A novel data hiding method based on deoxyribonucleic acid coding

Hongjun Liu a,*, Da Lin b, Abdurahman Kadir c

a School of Information Engineering, Weifang Vocational College, Weifang 261041, China
b School of Automatic and Electronic Information, Sichuan University of Science and Engineering, Zigong 643000, China
c School of Computer Science and Engineering, Xinjiang University of Finance and Economics, Urumqi 830012, China

ABSTRACT

The paper proposes a new data hiding method based on deoxyribonucleic acid (DNA) coding, using the Word document as carrier. The plain message becomes a cipher sequence after being encoded to a DNA sequence and encrypted by the addition operation. The cipher sequence is attached to a random DNA primer sequence and circularly shifted for finite times, then hide the whole sequence into a Word document through substituting each character's color. The plaintext can be extracted according to the keys, and the key space is large enough to resist brute force attacks. Experimental results show the feasibility of the scheme.

1. Introduction

Information security plays an important role in many fields, especially in confidential business and military affairs. In order to transmit secret information, the plaintext is usually encrypted, and embedded into various types of multimedia file as cover media, such as image, audio, video, Hypertext Markup Language (HTML), Word and PowerPoint files [1–3]. The security of data hiding techniques has attracted increasing research attention [4,5].

Nowadays PDF file [6] is widely used to transmit information, for the content of PDF file cannot be modified directly, so using it as carrier of cipher text is suitable. Zhong et al. use integer numerals specify the positions of the text characters in a PDF file to embed secret data, and take some measures to resist passive attacks [7]. Lee and Tsai encode the plaintext by a special ASCII coding, and then embed them into the locations between words and characters in PDF file [8]. Both of the literatures did not give the security analysis in detail, and their key spaces are very limited too.

In recent years, the research on chaos has attracted much attention. Due to some interesting intrinsic features, such as ergodicity, sensitive depending on initial conditions, the chaotic systems are widely applied to generate pseudo-random sequence to encrypt the data, for chaos-based algorithms have higher security and complexity than the classical encryption algorithm [9]. Kanso and Ghebleh [10] designed an image encryption scheme, a 3-D chaotic map is applied to mix and mask the shuffled pixels to resist differential and causality attacks. Seyedzadeh and Mirzakuchaki [11] proposed a fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic maps, the brilliant characteristics are high security, sensitivity and speed.

Recently, the characteristics of DNA computing, massive parallelism, huge storage and ultra-low power consumption have been found [12,13]. Some researchers turned to use the complementary rule of DNA to encrypt image. Gehani et al. [14] presented an image encryption algorithm of one-time pad cryptography with DNA strands. Zhang et al. [15,16]
proposed two Image encryption algorithms using the exclusive or (XOR) operation, bit shift and DNA addition and subtraction operation.

In this paper, a new data hiding method by embedding secret message into Word document is proposed. Firstly, we encode the plaintext by DNA coding to a DNA sequence. Secondly, we encrypt the DNA sequence by another equal-length DNA sequence generated by Chebyshev maps, and attach the result to a primer DNA sequence. After circularly shifting the whole sequence for finite times, we embed them into the Word document by modifying the fore color of the characters. Each character in the Word document can be embedded one character, i.e. 6-bit DNA coding. The initial values and parameter of the Chebyshev maps and shifting times are all served as keys, and the plaintext can be extracted successfully from the host document.

The rest of the paper is organized as follows. Section 2 presents the DNA coding. How to generate two aided pseudo-random sequences by Chebyshev maps are proposed in Section 3. The message hiding and recovery algorithms are designed in Section 4. Experimental results are given in Section 5, and the performance and security analysis are proposed in Section 6. The paper is concluded in Section 7.

