DNA STEGANOGRAPHY

Maths Project,

Submitted to-

Prof. Yamuna M

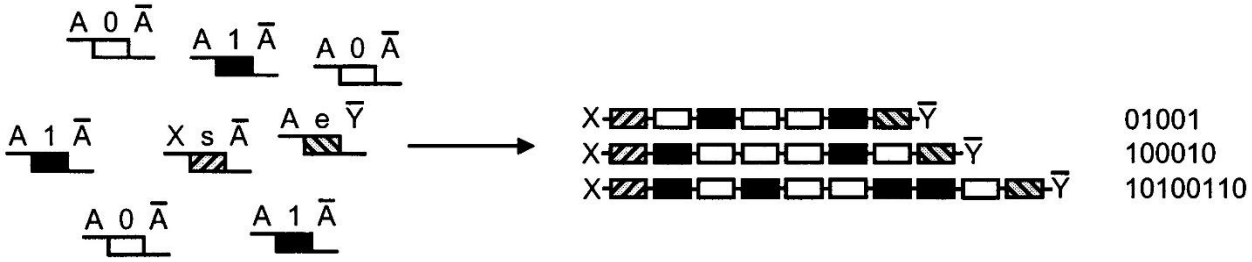
By

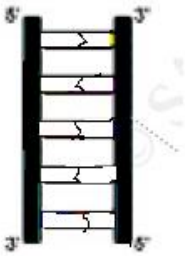
Y.Naga Malleswara Rao(10BCE0114)
Kolli Gopal Rao(10BCE0196)
Saketh P(10BC0495)

SLOT: F2+TF2
Date: 24th April 2013

ABSTRACT

In this paper, we see how messages are encrypted secretly into DNA sequences using few Biotechnological (recombinant DNA) methods which have been developed for a wide range of operations on DNA sequences. It also provides a comprehensive solution for mathematical problems such as Hamiltonian path problem and DES. It is based on plaintext message data encoded in DNA sequences by the use of a publicly known alphabet of short oligonucleotide sequences. By doing this, we can easily transmit messages using DNA without the involvement of a third degree category. Here, we secretly tag the input DNA and get a result of high simplicity. This paper will outline some of the basics of DNA and DNA computing and its use in the areas of steganography, authentication and cryptography.





ACKNOWLEDGEMENT

We would like to thank our professor Prof. Yamuna M for giving us this wonderful opportunity to show our skills and creativity in the field of graph theory and DNA cryptography and helping with all the stumbles and problems we faced in making this.

We have put in a great effort into this paper and implemented it with help of material and qualified friends who have all shown their good nature and made this a success. We also give our word as in putting our complete commitment into doing anything of sorts and keenly look forward to working on the techniques and algorithms in the mathematical field and teaming up with the same group to emphasize and dream of making wonders in this field.

EXTENDED ABSTRACT:

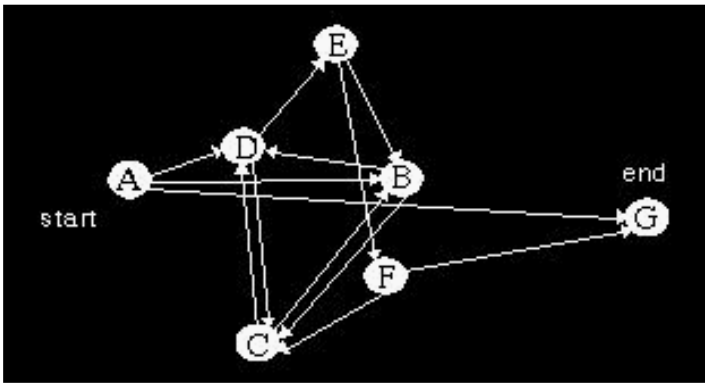
In this project, we try to code DNA sequences into binary numbers 0 and 1 and pass messages to a third person using any highly confidential encryption algorithms. By doing this, we can make sure the matter is kept highly confidential and can be easily transferred to the receiver. Once it is successfully sent we make an attempt in converting the string back to the string using any method suitable in graphical approach. The data to be transferred is as a first plain message is encoded into a DNA sequence using a randomly generated single substitution key.

