A Spatial Crypto Technique for Secure Data Transmission

Sk. Sarif Hassan[1], Pabitra Pal Choudhury[2] and Soumalya Ghosh[3]

[1]*Department of Mathematics, Pingla Thana Mahavidyalaya, India*

[2]*Applied Statistics unit, Indian Statistical Institute, India*

[3]*School of Computing Science & Engineering, Galgotias University, India*

Submission: April 04, 2017; **Published:** August 10, 2017

*Corresponding author: Sk Sarif Hassan, Department of Mathematics, Pingla Thana Mahavidyalaya, India; Email: sarimif@gmail.com

Abstract

This paper presents a spatial encryption technique for secured transmission of data in networks. The algorithm is designed to break the ciphered data packets into multiple data which are to be packaged into a spatial template. A secure and efficient mechanism is provided to convey the information that is necessary for obtaining the original data at the receiver-end from its parts in the packets. An authentication code (MAC) is also used to ensure authenticity of every packet.

Keywords: Data Packet; Clustal W; Star model; Key packet

Introduction

Security of network communications is the most important issue in the world. Information transactions related to banks, credit cards, and government policies are transferred from place to place with the help of network transmission. The high connectivity of the World Wide Web (WWW) has left the world 'open'. Such openness has resulted in various networks being subjected to multifarious attacks from vastly disparate sources, many of which are anonymous and yet to be discovered. This growth of the WWW coupled with progress in the fields of e-commerce and the like has made the security issue even more important [1-7].

In practice, in a computer network, data is transferred across the nodes in the form of packets of fixed or variable sizes. In usual practice, the implementation is done by some secured algorithms at the application level on the data and the enciphered data is packetized at lower levels (in the OSI architecture) and sent. Any intruder able to obtain all the packets can then obtain the enciphered data by appropriately ordering the data content of each of these packets. Then, efforts can be made to break the secured algorithm used by the sender. In the process of transmission, if it is possible to prevent any information release as to the structure of the data within the packets, an intruder

would know neither the nature of the data being transferred nor the ordering of the content from different packets. This is what

our algorithm achieves by using a genomical spatial envelope. We have used genomic steganography in enhancing the security in the algorithm.

The algorithm

We use the concept of Message Authentication Code (MAC) as suggested in [Rivest, 1998] to authenticate messages. For a packet of data, the MAC is calculated as a function of the data contents, the packet sequence number and a secret key known only to the sender and the receiver, and then it is appended to the packet. On receiving a packet, the receiver first computes the MAC using the appropriate parameters, and then performs a check with the MAC attached to the packet. If there is no match, then the receiver knows that the packet has been tampered with. A detailed explanation is provided in the next section.

Let the size of an encrypted packet data in a network be denoted as PS. PS has a value of 1024- bits or 4096- bits typically. The packet data is now to break arbitrarily in N parts. These N parts are to be inserted to into the blank envelope. The size of total data, number N and the nucleotide sequences with which alignment is to be performed using ClastalW is communicated between the receiver and sender using a secure channel. The packet data size is represented in number of bits required to represent the size of total data sent. We use (N-1)*2 bits (or N-1 ATGC's) to represent packet data number.

Sender encryption technique: S(P)

I.   Transform the data P to a cipher data C by a crypto transform.

II.   Now using the transformation σ (defined as σ (00)=A, Ø (01)=c, σ(10)=G AND σ (11)=T), transform S to its A,T, C and G texture data E.

III.   Tear the data E into N packets of arbitrary size where Eijk and Eij(k+1) give together E_{ij} Similarly Eij and $E_{i(j+1)}$ give together Ei.

IV.   Size (size of data plus size of packet number) of each packet (which has also undergone all the above transformations) is inserted at the starting of each packet before the packet number.

V.   Now take two or three (sender choice) nucleotide sequences and perform alignment using ClastalW.

VI.   Wherever nucleotides are different for those two or three sequences in the alignment we replace that particular position by one (ATCG text) from the E_{ijk}... until it is exhausted.