2. DNA coding

A DNA sequence contains four nucleic acid bases of A (adenine), C (cytosine), G (guanine) and T (thymine), where A and T, C and G are complementary pairs [15]. In the binary system, 0 and 1 are complementary pair, and 0 (00) and 3 (11), 1 (01) and 2 (10) are complementary pairs too. So in the total 4! = 24 kinds of coding, there are only 8 of them can meet the complementary rule, i.e.: 0123 can be expressed as CTAG, CATG, GTAC, GATC, TCGA, TGCA, ACGT, and AGCT. Here we randomly select one of them to encode the plaintext.

As an alternative encoding to the traditional binary encoding, the DNA coding proposes a novel encoding method. Nucleotides are programmed to be a quaternary code, and each character can be denoted by three nucleotides. The DNA coding length is 6-bit, which is shorter than the classical 8-bit ASCII coding.

Clelland and Risca proposed the translation table from alphabets to DNA nucleotides [17], as shown in Table 1.

If we use DNA nucleotides of C, A, T and G to denote 0 (00), 1 (01), 2 (10) and 3 (11) respectively, we can use CGA (001101) to denote the letter "A", and use CCA (000001) to denote the letter "B", etc. Then, the plain message M can be encoded to the DNA sequence MDNA, for example, "AB" can be expressed as CGACCA (001101000001). We can get the plaintext by decoding the DNA sequence.

Although the DNA coding can only express capital letters, numbers, and several punctuation marks, it is a novel coding mode and sufficient to encode the secret message in secure communication. The alphabet is composed of three nucleotides, for each nucleotide can be chosen from four nucleotides of C, T, A and G, there are total 4^3 = 64 kinds of permutation. In Table 1, the letter "A" has 64 kinds of possible expressions, the letter "B" has 63 kinds of possible expressions, etc., finally, the last punctuation mark “-” has 21 kinds of possible expression, so there are total \( S_c = P_{42}^{41} \) kinds of combination for the translation from alphabets to DNA nucleotides, and Table 1 shows only one of them.

3. Generate two aided pseudo-random sequences by Chebyshev maps

3.1. Chebyshev maps

The expression of Chebyshev maps is shown as follows:

\[ z_{i+1} = \cos(w \cos^{-1} z_i), \quad -1 \leq z_i \leq 1, \]

where \( w \) is the degree of Chebyshev maps.

Chebyshev maps have important properties of excellent cryptosystem [18]. If \( w \in [2,6] \), the Lyapunov exponent of Chebyshev maps is positive, as shown in Fig. 1, which predicates that Chebyshev maps are chaotic. The real number sequences generated by Chebyshev maps are orthogonal polynomial sequences. Furthermore, their correlation functions are all \( \delta \) function.

| A = CGA | H = CGC | O = GCC | V = CCT | 2 = TAG | 9 = CGG |
| B = CCA | I = ATG | P = GCA | W = CCG | 3 = GCA | –ATA |
| C = GGT | J = AGT | Q = AAC | X = CTA | 4 = GAG | &TGC |
| D = TTG | K = AAG | R = TCA | Y = AAA | 5 = AGA | –GAT |
| E = GGT | L = TGC | S = ACG | Z = AAT | 6 = GGG | &GCT |
| F = ACT | M = TCC | T = TTC | 0 = TTA | 7 = ACA | –ATT |
| G = TTT | N = TCT | U = CTG | 1 = ACC | 8 = AGG | –ATC |
3.2. Generating two pseudo-random sequences

After encoding the plaintext, we can generate two random aided DNA sequences by Chebyshev maps to encrypt and conceal the cipher text. Shannon has proved that one-time keys is the most secure algorithm\[19\], and Leier et al. proposed the cryptography with DNA binary strands\[20\], here we use the one-time keys in the hiding and recovery algorithms to ensure the security. Firstly we use the method proposed in Ref.[21] to generate the 2D true random number one-time keys stream sequence \(S_{\text{Keys}}\).

\[
S_{\text{Keys}} = \{(w_1, z_1), (w_2, z_2), \ldots, (w_n, z_n)\}.
\]

Before the data hiding process, we will generate two sequences of \(X_{\text{XOR}}\) and \(Y_{\text{Primer}}\) by Eq. (1). Each time we randomly select two groups from \(S_{\text{Keys}}\), such as \((w_i, z_i)\) and \((w_j, z_j)\), to serve as the initial values and parameters for Chebyshev maps.

