Data Security And Cryptography Based On DNA Sequencing

Nirmalya Kar, Atanu Majumder, Ashim Saha, Suman Deb
Dept. of Computer Science & Engineering
National Institute of Technology, Agartala
Jirania, Tripura, INDIA
nirmalya@nita.ac.in,
atanu.cse21@gmail.com,
ashim.nita@gmail.com,
sumandebs@gmail.com

Dr. Mukul Chandra Pal
Dept. of Mathematics
National Institute of Technology, Agartala
Jirania, Tripura, INDIA
palmukalme@gmail.com

Abstract:

Cryptography is one of the major concerned areas of computer networks and data security. The art of cryptography is to convert a plain message into a human unreadable format by encoding message using some cryptographic algorithms. In today’s world, security is one of the most significant and fundamental issues of data transmission, researchers are working on the evolvement of new cryptopgraphic algorithms. Cryptography is the process of providing security of data transmission via public network by encrypting the original data or message. Data security is concerned with the areas of data transmission. An efficient direction of providing data security can be termed as DNA based Cryptography. The encryption and decryption process proposed in this paper will use the DNA sequencing property of the DNA. We have proposed here how the DNA sequencing can be utilized in cryptographic algorithms and how the message can be made more secure and reliable for transmitting effectively via networks. The algorithm we have proposed here is a new algorithm which has two rounds of encryption. Moreover paper contains a new method of key generation and sharing for better security aspects. The encryption scheme is designed by using the mechanism of DNA sequencing.

Keywords- DNA; DNA Sequencing; Encryption; Decryption and Secret key.

I. Introduction

In order to obtain complex computation in the process of achieving the cipher text recent trends are focused on DNA computing and DNA based encrypting algorithms. One of widely used process of secret writing is called cryptography which provides data and information security and protects that information from several malicious attacks. Cryptography is defined as the process of transforming a known text or plaintext into a coded form that is a human unreadable format called cipher text by encrypting the original message using some cryptographic algorithms. Security is concerned with the protection and providing security on network and data while transmitting over the network. But to achieve complete security against attacks is a challenging issue of data communications. The conventional methods of encrypting are not strong enough today for providing the
data security and reliable data transmission. Unauthorized user or intruders may attack and can interrupt or intercept the message for doing some malicious tasks. In order to enhance the data security effective encryption algorithms are required.

Recent research has shown DNA as a medium for large scale computation system. One potential key application of large scale computation system is DNA based cryptography. A large number of researcher take an initiative for implementing DNA encoding concept in the applications like cryptography, scheduling, clustering, forecasting and even trying to apply this in signal and image processing application [2]. From few years back, most of the research works have been going on DNA based encryption schemes. Biological properties of DNA sequences are used in almost of the cryptographic works. In this paper, we have proposed a new technique where biological properties are not directly used. Instead, we have used different properties of DNA sequences in our proposed encryption scheme [2]. A DNA sequence is a sequence consisting of four DNA bases namely: A, C, G and T [11]. Each of the bases is related to a nucleotide. There are a large number of DNA sequences publicly available in various domains of bio-logical DNA. A rough estimation would put the number of DNA sequences publicly available in various web sites are around to be 55 million [2].

II. Prepare DNA Bio-Molecular Computation

Before For any organism in the biosphere, DNA is the blue print of genetic information. Bio-logical DNA comprising of two strands of nucleotides, each of these two are coded with four bases namely A, T, C, G (A – adenine, G – guanine, C – cytosine, T – thymine) [1, 3]. A DNA molecule has formed by two single-stranded DNA which is the double-stranded structure, bonded each other by hydrogen bonds: A with T and C with G. The DNA strands that bond each A with T and each C with G are called the complementary pair of DNA strands [5, 11].

III. Proposed System

Finally, For providing better security and reliable data transmission, a new method of encryption process is proposed here. This proposed algorithm is works on the binary values of the message or plaintext. In this algorithm, binary values or bits are read from the plaintext. A session key is shared through a secure channel between sender and receiver prior to communication establishment. This key bears the information about the key that is used for encrypting the message. The round 1 key for encryption is computed based on the response of a random number generator and the information about the key is send to the receiving side through a private channel [8]. Sender will use a random number generator to generate a random number, and then this number along with the shared secret key will go through a function that will produce round 1 encryption key (KE). The same function will generate the session key as the information of the key that is being used. Then each 8-bit block of plaintext will go through the round 1 encryption by round 1 encryption key (KE) using cipher block chaining methods. The output of one block will be used as the key for the next block.

