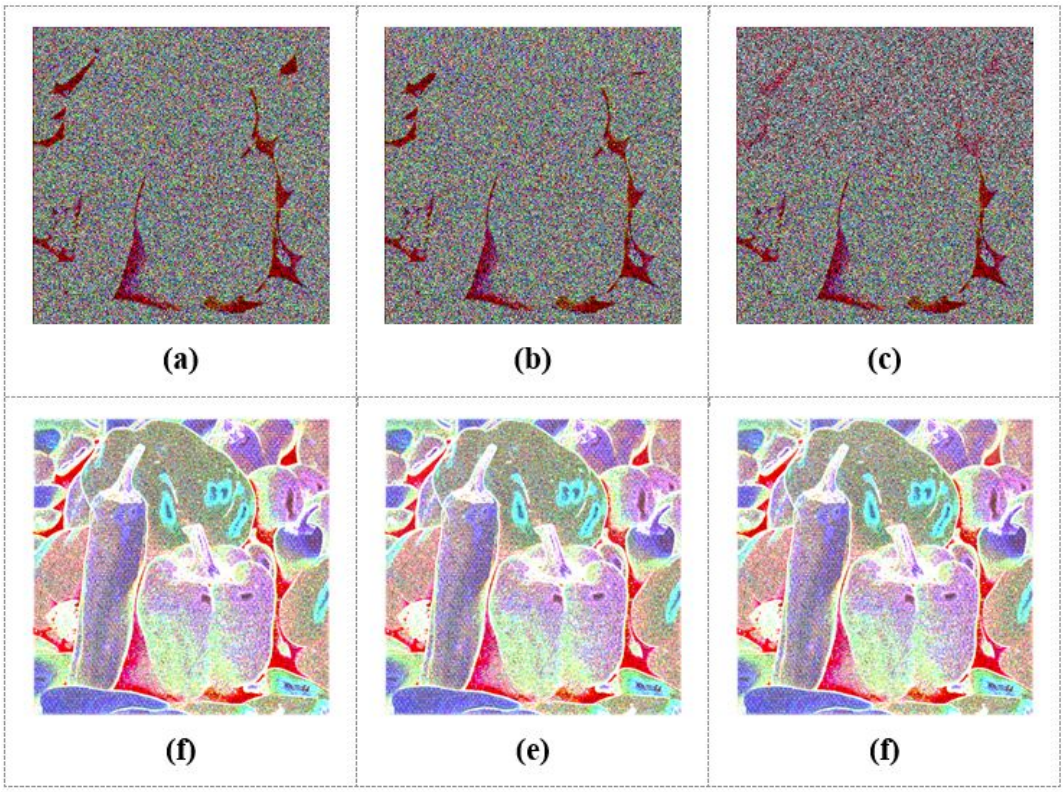


Fig. 5. Compare LSB and DNA-XNOR (b), (e) and classical LSB (c), (f) with original image (a), (f).

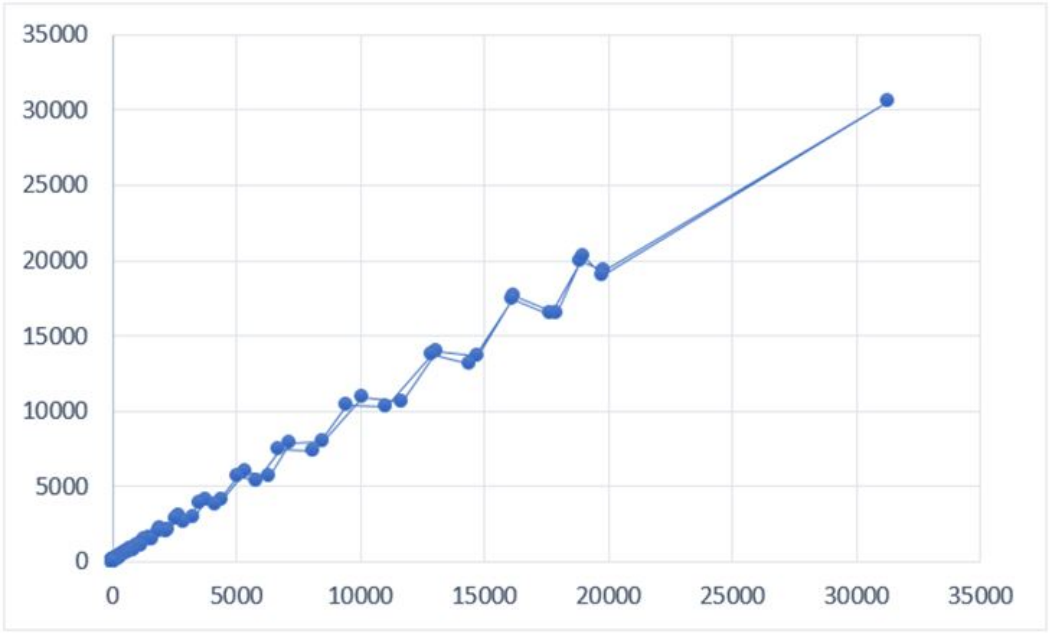


Fig. 6. Compare results when analyzing Pixel Difference Histogram.