The first sequence \(X_{\text{XOR}} = \{x_1, x_2, \ldots, x_{LM}\}\) is employed to encrypt the DNA coding sequence \(M_{\text{DNA}}\), which is converted from the plaintext \(M\), the aim is to resist the chosen plaintext attack and the known plaintext attack. After iterating Eq. (1) for \(m\) times, we begin to generate \(X_{\text{XOR}}\) by Eq. (3).

\[
x_i = \begin{cases} 
00. & (z_i > -1) \text{ and } (z_i < -0.5), \\
01. & (z_i > -0.5) \text{ and } (z_i \leq 0), \\
10. & (z_i > 0) \text{ and } (z_i < 0.5), \\
11. & (z_i > 0.5) \text{ and } (z_i < 1). 
\end{cases}
\]

The second sequence \(Y_{\text{Primer}} = \{y_1, y_2, \ldots, y_{LW-15}\}\) is served as the primer DNA sequence, where \(L_M\) denotes the length of the plaintext, and \(L_W\) denotes the number of characters in the Word document, \(L_W > L_M\).

After iterating Eq. (1) for \(n\) times, we begin to generate \(Y_{\text{Primer}}\) by Eq. (4).

\[
y_i = \begin{cases} 
00. & (z_i > 0.5) \text{ and } (z_i < 1), \\
01. & (z_i > 0) \text{ and } (z_i \leq 0.5), \\
10. & (z_i > -0.5) \text{ and } (z_i \leq 0), \\
11. & (z_i > -1) \text{ and } (z_i \leq -0.5). 
\end{cases}
\]

4. Message hiding and recovery algorithms

4.1. Encrypt the DNA coding sequence

With the rapid development in DNA computing, some biology operations and algebraic operations based on the DNA sequence have been proposed by researchers\[22,23\], such as the addition operation and subtraction operations, which are just like the XOR operation. For example, \(11 + 10 = 01, 01 - 11 = 10\). In Ref. [17], Zhang et al. use 00, 01, 10 and 11 to denote C, A, T and G respectively, the details of the addition and subtraction rules are shown in Tables 2 and 3.

We perform the addition operation to encrypt the DNA coding sequence \(M_{\text{DNA}}\), and use the subtraction operation to decrypt it, as shown in the following equations:
Table 2
Addition operation for DNA sequence.

<table>
<thead>
<tr>
<th>*</th>
<th>T</th>
<th>A</th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>C</td>
<td>G</td>
<td>T</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>T</td>
</tr>
<tr>
<td>C</td>
<td>T</td>
<td>A</td>
<td>C</td>
<td>G</td>
</tr>
<tr>
<td>G</td>
<td>A</td>
<td>T</td>
<td>G</td>
<td>C</td>
</tr>
</tbody>
</table>

Table 3
Subtraction operation for DNA sequence.

<table>
<thead>
<tr>
<th>-</th>
<th>T</th>
<th>A</th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>C</td>
<td>G</td>
<td>T</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>T</td>
</tr>
<tr>
<td>C</td>
<td>T</td>
<td>A</td>
<td>C</td>
<td>G</td>
</tr>
<tr>
<td>G</td>
<td>G</td>
<td>T</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>

\[
M'_{\text{DNA}} = M_{\text{DNA}} + X_{\text{XOR}}. \tag{5}
\]

\[
M_{\text{DNA}} = M'_{\text{DNA}} - X_{\text{XOR}}. \tag{6}
\]

Here \(M'_{\text{DNA}}\) denotes the result of addition operation, and the symbols of “+” and “−” denote the addition operation and subtraction operation for DNA coding sequence.