In encryption round 2, sender will select a DNA sequence randomly from publicly available DNA sequences [2, 9]. This DNA sequence is one of the key of encryption round 2. Receiving side must have the information about the used DNA sequence. Then this selected DNA sequence will be converted into binary string using binary coding scheme. This binary string is then segmented into 8-bit blocks, and then each of the 8-bit blocks will be transform to hexadecimal form. Each block of the intermediate cipher text is then converted
into hexadecimal form as well. When the length of DNA sequence is less than the length of the intermediate cipher text, the DNA sequence will be repeated. And when the case is reversed then the extra bit from the DNA sequence will be removed. After that hexadecimal addition will be performed between the intermediate cipher text blocks and the each blocks of selected DNA sequence. Then the result of each block will be transform again to binary and then to a faked DNA sequence using binary coding scheme of DNA [2]. The extra bits will be added both at the beginning and at the end of the newly generated DNA sequence as well, which will produce the final cipher text. This final cipher text has extra information including starting and ending primers that is not linked up with the original message like as biological DNA strand containing introns as extra information which are omitted by splicing process of biological DNA synthesizing. Actual cipher text that will be created using encryption algorithms and keys is hidden in the final cipher text.

![Flow diagram of the proposed algorithm](image)

Figure 1. Flow diagram of the proposed algorithm

IV. Procedure For Sharing Key

A. Round 1 Session Key Generation and Sharing

Define A common secret key (PK) is shared between sender and receiver prior to communication. (16-bit).

1) Sending End Computations:

Step 1: Sender will use a random number generator and select one random number which is of 16-bit (Rn).

Step-2: Divide the random number ‘Rn’ into 2 parts each of having 8-bit. (Rn_L and Rn_R)

Step-3: Divide the shared secret key into 2 parts as PK_L and PK_R. Both of these are of 8-bit.

Step-4: Now PK_L will get X-OR with Rn_R and PK_R will get X-OR with Rn_L.

\[ RL = PK_L \oplus Rn_R \]
\[ RR = PK_R \oplus Rn_L \]

Step-5: Both of the results will be further sub-divided into 2 parts namely R_L1, R_L2, R_R1, R_R2 having 4-bit each.
Step-6: Make 4-bit EX-OR operation between $R_{L1}$ and $R_{R2}$ and between $R_{L2}$ and $R_{R1}$.

$$T1 = R_{L1} \oplus R_{R2} \quad T2 = R_{L2} \oplus R_{R1}$$

Step-7: Concatenate these 4-bit results, T1 and T2, which will give Encryption key of 8-bit for round 1.

$$K_e = K_1 = \text{concate}(T1, T2)$$

Step-8: Session key computation for round 1:

Compute, $\text{tmp} = PK \oplus Rn$

Step-9: Divide ‘tmp’ by 16 and convert the remainder into its equivalent hex form and keep it in ‘$K_{S1}$’

Divide the result once again by 16 keeping hex form of the remainder in ‘$K_{S2}$’ ....

Do until the result is less than 16 ($K_{S3}, K_{S4} \ldots \ldots K_{Sn}$).

Step-10: Make together all the ‘$K_{Ss}$’ in order to get the round 1 session key KS

Step-11: Send ‘KS’ as round 1 session key through a secure channel along with round 2’s session key.

End

---

Session key computation:

$\text{tmp} = PK \oplus Rn = 49429 \oplus 6255 = 55674$

$K_{S1} = 55674 \% 16 = 10 = \text{A (hex form)}$

$\text{tmp} = 55674 / 16 = 3479$

$K_{S2} = 3479 \% 16 = 7 = \text{7 (hex form)}$

$\text{tmp} = 3479 / 16 = 217$

$K_{S3} = 217 \% 16 = 9 = \text{9 (hex form)}$

$\text{tmp} = 217 / 16 = 13$

$K_{S4} = 13 = \text{D (hex form)}$ Session key, $K_{S} = \text{A79D}$

2) Receiving End Computations:

Input: Shared secret key ‘PK’ and session key ‘$K_{S}$’

Step-1: Separate all the digits of ‘KS’ and convert these into their equivalent decimal form.
e.g. $K_S = A79D$ (say), $K_{S1} = 10, K_{S2} = 7, K_{S3} = 9, K_{S4} = 13$

**Step-2:** Computation of decryption key as follows:

\[
\text{tmp} = K_{Sn}
\]

\[
\text{tmp} = (\text{tmp} \times 16) + K_{S_{n-1}}, \text{ continue upto } K_{S_1}
\]

Random number, $R_n = \text{tmp} \oplus PK$

e.g. $\text{tmp}_1 = (13 \times 16) + 9 = 217$

\[
\text{tmp}_2 = (217 \times 16) + 7 = 3479
\]

\[
\text{tmp} = \text{tmp}_3 = (3479 \times 16) + 10 = 55674
\]

\[
R_n = (49429 \oplus 55674) = 6255
\]

**Step-3:** Divide the random number ‘$R_n$’ into 2 parts each of having 8-bit. ($R_n^L$ and $R_n^R$) and divide the shared secret key into 2 parts as $PK_L$ and $PK_R$. Both of these are of 8-bit.

