ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 3, Issue 1 (Jan. - Mar. 2015)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2015)

Encryption using Double Triangulation and Two Point Crossover

[I]Sainik Kumar Mahata, [II]Monalisa Dey, [III]Subhranil Som
[I,II]Asst. Professor, Dept. of CSE, JIS College of Engineering, West Bengal, India
[III]Head, Dept. of Computer Application, Greater Kolkata College of Engineering and Management, West Bengal, India

Abstract
This paper presents a novel approach of cryptography by implementing two approaches, namely, Double Triangulation, which is a bit-wise XOR operation scheme and Two point Crossover, which is a genetic algorithm scheme. This method also incorporates the use of a key to implement the encryption and decryption methods.

Keywords
Crossover, Cryptography, Genetic Algorithm, Triangulation, XOR

I. Introduction

In the recent times, when the world is getting more and more globally connected, the need to secure data transfer is becoming extremely important [1]. Thus, it is essential to come with a way of maintaining the confidentiality of data while protecting its integrity. The science of cryptography is often used for such purpose [2]. It is a technique of using mathematical techniques and algorithms to jumble up the contents of the data to make it non-comprehensible to unauthorized entities. The two key concepts of cryptography are encryption and decryption. Encryption is the method of jumbling the data, whereas, decryption is a way to un-jumble it. In most cases, a value known as the "key" is used as a way of encoding data more efficiently [3]. As the technology is advancing, the algorithms of cryptography are also becoming complex enough o fool the intruders [4].

In this paper a technique known as genetic algorithm is used to make the art of cryptography more productive and difficult to crack. Genetic algorithms involve explorative methods based on the idea of natural selection and a fitness quotient. It basically follows the concept "Survival of the Fittest" [4, 5, 6].

The following sections propose an algorithm for efficient encoding and decoding of data using two approaches, viz. double triangulation and two point crossover.

II. Algorithm

A. Double Triangulation

We take a string of binary numbers, which is obtained from a word or a sentence and break it down into two equal halves. On each of the halves, we apply the following steps to obtain the intermediate cipher text.

- The data string initialized is taken as it is.
- Bit wise XOR operation is performed of all the bits; however, the MSB is not kept constant .This step is considered as the 1st iteration.
- The iteration process is continued until the data string is reduced to a single bit.
- The MSB's from the data string obtained from each of the iterations and are then joined together and taken as the new output.
- If we take the new output as the data string, and perform the above mentioned steps, i.e, step1-step4, we get the original output.

For example:

We take the initial string as 10110011, the iterations will be as follows.

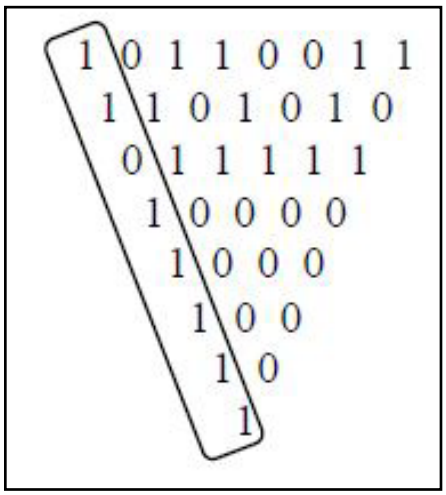

Fig. 1: Steps of Triangulation

So, the cipher text is 11011111

B. Two Point Crossover

Initially we select two parents. After the selection is done, two points on the parents are selected. The crossover proceeds in the way shown in Figure 2.

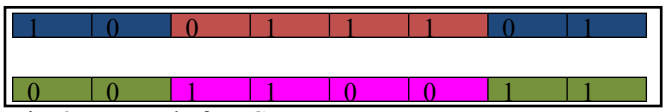

Fig. 2: Parents before Crossover

We select two points, namely 2 and 7, on both the parents. The data before and after the selected points are shaded in blue and green. The data excluding the previously mentioned data are shaded in orange and violet.