VII.   After E_{ijk}... is exhausted we insert packet number at next positions where nucleotides are different.

VIII.   The remaining positions (if any) where nucleotides are different are replaced randomly by ATGC sequence.

IX.   Find MAC corresponding to the packet data D_{Eij}...

X.   Repeat Step 5, 6, 7 and 8 for each E_{ijk}... for all N parts. Each of these new parts is renamed as DE_{ijk}....

XI.   Send each DE_{ijk}... to the receiver in different time and different networks.

Receiver's decryption technique: R(DE)

In receiver end the following steps are required to decrypt the message:

i.   Receive each DE_{ijk}... from the sender in different time and different networks.

ii.   Perform a check on MAC corresponding to each packet. If satisfied proceed to next step.

iii.   Now take the nucleotide sequences (chosen and communicated by sender) and perform alignment using ClastalW.

iv.   Wherever nucleotides are different for those two or three sequences in the alignment we extract (ATCG text) from each DE at that particular position.

v.   Size of each packet (expressed in fixed number of ATGC's) is extracted from the starting of each packet.

vi.   Use the transformation σ' (defined as σ' (A)=00, σ' (C)=01, σ' (G)=10 and σ' (T)=11) on the data size.

vii.   Conserve the data packet up to the determined data size and reject the remaining ATGC's and rename the packet as E.

viii.   The packet number is extracted using σ' from the end of each packet and the packets are ordered and joined according to packet numbers i.e. E_{ijk} and Eɨj(k+1) give together E_{ij} Similarly E_{ij} and $E_{i(j+1)}$ give together Ei.

ix.   Now using the transformation σ' convert the A,T, C and G texture data into cipher text C.

x.   Transform the cipher data C to a data P by a crypto transform.

Demonstration of the algorithm

The plain text (P) is

I AM SUGATA SANYAL

The binary text (B) corresponding to (P):

010010010010000001000001010011010010000001010011010101010101000111010000010101010001000001001000000010100110100000101001110010110010100000101001100 (Size is 144)

The cipher text (C) corresponding to (B) The crypto map is T (0) =1; T (1) =0.

101101101101111101111101011001011011111101011001010101010111000101111101010101011101111101101111110101100101111101011000110100110101111101011001 1

We have to transfer the each data into its ATCG from by the following transformation: σ(00)=A; σ(01)=C, σ(10)=G and σ(11)=T

The transformed text (E) corresponding to (C) is:

GTCGTCTTGTTGGTAGTCTTGGTAGGGGGTGAGTTGGGGTGTTGTCTTGGTAGTTGGTACGGCGGTTGGTAT

Now we need to break the data E into arbitrarily N parts.

$E_{1=000001}$: GTCGTCTTGTTGGTAGTCTTGGT

(Size 23)

E_2: AGGGGGTGAGTTGGGGTGTTGTCTTGGTAGTTGGTACGGCGGTTGGTAT (Size 49)

$E_{21=001001}$: AGGGGGTG (Size 8)

How to cite this article: Sarif H, Pabitra P C, Soumalya G. A Spatial Crypto Technique for Secure Data Transmission. Biostat Biometrics Open Acc J. 2017: 2(4): 555592.

E_{22}: AGTTGGGGTGTTGTCTTGGTAGTTGGTACGGCGGTTGGTAT (Size 41)

$E_{221=101001}$: AGTTGGGGT (Size 9)

$E_{222=101010}$: GTTGTCTTGGTAGTTGGTACGGCGGTTGGTAT (Size 32)

Suppose we take the break data set as {E_1, E_{21}, E_{221}, E_{222}}

Now we add packet size (size of data and size of packet number) at the starting of each packet.

Since our data size before packet formation is 144 which can be represented in (10010000) 8 bits we use a 8 bits (4 ATGC's) representation of packet size.

Sizes of each packet are given below:

E_1: 26 (size of packet + size of packet number=23+3)

$(26)_{10}=(00011010)_2$

Using crypto map we get size as 11100101

Using σ size is represented as TGCC

E_{21}: 11 (size of packet + size of packet number=8+3)

$(11)_{10}= (00001011)_2$

Using crypto map we get size as 11110100

Using σ size is represented as TTCA

$E221$: 12 (size of packet + size of packet number=9+3)

$(12)_{10}= (00001100)_2$

Using crypto map we get size as 11110011

Using σ size is represented as TTAT

E_{222}: 35 (size of packet + size of packet number=32+3)

$(35)_{10}= (00100011)_2$

Using crypto map we get size as 11011100

Using σ size is represented as TCTA

Now we need to packetize the above data into star model (OR1D2, OR1D4 and OR1D5) ....