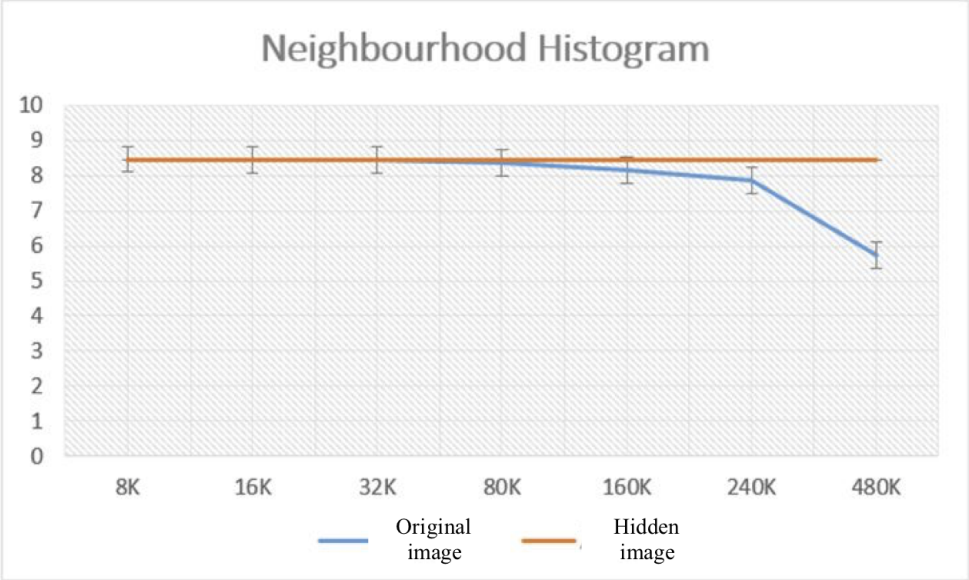


Fig. 7. Compare neighboring pixels of the original image with the hidden image according to the embedding ratio.

in the image. Embedded information will be encoded according to DNA coding rules, the embedding mechanism will choose the best value to embed in the image. There are 24 encryption rules applied to encrypt the message. There are 24 embedding and extraction rules respectively. For example, the message is encrypted according to rule 1, when embedding the message in the image, we use any of the 24 rules, so the security and transparency of the hidden information is enhanced. Thus, when experimentally comparing the two methods, we have some conclusions as follows for the hiding algorithm DNA-XNOR, the article has achieved the following criteria:

(i) The optimization of the information embedding algorithm: In the proposed algorithm, the author used the hidden way data into R, G, B color channels and a shared scheme DNA-XNOR secret key together.

(ii) Optimization of information extraction mechanism: The image is split into R, G, B channels; afterward use DNA-XNOR operator to extract information. This algorithm is the optimal way to obtain information extraction.

(iii) Information hiding capacity: Algorithm allows embedding information with high capacity. In this article, we used the embedding 2 bpp (bits above per pixel) for color images.

(iv) Persistence: Information cannot be extracted when the image is hacked.

(v) Transparency (invisibility of hidden information): The proposed method uses the technique of sharing secret key DNA-XNOR and multiple values are used to share. This method proposes a way to choose the optimal value for embedding (select the value with the smallest mean square error), so the proposed method has very high transparency.

Fig. 8. Compare original image (a) and hidden image (b) with embedding rate 30.5%.

(vi) Reliability: Secret key sharing technique DNA-XNOR uses an image as a secret key. The operator DNA-XNOR uses 24 coding rules different ways for encoding and decoding. Reliability of the information provided using the key sharing technique mentioned output in the algorithm.

5. Conclusions and future works

The DNA-XNOR algorithm approach is proposed from the point of view of efficiency of hiding information and ensuring image quality after hiding information. The algorithm uses 2bpp, but the result of the algorithm's average PSNR is at 54.45 dB (greater than 48.13 dB) and the SSIM is at 0.99997. Algorithm DNA algorithm - XNOR is

a simple, easy to install, safe algorithm; provides an optimal information extraction and embedding mechanism, large hiding capacity, high image quality and ensures the reliability of information. In addition, the algorithm also proposes to use the shared key DNA-XNOR operator secret on the basis of 24 DNA coding rules.

References

[1] V. Barannik, B. Gorodetsky, and N. Barannik, "Steganography – theory and practice," *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, vol. 9, no. 1, pp. 45–48, Mar. 2019. [Online]. Available: https://ph.pollub.pl/index.php/iapgos/article/view/781

[2] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[3] S. Katzenbeisser and F. A. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, 1st ed. USA: Artech House, Inc., 2000.

[4] C.-S. Lu, *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*. USA: IGI Global, 2004.