After the crossover takes place, the children will look like that shown in Figure3.

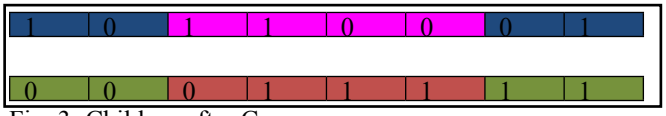

Fig. 3: Children after Crossover

To summarize the algorithm, parts from both the parents, excluding the points selected, are exchanged to create the new children, as shown in Fig.3.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

C. Encryption Algorithm

- We take the plaintext.
- We select the key K between 1 to (n*8)/4, where n is the number of alphabets in the plaintext.
- We find the ASCII equivalent of each alphabet in the plaintext.
- We obtain the corresponding binary equivalents of the ASCII characters.
- We concatenate the multiple binary strings into a single binary string.
- We divide the string into two equal halves.
- On each halve we apply triangulation algorithm.
- The results after the triangulation iterations are named as I1 and I2.
- On concatenating I1 and I2, we obtain IC, i.e, the intermediate cipher.
- The two sub binary strings I1 and I2 act as the parents for the crossover.
- Two points are selected on the parents, first being the key K and the second point is obtained using the formula (m-K)+1, where m is the number of elements in the parents.
- After the crossover is done, we concatenate the elements of the children.
- We break the result into n equal parts, where n is the number of alphabets in the plaintext.
- We find the decimal equivalents of the binary strings.
- We find the ASCII equivalents, to obtain the final ciphertext.

D. Decryption Algorithm

- We take the ciphertext.
- We have the key K.
- We segregate the characters and find the ASCII equivalent of those.
- We find out the corresponding binary equivalent of the ASCII characters obtained
- We concatenate the binary strings obtained into a single string.
- We divide the string obtained into two equal halves that will act as the two parents for the crossover process
- Two points are selected on the parents, first being the key K and the second point is obtained using the formula (m-K)+1, where m is the number of elements in the parents.
- We name the children as I1 and I2.
- We apply triangulation on both I1 and I2
- We concatenate the results of the double triangulation
- We find the Decimal equivalents of the binary characters
- We find the ASCII equivalents, to get the actual Plaintext.

III. Methodology

Let us consider that we want to transmit the word ABC.

So, plain text: ABC

Let us select a key 'K', between 1 to (n*8)/4, where n is the number of alphabets in the plain text.

In this example, we choose K as 3, which are between 1 and 6 ((3*8)/4=6).

A. Encryption

We find out the ASCII equivalent of each alphabet, i.e, A=65, B=66 and C=67.

Next, we find out the binary equivalents of the obtained ASCII characters, i.e,
65= 01000001
66= 01000010
67= 01000011

Next, we concatenate the obtained binary numbers to form a single binary string, i.e, 010000010100001001000011.

Next, we divide the obtained binary string into two equal halves, namely A and B, i.e, A= 010000010100 and B= 001001000011.

Now, we apply Triangulation on both A and B.