Note that we need to send the packet number along with this data packet. We insert the data content followed by the packet number.

Therefore the data packets are as follows:

DE_{000001}: (TGCC) GTCGTCTTGTTGGTAGTCTTGGT (AAC)

ATGGATGGAGTGAACCAGAGTGACCGTTCACAGTTCCTTCTCCTGGGGAT

GTCAGAGAGTCCTGAGCAGCAGCTGATCCTGTTTTGGATGTTCCTGTCCA

TGTACCTGGTCACGGTGCTGGGAAATGTGCTCATCATCCTGGCCATCAGC

TCTGATTCCCTCCTGCACACCCCCTTGTACTTCTTCCTGGCCAACCTCTC

CTTCACTGACCTCTTCTTTGTCACCAACACAATCCCCAAGATGCTGGTGA

ACGTCCAGTCCCATAACAAAGCCATCTCCTATGCAGGGTGTCTGACACAG

CTCTACTTCCTGGTCTCCTTGGTGTCCCTGGACAACCTCATCCTGGCGGT

GATGGCGTATGATCGCTATGTGGCCAACTGCTGCCCCCTCCACTAGTCCA

CAGCCATGAGCCCTTTGCTCTGTGTCTTGCTCCTTTCCTTGTGTTGGGAA

CTCTCAGTTCTCTATGGCCTCGTCCACACCTTCCTCGTGACCAGCGTGAC

CTTCTGTGGGACTGGACAAATCCACTACTTCTTCTGTGAGATGTAATTGC

TGCTGTGGATGGCATGTTCCAACAGCCATATTAATCACACAGGGGTGATT

GCCACTGGCTGCTTCATCTTCCTCACACCCTTGGGTTTCATGAACATCTC

CTATGTACGTATTGTCAGACCCATCCTATAAATGCCCTCCGTCTCTAAGA

AATACAAAGCCTTCTCTACCTGTGCCTCCCATTTGGGTGTAGTCTCCCTC

TTATATGGGATGCTTCATATGGTATACCTTGAGCCCCTCCATACCTACTC

GATGAAGGACTCAGTAGCCACAGTGATGTATGCTGTGCTGACACCCATGA

TGAATCCGTTCATCTACAGACTGAGGAACAATGACATGCATGGGGCTCTG

GGAAGACTCCTATGAATACGCTTTAAGAGGCTCATA

DE$_{21=001001}$: (TTCA) AGGGGGTG (AGC)