**Step-4:** Now $PK_L$ will get X-OR with $R_n^R$ and $PK_R$ will get X-OR with $R_n^L$.

\[
R_n^L = PK_L \oplus R_n^R \quad R_n^R = PK_R \oplus R_n^L
\]

\[
R_n^L = 193 \oplus 111 = 174 \quad R_n^R = 21 \oplus 24 = 13
\]

**Step-5:** Both of the results will be further sub-divided into 2 parts namely $R_{L1}, R_{L2}$ and $R_{R1}, R_{R2}$, having 4-bit each.

**Step-6:** Make 4-bit EX-OR operation between $R_{L1}$ and $R_{R2}$ and between $R_{L2}$ and $R_{R1}$

\[
T_1 = R_{L1} \oplus R_{R2} \quad T_2 = R_{L2} \oplus R_{R1}
\]

\[
T_1 = 10 \oplus 13 = 7 \quad T_2 = 14 \oplus 0 = 14
\]

**Step-7:** Concatenate these 4-bit results, $T_1$ and $T_2$, which will give Decryption key of 8-bit for round 1.

\[
K_1 = \text{concat}(T_1, T_2)
\]

Decryption key, $K_D = K_1 = \text{concat} (0111, 1110) = 01111110 = 126$

End.

**B. Key Sharing for Round 2**

Generate a number randomly that will be used as round 2 key for the second round of encryption process. This key will give the size of extra bit that is to be added with the cipher text, to make the cipher text more complicated to the intruders. Select a DNA sequence randomly from publicly available DNA sequences [2]. The round key 2 and the selected DNA sequence are to be sent at the receiver end prior to communication.
V. Algorithm

A. Procedure for Encryption

Round 1

Input: Sender’s message for the receiver that is plaintext

Output: Intermediate form of cipher text

Step-1: Enter the data or message that is to be sent.

Step-2: Read the bit values or binary form of the message into an array

Step-3: Round 1 key operation on the bit values taking one 8-bit block at once. Then perform bit wise EX-OR
operation between the plaintext blocks and the key in cipher block chaining method. That is one
cipher text block will be used as the key for the next block of plaintext for creating next block of
cipher text.

\[ IC_1 = KE(P_1) \]

\[ IC_2 = IC_1(P_2) \ldots \]

\[ IC_n = IC_{n-1}(P_n) \]

where \( IC_i \) - Intermediate cipher blocks, \( KE \) - Round 1 encryption key and \( P_i \) - Plaintext blocks

Step-4: By taking each ‘\( IC_i \)’ transforms it into 8-bit binary equivalent. Intermediate form of cipher text,

\[ IC = \text{concatenate}(IC_1, IC_2, \ldots, IC_n) \]

\[ \text{e.g. let the value of ‘}IC_i\text{’ is } 189 \text{ and } IC_j = 205 \]

\[ 189 = 10111101 \]

\[ 205 = 11001101 \] (May use binary encoding scheme here, 00=A, 11=T, 01=C, 10=G)

Intermediate Cipher(\( IC \)) = 1011110111001101

B. DNA Encoding

DNA has four kind of bases namely adenine (A) and thymine (T) or cytosine (C) and guanine (G) in
DNA sequence [2]. One simplest coding patterns for encoding the 4 nucleotide bases (A, T, C, G) is by means
of 2-bit binary: 0(00), 1(01), 2(10), 3(11). As it’s known to us that, in a double helix DNA string, two DNA
strands are held together complementary in terms of sequence, that is A to T and C to G according to Watson-
Crick complementarities rule. Taking DNA digital coding into account, it is required to reflect the biological
characteristics of 4 nucleotide DNA bases, the complementary rule that (\( \sim 0 = 1 \)), and (\( \sim 1 = 0 \)) is proposed in this
DNA digital coding [2]. As per this complementary rule, 3(11) is the complement of 0(00) and 2(10) of 1(01).
Thus A and T are correspond to ‘00’ and ‘11’ respectively and C and G to ‘01’ and ‘10’ respectively.
Substitution rule: A=00, T=11, C=01 and G=10.

Round 2
Input: Intermediate Cipher text IC. Randomly selected publicly available DNA sequences say $S_{qn}$.

Output: Final cipher text in the form of DNA sequence

IC= ‘1011110111001101’ $S_{qn} = AAGCTTAC$ (Let)

Step-1: Convert ‘$S_{qn}$’ into a binary sequence by using the binary coding scheme [3]. Thus the sequence ‘$S_{qn}$’ is ‘0000100111110001’

Step-2: Divide ‘$S_{qn}$’ into 8-bit block segments leaving the remainder part.