[5] O. D. King and P. Gaborit, "Binary templates for comma-free dna codes," *Discrete Applied Mathematics*, vol. 155, no. 6, pp. 831–839, 2007, computational Molecular Biology Series, Issue V. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166218X06003714

[6] T. Xie, Y. Liu, and J. Tang, "Breaking a novel image fusion encryption algorithm based on dna sequence operation and hyper-chaotic system," *Optik*, vol. 125, no. 24, pp. 7166–7169, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030402614011358

[7] D. Wang, X. Li, and F. Yi, "Probabilistic (n, n) visual secret sharing scheme for grayscale images," in *Information Security and Cryptology*, D. Pei, M. Yung, D. Lin, and C. Wu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 192–200.

[8] A. Tanchenko, "Visual-psnr measure of image quality," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 874–878, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320314000091

[9] C.-C. Wang, Y.-F. Chang, C.-C. Chang, J.-K. Jan, and C.-C. Lin, "A high capacity data hiding scheme for binary images based on block patterns," *Journal of Systems and Software*, vol. 93, pp. 152–162, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121214000545

[10] H.-K. Pan, Y.-Y. Chen, and Y.-C. Tseng, "A secure data hiding scheme for two-color images," in *Proceedings of the Fifth IEEE Symposium on Computers and Communications (ISCC 2000)*, ser. ISCC '00. USA: IEEE Computer Society, 2000, p. 750.

[11] M. M. Abd-Eldayem, "A proposed security technique based on watermarking and encryption for digital imaging and communications in medicine," *Egyptian Informatics Journal*, vol. 14, no. 1, pp. 1–13, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110866512000515

[12] T. Tuncer and E. Avci, "A reversible data hiding algorithm based on probabilistic dna-xor secret sharing scheme for color images," *Displays*, vol. 41, pp. 1–8, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141938215300238

[13] S.-H. Lee, "Reversible data hiding for dna sequence using multilevel histogram shifting," *Security and Communication Networks*, 2018.

[14] O. O. Babatunde, "Deoxyribonucleic acid (dna) as a hypothetical information hiding medium: Dna mimics basic information security protocol," 2011.

[15] T. Zhang and X. Ping, "Reliable detection of lsb steganography based on the difference image histogram," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 3, 2003, pp. III–545.

Manuscript received 12-2-2022; Accepted 25-5-2022.

■

Huy Cuong Do is currently studying Network Technology in Le Quy Don University, Vietnam. His field of research is digital watermarking for digital multimedia.
Email: hoangdung240994@gmail.com.

Thai Hung Pham received the degree of IT engineer and Master of Information System from Le Quy Don University of Technology in 2007 and 2013. (From September 2007 to January 2019: he was an assistant, February 2019 - now: Head of IT Center/LQD University of Technology). His research interests are in the fields of digital watermarking, network technology, information security and machine vision. E-mail: hungpt@lqdtu.edu.vn

Minh Thanh Ta is currently an associate professor and vice dean of Faculty of Information Technology in Le Quy Don Technical University, Vietnam. He is also a Postdoctoral Fellow of the Department of Mathematical and Computing Sciences at Tokyo Institute of Technology. He received his B.S. and M.S in Computer Science from National Defense Academy, Japan, in 2005 and 2008 and his Ph.D. from Tokyo Institute of Technology, Japan, in 2015, respectively. He is a member of IPSJ Japan and IEEE. His research interests lie in the area of watermarking, network security, and computer vision.
Email: thanhtm@lqdtu.edu.vn.

GIẤU TIN THUẬN NGHỊCH SỬ DỤNG PHƯƠNG PHÁP CHIA SẺ BÍ MẬT DNA-XNOR

Đỗ Huy Cường, Phạm Thái Hưng, Tạ Minh Thanh

Tóm tắt

Trong bài báo này, chúng tôi đề xuất một giải pháp giấu tin dựa trên phương pháp chia sẻ bí mật sử dụng DNA exclusive-NOR (DNA-XNOR) cho ảnh màu. Phương pháp chia sẻ bí mật DNA-XNOR được dùng như một ma trận khoá bí mật để giấu và trích xuất thông tin. Chúng tôi sử dụng sai số trung bình bình phương (MSE) để lựa chọn giá trị nhúng tối ưu cho từng điểm ảnh. Giá trị này sẽ được sử dụng để nhúng vào ảnh từ đó thu được ảnh sau giấu thông tin có giá trị PSNR cao nhất. Trong giải pháp của chúng tôi, thông tin mật sẽ được nhúng trên cả ba kênh màu Red (R), Green (G), và Blue (B) của ảnh gốc. Phép toán đề xuất DNA-XNOR là phương pháp mới để chia sẻ thông tin mật trong lĩnh vực giấu tin. Kết quả thực nghiệm của chúng tôi cho thấy hiệu quả của thuật toán và chứng minh khả năng ứng dụng trong thực tế.