ATGGATGGAGTTAACCAGAGTGACAAGTCAGAGTTCCTTCTCCTGGGGAT

GTCAGAGAGTCCTGAGCAGCAGCGGATCCTGTTTTGGATGTTCCTGTCCA

TGTACCTGGTCACGGTGGTGGGAAATGTGCTCATCATCCTGGCCATCAGC

TCTGATTCCCTCCTGCACACCCCCGTGTACTTCTTCCTGGCCAACCTCTC

CTTCACTGACCTCTTCTTTGTCACCAACACAATCCCCAAGATGCTGGTGA

ACATCCAGTCCCAGAACAAAGCCATCTCCTATGCAGGGTGTCTGACACAG

CTCTACTTCCTGGTCTCCTTGGTGCCCCTGGACAACCTCATCCTGGCAGT

GATGGCTTATGAGCGCTATGTGGCCACCTGCTGCCCCCTCCACTAATGCA

CAGCCATGAGCCCTAGGCTCTGTTTCTTCCTCCTATCCTTGTGTTGGGCT

CTGTCAGTTCTCTATGGCCTCCTGCACACCATCCTCTTGACCAGGGTGAC

CTTCTGTGGGACGTGATAAATCCACTACATCTTCTGTGAGATGTACCTAT

TGCTGAGGTTGGCATGTTCCAACAGCCACATTAGTCACACAGAGGTGATT

GCCACGGGCTGCTTCATCTTCCTCAGACCCTTCGGTTTCATGAACATCTC

CTATGTACGTATTGTCAGAGCCATCCTCATAATACCCTCAGTCTCTAAGA

AATACAAAACCTTCTCTACCTGTGCCTCCCATTTGGGTGGGGTCTCCCTC

TTATATGGGAAACTTGGTATGGTCTACCTACAGCCCCTCCATACCTACTC

AATGAAGGACTCAGTAGCCACAGTGATGTATGCTGTGCTGACACCAATGA

TGAAACCTTTCATCTACAGGCTGAGGAACAACGACATGCATGGGGCTCAG

GGAAGAGTCCTAATAAAACGCTTTCAGAGGCTTAAA

DE$_{221=101001}$: (TTAT) AGTTGGGGT (GGC)

ATGGATGGAGTTAACCAGAGTGAATAGTCATAGTTCCTTCTCCTGGGGAT

TTCAGAGAGTCCTGAGCAGCAGCGGATCCTGTTTTGGATGTTCCTGTCCA

TGTACCTGGTCACGGTGGTGGGAAATGTGCTCATCATCCTGGCCATCAGC

TCTGATTCCCGCCTGCACACCCCCGTGTACTTCTTCCTGGCCAACCTCTC

CTTCACTGACCTCTTCTTTGTCACCAACACAATCCCCAAGATGCTGGTGA

ACTTCCAGTCCCAGAACAAAGCCATCTCCTATGCAGGGTGTCTGACACAG

CTCTACTTCCTGGTCTCCTTGGTGGCCCTGGACAACCTCATCCTGGCCGT

GATGGCATATGATCGCTATGTGGCCAGCTGCTGCCCCCTCCACTAATGCA

CAGCCATGAGCCCTATGCTCTGTGTCTTCCTCCTATCCTTGTGTTGGGTG

CTATCTGTGCTCTATGGCCTCCTACTCACCGTCCTCCTGACCAGAGTGAC

CTTCTGTGGGACTGGACAAATCCACTACTTCTTCTGTGAGATGTACCTCA

TGCTGAGGTTGGCATGTTCCAACAACCAAATAATTCACACAGAGTTGATT

GCCACAGGCTGCTTCATCTTCCTCATGCCCTTCGGATTCTTGAGCACATC

CTATGTACGTATTGTCAGACCCATCCTATGAATCCCCTCAGTCTCTAAGA

AATACAAAACCTTCTCTACCTGTGCCTCCCATTTGGGTGGCGTCTCCCTC

TTATATGGGATGCTTATTATGGTGTACCTCAAGCCCCTCCATACCTACTC

TATGAAGGACTCAGTAGCCACAGTGATGTATGCTGTGGTGACACCTATGA

TGAAACCGTTCATCTACAGGCTGAGGAACAATGACATGCATGGGGCTCTG

GGAAGAATCCTATGCAAACCCTTTTAGAGGCAAATA

$DE_{222=101010}$: (TCTA) GTTGTCTTGGTAGTTGGTACGGCGGTTGGTAT (GGG)