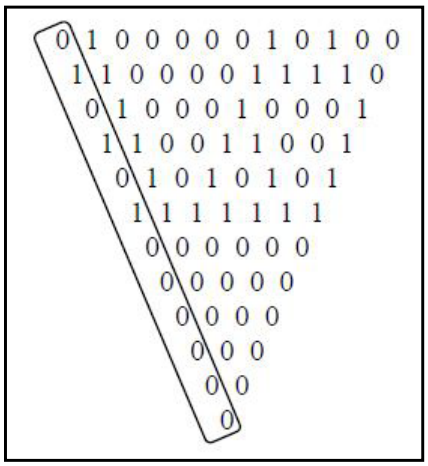

Fig. 4: Triangulation steps on sub binary string A

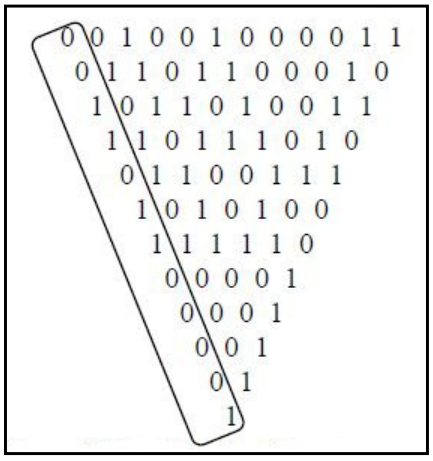

Fig. 5: Triangulation steps on sub binary string B

Applying Triangulation on A, we get 010101000000, which we name as I1 and applying triangulation on B, we get, 001101100001, which we name as I2.

Concatenating I1 and I2, we get the intermediate cipher text IC, i.e, 010101000000001101100001.

Now, taking I1 and I2 as the parents, we apply Two Point Crossover to obtain the Children.

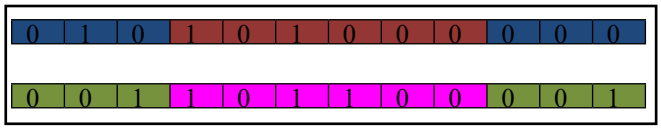

Fig. 6: Parents before Crossover

The points that are selected on the parents are 3 and 10, i.e, the first point is the Key K and the next point is (m-K) +1, where m= no. of elements in the parent. Here, m= 12, so the second point comes as (12-3) +1=10.

The children after the crossover are

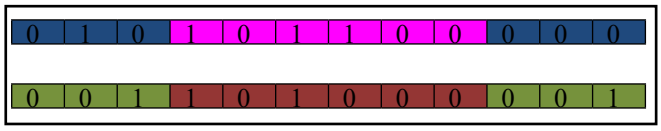

Fig. 7: Children after Crossover

The data in the children are concatenated to get 010101100000001101000001.

Now, we find the corresponding decimal equivalents of the data, i.e,
01010110= 86
00000011= 3
01000001= 65

Now, we find the corresponding ASCII equivalents to get the final Cipher Text, i.e, VETXA

B. Decryption

We have the Cipher Text as VETXA
We have the key K as 3.
Now, we segregate the characters and find the ASCII equivalent of those, i.e,
V= 86
ETX= 3
A= 65

Now, we find out the corresponding binary equivalent of the ASCII characters obtained, i.e,
86= 01010110
3= 00000011
65= 01000001

Now, we concatenate the binary strings obtained into a single string, i.e, 010101100000001101000001

We divide the string obtained into two equal halves that will act as the two parents for the crossover process.

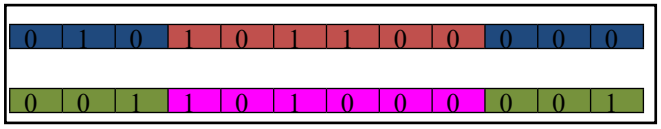

Fig. 8: Parents before Crossover

As we have the key as 3, we get the two points as 3 and 10. The first point is the key itself and the second point is obtained using the formula (m-K) +1.
After the crossover takes place, we get the children as shown in Figure 9.

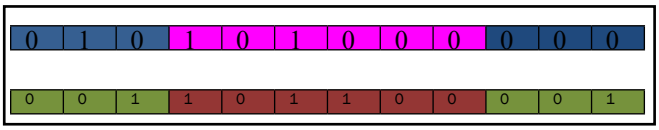

Fig. 9: Children after Crossover

We name the children as I1 and I2, i.e,
I1= 010101000000
I2= 001101100001

We apply triangulation on both I1 and I2, shown in Figure 10 and Figure 11.