Basically, steganography is the technique or art or science of hiding information. Cryptography has the mission of manipulating and changing the content, but the other technique steganography works on put that out of the track as in completely covering it. Other than this property, both have the same aim.

For encryption, we can use methods like insertion and complementary pair approach, because they provide more flexible and a secure system to work with. In the insertion method RNS is used as a 2 way modulator and in the later RNS along with complementary pair generation and random number generation is used as 2 extra steps of progress. For concluding the performance is measured and the results are taken. We check the strength of the mean auto-correlation i.e., the binding between the strands in both cases and try to observe it being high in case of an RNS approach.

Learning about the basic DNA structure can be made in any paper posted or published as an article in journals, and cryptography or encryption strategies can also be learnt in the same method aforesaid. Everything started in 1994 when Adleman put his theory on DNA sequencing to frame a Hamiltonian path on a problem called the travelling salesman problem and that's how DNA sequencing developed algorithmically and came into existence. On an algorithmic computational approach, this technique is called as the NP (non-deterministic polynomial time) problem. This method has certain constraints, rules and steps to follow and this done helps a lot to achieve perfect and complete steganography.

Finally to add a proper use and utility to the project, we can see how DNA computing helps in providing help as a beneficiary to the Information security field.



INTRODUCTION:

As the security in today's world is getting breached by third party cases over and over, we as computer science students should invent or at least try to invent new algorithms to keep them intact and messages safe for distant communication. Now, we implement a message transfer using what we call as DNA security. We try to implement it by taking a graph and different assumptions and test cases to protect the data that is send over to another person. Let's have the sender's name to be Alice and receiver termed Bob as the classic example.[1]

i) Alice and Bob exchange the generated key over a secure communication channel. It is expected that only they have the knowledge about the key that we are going to use in the communication.
ii) Alice generates a table and transmits an encrypted message to Bob for his convenience.
iii) If a wrong or dummy data is generated, Bob can easily find it out so as to discard his message.
iv) It may use an open channel to send this message.
v) Bob decodes the right message.

This is what happens in the communication procedure so that the message is transmitted safely from Alice to Bob and an intervening of a third party (Darth) is not possible.

Working:

Let us start by demonstrating all the assumptions and techniques that we have taken in the beginning. This method is to transfer alphabets through graphs in a very secure and a comfortable manner. In the field of DNA, we have the strands-

- A and G complementing each other.(with a triple strand)
- C and T complementing each other. (with a double strand)
- We assign the usage of A and C in a molecule to Alice(the sender) and G and T to Bob(the receiver)
- Alice uses A and C as an encoder and sends the message, Bob gets the message in the format of A and C but his key specifies G and T, which helps him decode the message comprehensively.

Let's take a complete digraph with 27 vertices named with numbers that are only known to Alice and Bob which are generated in a purely random fashion so that the outsider doesn't have any kind of a knowledge about them.

Now, in one such case the random numbers that were generated were-

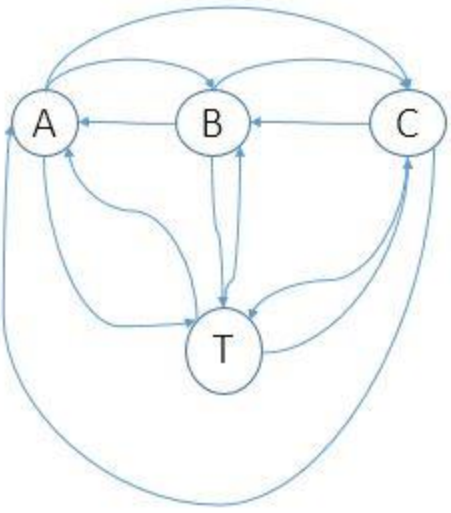
A-11	O-4
B-21	P-16
C-13	Q-20

D-1	R-26
E-9	S-19
F-23	T-14
G-5	U-8
H-22	V-18
I-2	W-15
J-10	X-24
K-17	Y-3
L-6	Z-25
M-12	(Blank space/Null Character)-7

Since, it is a complete digraph all the vertices have an incoming and outgoing edge from every other edge. If we take an edge from M->Y it is termed as 12_A 3_C. This way, we can generalize them to form each outgoing edge while encoding has a suffix A and each incoming edge has suffix C.