ATGGATGGAGTCAACCAGAGTGATAGTTCATAGTTCCTTCTCCTGGGGAT

GTCAGAGAGTCCTGAGCAGCAGCTGATCCTGTTTTGGATGTTCCTGTCCA

TGTACCTGGTCACGGTGCTGGGAAATGTGCTCATCATCCTGGCCATCAGC

TCTGATTCCCTCCTGCACACCCCCTTGTACTTCTTCCTGGCCAACCTCTC

CTTCACTGACCTCTTCTTTGTCACCAACACAATCCCCAAGATGCTGGTGA

ACGTCCAGTCCCAGAACAAAGCCATCTCCTATGCAGGGTGTCTGACACAG

CTCTACTTCCTGGTCTCCTTGGTGTCCCTGGACAACCTCATCCTGGCAGT

GATGGCGTATGATCGCTATGTGGCCATCTGCTGCCCCCTCCACTAGGTCA

CAGCCATGAGCCCTACGCTCTGTGTCTTGCTCCTCTCCTTGTGTTGGGGG

CTTTCTGTGCTCTATGGCCTCGTTCACACCTTCCTCGTGACCAGGGTGAC

CTTCTGTGGGGCATGAGACATCCACTACATCTTCTGTGATATGTAGCTCA

TGCTGAGGTTGGCATGTTCCAACAGCCAAATTATTCACACAGCGCTGATT

GCCACCGGCTGCTTCATCTTCCTCATGCCCTTAGGTTTCATGATCAGCTC

CTATGTACGTATTGTCAGACCCATCCTTCAAATCCCCTCAGTCTCTAAGA

AATACAAAACCTTCTCCACCTGTGCCTCCCATTTGGGTGTAGTCTCCCTC

TTATATGGGAGTCTTCTTATGGTATACCTAGAGCCCCTCCATACCTACTC

ATTGAAGGACTCAGTAGCCACAGTGATGTATGCTGTGCTGACACCAATGA

TGAAACCCTTCATCTACAGGCTGAGGAACAAAGACATGCATGGGGCTCTG

GGAAGATTCCTATACAAACCCTTTAAGAGGCCAATA

The above packets are sent at various times and different networks. The data packet size can be considered as the header and packet number as the trailer of data packet. We consider that two information the total message length and number of packets N (144 and 4 respectively for above example) has been communicated to the receiver through a secure channel. These packets when received on the receiver's end can be decrypted using the keys (nucleotide sequences used in ClustalW, the crypto transform and the relation σ') and following the steps mentioned in the algorithm.

Comments on the security aspects of the proposed algorithm

In the proposed algorithm, the data, ready to be sent, is broken into N arbitrary parts. These parts are to be sent through different communication channels at different times. As a consequence, it would be almost impossible for any attacker to get back all those parts. Even if one gets all the parts again it would be difficult for him/her to get the original information by properly merging the parts because key packets have to be recognized properly. In our algorithm a steganographic method is used, the strength of DNA steganography lies in our conjecture that deciphering on the basis of cryptanalysis techniques like chosen plain text attack, Chosen-cipher text attack or adaptive chosen-cipher text attack etc become fruitless because it is mathematically infeasible to extract the whole information from these parts in reasonable time.

Concluding remarks and future endeavors

The aim of the any cryptosystem is to procure a reasonable amount of security in transmission of a packet data. Although our algorithm is not in the form of conventional cryptosystem due to the fact that in our system there is a provision of key packet sending but definitely our system deserves the attention of the crypto-community in terms of the security of transmission of data. In spite of our best efforts we have not been successful in deciphering any appreciable information encoded in the DNA sequences for the last decades. So we are hopeful that we will be able to settle the conjecture as posed in the section 3 that it would be computationally infeasible by any present cryptanalytic techniques to decipher the encoded information.

References

1. Vasudevan RA, Abraham A, Sanyal S (2005) A Novel Scheme for Secured Data Transfer over Computer Networks. Journal of Universal Computer Science 11(1): 104-121.

2. Ashish G, Thomas L, John R (2000) DNA-Based Cryptography. Aspects of Molecular Computing. Lecture Notes in Computer Science 29502: 167-188.

3. Vasudevan RA, Abraham A, Sanyal S, Agrawal DP (2004) Jigsaw-based secure data transfer over computer networks. International Conference on 2004, USA. Information Technology: Codin Pp 2-6.

4. Arunava G, Pabitra PC, Amita P, Brahmachary RL, Sarif Hassan Sk (2010) L-Systems: A Mathematical Paradigm for Designing Full Length Genes and Genomes. Genome Biol 11(Suppl 1): P15.

5. https://arxiv.org/abs/1405.2684

6. Chi KC, Cheng LM (2004) Hiding data in images by simple LSB substitution. The Journal of pattern recognition society, Pattern Recognition 37(3): 469-474.

7. Min Wu, Bede Liu (2004) Data hiding in binary image for authentication and annotation. IEEE Transactions on Multimedia 6(4): 528-538.

Your next submission with Juniper Publishers
will reach you the below assets

- Quality Editorial service
- Swift Peer Review
- Reprints availability
- E-prints Service
- Manuscript Podcast for convenient understanding
- Global attainment for your research
- Manuscript accessibility in different formats
  (Pdf, E-pub, Full Text, Audio)
- Unceasing customer service

Track the below URL for one-step submission
https://juniperpublishers.com/online-submission.php