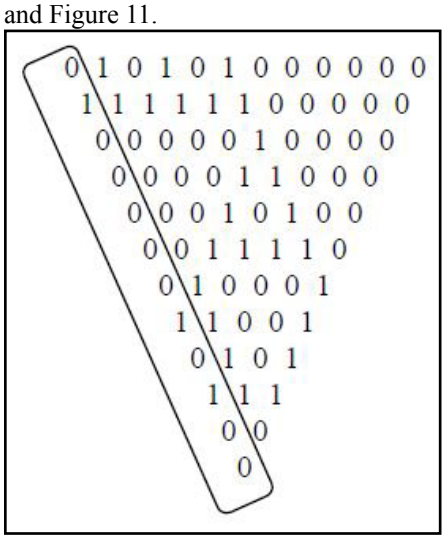

Fig. 10: Triangulation steps on I1.

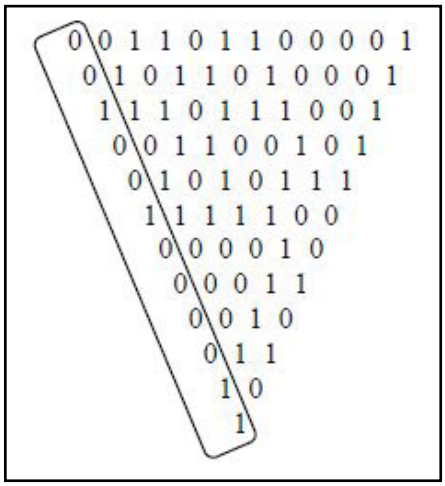

Fig. 11: Triangulation steps on I2

We concatenate the results of the double triangulation to get 010000010100001001000011

We find the Decimal equivalents of the binary characters, i.e,
01000001= 65
01000010= 66
01000011=67
We find the ASCII equivalents, i.e,
65= A
66= B

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

67= C

So, the Plain text is ABC.

IV. Conclusions

The following algorithm is applicable to all kinds of data because it deals with binary data, and every data, be it sound, video, files etc. are all stored as binary in a computer. Moreover this approach is not susceptible to brute force attack as well as man in the middle attack. The proposed algorithm is also easy to implement as well as time taken to execute this is also less. Lastly the key generation process and the intermediate cipher generation algorithm render more security during data transmission.

References

[1] *S. Som, M. Banerjee, "Cryptographic Technique using Substitution through Circular Path Followed by Genetic Function", International Journal of Computer Applications, Special Issue, pp. 1-5, 2012*

[2] *M. Dey et. al., "An Improved Approach of Cryptography using Triangulation and MSB Iteration Technique", International Journal of Computer Applications, Special Issue, pp. 16-18, 2012.*

[3] *S. Mahata et. al., "A Novel Approach to Cryptography using Modified Substitution Cipher and Hybrid Crossover Technique", International Journal of Computer Applications, Special Issue, pp. 33-37, 2013.*

[4] *M. Mitchell, "An Introduction to Genetic Algorithms," The MIT Press, Cambridge, USA, 1999.*

[5] *S., N. Sivanandan, S. N. Deepa, "Introduction to Genetic Algorithm",Springer Verlag Berlin Heidelberg, 2008.*

[6] *A.Tragha , F.Omary, A.Mouloudi ," ICIGA: Improved Cryptography Inspired by Genetic Algorithms", International Conference on Hybrid Information Technology ,IEEE, 335-341,2006.*

Author/s Profile

Sainik Kr. Mahata has completed his B.E and M.Tech in the field of Computer Science and Engineering and is currently employed as an Assistant Professor, in Computer Science and Engineering Department of JIS College of Engineering. His research interests are Cryptography, Network Security and Natural Language Processing.

Monalisa Dey, has completed her M.Tech from National Institute of Technology, Durgapur in the field of Computer Science and Engineering. She is currently employed in Computer Science and Engineering Department of JIS College of Engineering. Her research interests are Cryptography, Network Security and Natural Language Processing.

Subhranil Som has completed his Ph.D in Computer Science and Engineering. He is empanelled as a Ph.D supervisor in West Bengal University of Technology. He has several research publications as well as has authored many books. He is currently serving as Head of Department of Computer Applications Department of Greater Kolkata College of Engineering and Management, West Bengal.