4.2. Data hiding algorithm

A bit can contain different amount of information depending on its position in the pixel. For example, “1” at the 8th bit of a pixel represents \(128 \times 2^7\), but it only represents \(1 \times 2^0\) at the first bit. According to Eq. (7), the higher four bits (8th, 7th, 6th and 5th) carry 94.125% of the total information, and on the other hand, the lower two bits (2nd and 1st) carry only 1.1765% of the total information, so it is usually used to hide information. The percentage of the pixel information is shown in Table 4.

\[
p(i) = \frac{2^{i-1}}{255}, \quad i = \{1, 2, \ldots, 8\}. \tag{7}
\]

The principle for hiding the DNA sequence is that, for each character in the Word document, its fore color can be decomposed of three components of red, green, and blue. The grayscale value of each component is between 0 and 255, and can be expressed as an 8-bit binary number. Each character of the plaintext can be expressed as three DNA nucleotides, i.e. a 6-bit binary number. For each character in the embedding location, we convert its fore color to three 8-bit binary numbers, and substitute their least significant 2-bit respectively to embed into the 6-bit binary number.

The algorithm is designed to ensure the plaintext to be embedded into the characters, and can also skip the non-character contents in the Word document, such as image and object, the same to the extracting process.

If we hide the cipher text sequence \(M'_{\text{DNA}}\) in the Word document by selecting the start location randomly, the start location can be easily detected by the color analysis, so we take two measures to modify the fore color of all the characters.

Firstly, attach \(M'_{\text{DNA}}\) to a primer sequence \(Y_{\text{Primer}}\) to get sequence \(M'Y\), we set the total length of \(M'Y\) equal to the number of characters in the Word document, so the fore color of all the characters will be substituted. Secondly, circularly shift \(M'Y\) to the right or left for \(sn \in [1,2000]\) times, then embed the whole sequence into the Word document through substituting each character’s fore color. The embedding process for one character in the plaintext is showed in Fig. 2.

Table 4
Percentage of pixel information contributed by different bits.

<table>
<thead>
<tr>
<th>Bit position i in the pixel</th>
<th>Percentage (p(i)) of the pixel information (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3922</td>
</tr>
<tr>
<td>2</td>
<td>0.7843</td>
</tr>
<tr>
<td>3</td>
<td>1.5686</td>
</tr>
<tr>
<td>4</td>
<td>3.1373</td>
</tr>
<tr>
<td>5</td>
<td>6.275</td>
</tr>
<tr>
<td>6</td>
<td>12.53</td>
</tr>
<tr>
<td>7</td>
<td>25.10</td>
</tr>
<tr>
<td>8</td>
<td>50.20</td>
</tr>
</tbody>
</table>
Fig. 2. The embedding process for one character in plaintext.

Fig. 3. The flow chart of data encryption and hiding algorithm.
If we convert the Word document to its corresponding PDF file, the changes of fore color can still be remained and cannot be directly modified, so the PDF file can also be served as the host file.

For the sender, the data encryption and hiding algorithm can be divided into the following steps, and the flow chart is shown in Fig. 3.

Step 1: Get the length $L_M$ of the plaintext $M$, and the number of characters $L_W$ in the Word document $W$.

Step 2: Encode $M$ to the DNA sequence $M_{DNA}$ by DNA coding.

Step 3: Generate two pseudo-random DNA sequences of $X_{XOR}$ and $Y_{Primer}$ by the Chebyshev maps with one-time keys from sequence $S_{Keys}$.

Step 4: Set $M'_{DNA} = M_{DNA} + X_{XOR}$.

Step 5: Attach $M'_{DNA}$ to the right of $Y_{Primer}$, to get the sequence $M'Y$, then circularly shift it to the left or right for $sn$ times to get the DNA sequence $E$, the shift times $sn \in [1,2000]$.

Step 6: Embed $E$ into the Word document $W$ through substituting each character’s fore color.

Step 7: Send the Word document to the receiver directly, or convert it to a PDF file firstly.

4.3. Data extracting and recovery algorithm

The sender can send the Word document or PDF file to the receiver. If the receiver receives the Word document, he can extract the plain message from the Word document directly, if he receives the PDF file, he can also convert the PDF file to Word document firstly, then extract the plain message.