$S_{qn} = 00001001$ (first 8-bit). If the length of the DNA sequence is less than the length of IC, then this selected DNA sequence will be repeated.

Step-3: Taking 8-bit from both the intermediate cipher IC and DNA sequence, transform these into hexadecimal form.

Step-4: Perform hexadecimal add operation. If any carry is generated then discard that carry.

IC = 10111101, 11001101 $S_{qn} = 00001001, 11110001$

IC = BD, CD $S_{qn} = 09, F1$ (Hexadecimal form)

**Addition**

<table>
<thead>
<tr>
<th>B</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
</tr>
</tbody>
</table>

Step-5: Convert these hexadecimal results into their equivalent 4-bit binary and club together all. ‘1100011010111110’

Step-6: Perform DNA binary coding on this ‘1100011010111110’ that give rises to a faked DNA sequence ‘TACGGTTG’.

… (Repeat from step-3, until all the bits of the IC has come under DNA coding scheme)

Step-7: Add some extra information (starting and ending primers) at the beginning and at the ending as well of the faked DNA sequence and send the total form of DNA sequence as Cipher Text through public channel at the receiver end.

**C. Procedure for Decryption**

At the receiver end the cipher is decrypted by using the keys that have been shared between sender and receiver. The procedure of decryption is given as follows:

Round 1:

Step-1: Perform binary coding scheme on the cipher text to convert the DNA sequence into a binary string.

Step-2: Truncate the extra information bit (starting and ending primers) from the binary string of the cipher text. (The position of the extra information used as primers, is available at the receiving side)
Step-3: Compute the intermediate cipher(IC) text using shared DNA sequence

Step-4: Transform the shared DNA sequence into binary string using binary coding scheme. Then divide the binary string into 8-bit blocks and transform into equivalent hexadecimal form

S\textsubscript{qn} = AAGCTTAC, S\textsubscript{qn} = 00001001 11110001 (binary coding, 8-bit blocks),
S\textsubscript{qn} = 09, F1 (hex form)

Step-5: Taking 8-bit from the cipher text convert it into hexadecimal form.

e.g. CT = 'TACGTTTG', CT = 11000110, 10111110

and CT in hexadecimal form = C6, BE

Step-6: Perform hexadecimal subtraction of S\textsubscript{qn} from CT blocks ignoring the borrow

Subtraction

\[
\begin{array}{c|c|c|c|}
\text{C} & \text{6} & \text{B} & \text{D} \\
\hline
\text{0} & \text{9} & \text{1} & \text{1} \\
\end{array}
\]

Step-7: By taking together all the 8-bit sub-strings after round 1 decryption, a long string is generated that is the intermediate cipher text (IC).

Round 2:

Step-1: Divide the intermediate cipher text (IC) into 8-bit blocks.

Step-2: Each block of ‘IC’ will go through the round 2 decryption algorithm. The key used here is K\textsubscript{D}.

Decryption process is as follows:

P\textsubscript{i} = K\textsubscript{D}(IC\textsubscript{i})
P\textsubscript{2} = IC\textsubscript{1}(IC\textsubscript{2}) ....
P\textsubscript{n} = IC\textsubscript{n-1}(IC\textsubscript{n})

where IC\textsubscript{i} - Intermediate ciphertext block, K\textsubscript{D} - Round 2 decryption key and P\textsubscript{i} - Plaintext blocks

Step-3: Put together all the bits of each Pi, which give rises to the binary form of the sender’s message that is the plaintext.

Step-4: Write the binary values into a file and save the file according to the file type.

VI. Conclusion

The encoding method we have proposed in this paper is based on a new concept of conventional symmetric key cryptography methods. Rather than sharing the actual keys directly between the sender and receiver, session keys are shared between them that bears the information about the encryption keys. In this paper, we have utilized the DNA sequences properties of DNA for encrypting the message. As there are many
web-sites and approximately 55 million publicly available DNA sequences, so it is almost impossible to predict this sequence. This concept is proposed to facilitate the understanding of the principles and some techniques of the newly born field of DNA based cryptography. In the research domain of cryptography, DNA computing and encoding techniques are growing up rapidly day by day and thus the proposal can be used to enhance the security of data with much more new derived concepts such as steganography, integrity, signature and authentication. This method is more reliable, efficient and it provides more power against certain attacks. Extra information that contained in the cipher text that are not the part of actual cipher text, provides more security to the algorithm. In this paper, we have designed an encryption scheme by using the technologies of DNA synthesis. The extra bit and the faked DNA sequence which are present in the cipher text make the message more secure from intruders. It will be more difficult for intruders to predict the main cipher text for cryptanalysis as cipher text contains extra bit of information. The improved concept that is proposed in this paper can be used in the security concerned of real time security of distributed network systems.

REFERENCES