So, generally after each message is passed, a table is created with all the numbers (alphabets) in the rows and an A or C as row elements, with the serial number as the column element.

Eg- the look of the complete digraph is shown here.



Let us generalize our communication pattern using an example.

Suppose, Alice wants to send CAT ROCKS to Bob, she sends it in the following fashion.

Encoding phase:

The graph is used to form the order to send. It comes in the following fashion.

$13_A\ 11_C - 11_A\ 14_C\text{-}14_A\ 7_C - 7_A\ 26_C - 26_A\ 4_C - 4_A\ 13_C - 13_A\ 17_C - 17_A\ 19_C$

This shows the traversal in a graph from C->A->T->Blank->R->O->C->K->S and this makes us completely use graph theory.

Now, Bob gets the message like:

$13_G\ 11_T - 11_G\ 14_T\text{-}14_G\ 7_T - 7_G\ 26_T - 26_G\ 4_T - 4_G\ 13_T - 13_G\ 17_T - 17_G\ 19_T$

He converts it to form this format to suffixes of G (equivalent to A in the decoding phase) and T (equivalent to C in the decoding phase), after this Bob follows the same ordering and decodes in the actual order to get the expected output of CAT ROCKS.

	i	ii	iii	iv	v	vi	vii	viii	ix	x	xi	xii	xiii	xiv	xv	xvi
1																
2																
3																
4										C	A					
5																
6																
7						C	A									
8																
9																
10																
11		C	A													
12																
13	A											C	A			
14				C	A											
15																
16																
17														C	A	
18																
19																C
20																
21																
22																
23																
24																
25																
26								C	A							
27																

(**Encoding Table)** for the message CAT ROCKS

	i	ii	iii	iv	v	vi	vii	viii	ix	x	xi	xii	xiii	xiv	xv	xvi
1																
2																
3																
4										T	G					
5																
6																
7						T	G									
8																
9																
10																
11		T	G													
12																
13	G											T	G			
14				T	G											
15																
16																
17														T	G	
18																
19																T
20																
21																
22																
23																
24																
25																
26								T	G							
27																

(DECODING TABLE) for CAT ROCKS

Similarly, we can generalize it for other data that can be sent using graphs. As in all alphabets can be sent without the knowledge of third party users and can be applicable likewise.

Limitations:

i) Since we have to create numbers for alphabets each and every time, it becomes a burden for the sender (Alice) and the receiver (Bob) to use a function. One advantage of this method is that we can prevent the third party intervening in this.

ii) It is time taking to send a whole table of data. So, we can just send the data line but it may not be safe to do this at regular intervals.

iii) While doing it for a huge data word, even though it is very safe, it may become very redundant. Thus, we can try to develop new algorithms that easily beat this in speed and performance.

REFERENCES:

[1] Private and Public key DNA steganography, (C. Richter, A. Leier, W. Banzhaf, and H. Rauhe, Cho, Dong Yeon)

[2] Innovative approach to improve hybrid cryptograph by using DNA steganography(Mohammad Reza Najaf Torkaman1, Nazanin Sadat Kazazi1, Azizallah Rouddini2 1 Faculy of Computer Science and Information System, UNIVERSITI TEKNOLOGI MALAYSIA (UTM), Kuala Lumpur, Malaysia 2Faculty of Management and Human Resources Development, UNIVERSITI TEKNOLOGI MALAYSIA (UTM), Kuala Lumpur, Malaysia))

[3] DNA and DNA Computing in Security Practices – Is the Future in Our Genes?(Global Certification paper)

[4] Cryptography with DNA binary strands (Andre´ Leier, Christoph Richter, Wolfgang Banzhaf, Hilmar Rauhe) University of Dortmund, Department of Computer Science, Chair of Systems Analysis, 44221 Dortmund, Germany
Received 21 January 2000; received in revised form 10 April 2000; accepted 14 April 2000

[5] DNA-based Cryptography (Ashish Gehani, Thomas H. LaBean and John H. Reif
5th Annual DIMACS Meeting on DNA Based Computers
(DNA 5), MIT, Cambridge, MA, June 1999.

[6] DNA Computing and Its Application to Information Security Field (Guangzhao Cui, Cuiling Li, Haobin Li, Xiaoguang Li) Henan Key Lab of Information-based Electrical Appliances, Zhengzhou 450002.