The keys can be transmitted from the sender to the receiver through a secure channel. According to the same keys, the receiver can extract the plaintext. The steps are shown as follows, and the flowchart is shown in Fig. 4.

Step 1: For the Word document, go to Step 2. For the PDF file, copy the contents to a new Word document firstly.

Step 2: Extract the DNA sequence $E$ from the least 2-bit of the three color components of each character.

Step 3: Circularly shift $E$ to the reverse direction for $sn$ times to get the sequence $E'$. According to the length $L_M$ of $M$ to extract the sequence $M_{DNA}$.

Step 4: Generate the pseudo-random chaotic sequence $X_{XOR}$ by the Chebyshev maps with the same keys as the hiding process.

Step 5: Set $M'_{DNA} = M_{DNA} - X_{XOR}$.

Step 6: Decode $M_{DNA}$ by the DNA coding to get the plaintext $M$.

5. Experimental results

The experimental environment is Visual Basic 6.0, Matlab 7.0 and Word 2003 under the operating system of Microsoft Windows XP. We use several versions of the Adobe Reader, including the versions of 6.0 Professional, 7.0 Professional, 8.1.2, and 9.4.0, to open the PDF file.
Here we set the secret message “JUNE 6 INVASION: NORMANDY” as the plaintext $M$. According to the coding in Table 1, we encode $M$ to get $M_{DNA}$, as shown in Fig. 5.

Here we use $C, A, T, G$ to denote 0 (00), 1 (01), 2 (10), 3 (11), and use the keys ($w_1 = 4.2991289$, $z_1 = 0.2182135678932276$), and ($w_2 = 5.1682892$, $z_2 = 0.7182225678934565$) from $S_{Key}$ to get $X_{opt}$ and $Y_{primer}$.

The screenshots of two Word documents before and after hiding the sequence $E$ are shown in Fig. 6a and b, their difference is shown in Fig. 6c and d is the negative of Fig. 6c when we set $\gamma = 0.02$. For the changes in the least significant 2-bit may be “00”, “01”, “10” or “11”, even the tiny changes can be detected, if the keys are unknown, it is still impossible to extract the secret data.

(a) The screenshot of Word document of before hiding (b) The screenshot of Word document after hiding (c) The difference between (a) and (b) (d) The negative of (c) when set $\gamma = 0.02$

Fig. 6. The screenshots of PDF files and their difference.
6. Performance and security analysis

6.1. The variation tendency of the host file sizes

Both the Word document and the PDF file can all be served as the host file, here we test the growth ratio of them with the number of hidden characters. The results are shown in Figs. 7 and 8, from the charts we can find that their growth ratios slowly increase with the number of hidden characters, the variation tendency is nonlinear.

6.2. The comparison of the hiding capacity

About the hiding capacity of the proposed method, for each character in the Word document can be hidden in a 6-bit binary DNA coding, i.e. a character encoded by the DNA coding, so the ratio of embedding capacity of our method is 100%. We compare the hiding capacity with the algorithms proposed by Refs. [7,8], and the results are shown in Table 5. From the table we can find that the proposed hiding capacity is higher than the previous algorithms.

If more than least significant 2-bit are substituted, the visual quality will be decreased with the increase of hiding capacity.

6.3. The key space

We assume that the algorithm is known to the public, so the solution to security is the keys. Our algorithm actually does have some of the following keys: ① the total combinations \( S_c \) of translation from alphabets to DNA nucleotides, ② initial
condition \((w_i, z_i)\) for Chebyshev maps, \(^3\) iteration times \(m\) for Chebyshev maps, \(^4\) the shift direction, and the shift times \(sn\), \(^5\) the length of plaintext \(Lm\). In the receiver, the group \((w_i, z_i)\) and iteration times \(n\) are useless, so they are not served as keys.

