Implementation of Elliptic Curve Cryptography in DNA computing

[1]Sourav Sinha, [2]Shubhi Gupta
[1]Student: Department of Computer Science, [2]Assistant Professor
Amity University (dit school of Engineering)
Greater Noida, India

Abstract— DNA computing is the recent and powerful aspect of computer science Iin near future DNA computing is going to replace today's silicon-based computing. In this paper, we are going to propose a method to implement Elliptic Curve Cryptography in DNA computing.

Keywords—DNA computing; Elliptic Curve Cryptography

1. INTRODUCTION (*HEADING 1*)

The hardware limitation of today's computer is a barrier in a development in technology. DNA computers, also known as molecular computer, have proven beneficial in such cases. Recent developments have seen massive progress in technologies that enables a DNA computer to solve Hamiltonian Path Problem [1]. Cryptography and network security is the most important section in development. Data Encryption Standard(DES) can also be broken in a DNA computer, due to its ability to process parallel[2].

The main aim of this work is to provide a good security in DNA computing to take this emerging technology to next level. The main feature of DNA computers are that it has four bits or nucleotide base namely Adenine (A), Thymine (T), Cytosine (C), Guanine (G). These nucleotide base are the main reason that the standard algorithms are not applicable on this type of computer.

The benefit of DNA computing is that

1. There is large degree of parallelism which helps increasing computing speed.
2. No need of continuous power supply for operation.
3. It can store large amount of data in very less space.

2. WORKING OF DNA COMPUTER

2.1 Overview of DNA

DNA or deoxyribonucleic acid is the basic structure to support life. It stores instruction to build cells. They are blue print or basic code of any living organism.

Genes are the DNA sequence that contains genetic information, other sequence are for structural and regulation purpose. The double helix structure of DNA is maintained by the Hydrogen bonds that attached to nucleotide bases. Hydrogen bonds attach Adenine (A) with Thymine (T) Cytosine (C) with Guanine (G). The bases are further attached to sugar-phosphate to make strands of DNA.

2.2 DNA computer

The DNA computer is different from Modern day's classic computers. The DNA computer is nothing just a test tube containing a DNA and solvents for better mobility. The operations are done by chemical process and protein synthesis. DNA does not have any operational capacities, but it can be used as a hard drive to store and transfer data.

3. DNA-BASED ELLIPTIC CURVE ALGORITHM

The Elliptic curve cryptography makes use of an Elliptic Curve to get the value of Its variable coefficients.

To understand the Elliptic curve we need to understand following:

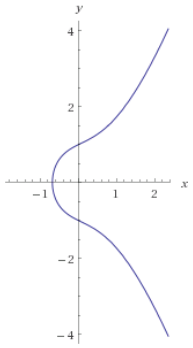
3.1 Equations and Calculations

The equation of a quadratic curve is

$$Y^2=X^2+aX+b. \quad \text{……. (1)}$$

The equation (1) contains variable X & Y and coefficients a and b.

The plot for equation (1) for a=1 and b=1 is



The equation (1) is modified with the mod of P $E_P(a,b) \quad Y^2 = (X^3+aX+b) \bmod P$ ……(2)

Where P is a large prime number over which the prime curve is defined. N_a and N_b are two random integers. G is the base point of $E_P(a,b)$.

P_a is the Public key of user A and P_b is the Public key of user B. The P_a is calculated by N_a*G & P_b by N_b*G.

P_a and P_b are broadcast over and used for Encryption purpose.

The Decryption key K_a and K_b are calculated by

$K_a = N_a* P_b$ & $K_b = N_b* P_a$

3.2 Key Sharing(Public Key)

User A

- Choose a large integer N_a such that $1< N_a<N$.
- Compute $P_a = N_a*G$.
- Broadcast P_a to user B.
- Value of N_a is kept Private to A
- Receive the value of P_b from B
- Compute the common Key by $K_a = N_a* P_b$

User B

- Choose a large integer N_a such that $1< N_b<N$.
- Compute $P_b = N_b*G$.
- Broadcast P_b to user A.
- Value of N_b is kept Private to B.
- Receive the value of P_a from A.
- Compute the common Key by $K_b = N_b* P_a$.

3.3 Algorithm for DNA Computing

Step 1– Take a large prime integer P.

Step 2 – Consider an equation –

$$y^2 = (x^3 + ax + b) \bmod p$$

Step 3 – Take all values between 0 to P-1 and calculate (X,Y) to satisfy the equation.

Step 4 – Take a supposed value G from the point list from last step.

Step 5 – Calculate iG such that (i be a least integer) X coordinate of this point, iG is same as the point G and value of y coordinate is prime number minus the value of Y coordinate of G. the Y coordinate is called Order of G.

Step 6 – Compute Sender and receiver Private and Public Keys K_a and K_b & P_a and P_b respectively.

Step 7 – Encryption:

A(sender) will encrypt message using B(receiver) 's public key i.e. P_b .

1. Let plain text P_m belongs to the point set.
2. Let O be a random integer between 1 to n.

Compute cipher – $(OG, P_m + OP_B)$.

Step 8 – Convert the $(OG, P_m + OP_B)$ into Binary bits.

Step 9 – Convert the Binary to its DNA codons as per Table 1.

Step 10 – Code the DNA codons into DNA.

Step 11 – Decryption:

1. Convert DNA codons to Binary.
2. Convert Binary into Cipher
3. Compute OGn_B.
4. Compute $P_m + OP_B - OGn_B$ to get P_m.

TABLE I. DNA CODONS FORBINARY VALUES

Binary Data	DNA
00	AA
01	T
10	C
11	GG
0	A
1	G

4.CONCLUSION

By using this approach of encoding message into DNA, we can easily work on a DNA computer. The main difference to the modern and DNA computer is that DNA computer have only 4 codons not bytes. It makes a DNA computer unique.

The best thing is that we can use this method to encode any message into a normal DNA and then we can transmit it. Only those who have knowledge can take the message out and decipher the message.

5. REFERENCES

[1] S Mona Sabry, Mohamed Hashem, Taymoor Nazmy, "A DNA and Amino Acids-Based Implementation of Playfair Cipher", International Journal of Computer Science and Information Security, Vol. 8, No. 3, 2010.

[2] Dan Boneh, Cristopher Dunworth, and Richard Lipton. "Breaking DES Using a Molecular Computer". Technical Report CS-TR-489-95, Department of Computer Science, Princeton University, USA, 1995.

[3] William Stallings. "Cryptography and Network Security", Third Edition, Prentice Hall International, 2003.