There are total \(S_e = P_{64}^{2^{10}} = 5.2 \times 10^{19}\) kinds of combination of translation from alphabets to DNA nucleotides. Only the group \((w_i, z_i)\) serves as the key, the variation of the parameter \(w\) in the chaotic region is between 2 and 6 with a step of \(10^{-7}\), so \(S_{w0} = 4 \times 10^5\). For any chaotic system is sensitive depending on the initial conditions, even the initial value \(z_i\) is changed with a tiny step of \(10^{-10}\), their orbits of the Chebyshev maps will be completely different, so we set the key space for initial value \(z_i\) to \(S_z = 10^{16}\) [18].

We set the iteration times \(m \in [100, 1000]\), so \(S_m = 9 \times 10^2\), the shift direction is left or right, so \(d = 2\). We set the shift times \(sn \in [1, 2000]\), so \(S_{sn} = 2 \times 10^3\). The length of plaintext \(Lm \in [1, 1000]\), then \(S_{Lm} \approx 10^9\).

The estimation of the key space \(S = dS_{w0}S_{sn}S_mS_{sn}S_{Lm} \approx 3.744 \times 10^{103}\), which is much larger than \(2^{100}\). Alvarez and Li suggested that the key space should be at least \(2^{100}\) for a sufficient security level against brute-force search attacks [24]. Our cryptosystem has fulfilled this requirement, so it is large enough to make any brute force attack ineffective.

6.4. Possible attacks

If the Word document is served as the carrier of secret message, the active attacker can only get the DNA coding sequence \(E\), but cannot decode it. For the active attacker, we can compute the message digest 5 (MD5) hash value of the Word document or the PDF file and send it to the receiver by secret channel, if these files are modified by attackers, their MD5 hash values will be different completely [25].

For the passive attacker, we have the large key space to resist brute force attacks. Even though he gets the whole ciphered DNA sequence, it is unable to decide where the starting location of the secret message is, and how to decrypt it.

7. Conclusion

In this paper, we propose a novel data hiding method based on DNA coding and using Word document as host file. Three measures are taken to enhance the robust. Firstly, we encode the plaintext using DNA coding instead of using 8-bit ASCII coding, to shorten the cipher size. Secondly, we generate two random aided DNA sequences to encrypt and conceal the cipher sequence. Finally, we hide the cipher sequence into a Word document by substituting the least significant 2-bit of the three color components. No matter the carrier file is Word Document or PDF file, the secret message can be extracted successfully. The security analysis shows that the key space of the proposed algorithm is large enough to resist all kinds of brute force attacks. Therefore, we can conclude that the proposed method is well suited to some data hiding applications, such as fragile watermarking, secret communication, and online content distribution systems.

Acknowledgements

The research is supported by the Science and Technology Development Plan of Weifang (Nos. 20121346, 20122046), the Youth Foundation of Sichuan Provincial Education Department (No. 11ZB097), the Talents Project of Sichuan University of Science and Engineering (No. 2011RJ07), the Key project of Artificial Intelligence Key Laboratory of Sichuan Province (No. 2011RJ02), the Minority Nationality Technology Talent Cultivation Plan of Xinjiang (No. 201123116).

References


Hongjun Liu was born in Shandong, China, in 1970. He received his M.S. degree in computer application technology from Ocean University of China, Shandong, China, in 2005. He is currently an Associate Professor at the Weifang Vocational College. His research interests include chaos, image encryption and secure communication.

Da Lin was born in Shandong, China, in 1974. He received his M.S. degree from Jilin University in 2007. He received his Ph.D. degree in Biomedical engineering from Dalian University of Technology in 2011. He is currently an Associate Professor at the Sichuan University of Science and Engineering. His research interests include biomedical information, and chaos.

Abdurahman Kadir was born in Urumqi, China, in 1975. He received his M.S. degree from Dalian University of Technology in 2007. He received Ph.D. degree from Dalian University of Technology, Liaoning, China, in 2011. He is currently an Associate Professor at the Xinjiang University of Finance and Economics. His research interests include information security, and chaos.