International Research Journal of Computer Science (IRJCS)
Issue 7, Volume 2 (July 2015)
ISSN: 2393-9842
www.irjcs.com

# Implementation of Cryptographic Protocol applying Genetic Algorithm on Staircase Substitution Technique and Randomness Testing

Ranajay Kumar Senapati
*Dept. of Computer Sc. & Engineering,*
*JIS College of Engineering,*
*Kalyani, West Bengal, India.*

Sainik Kumar Mahata
*Dept. of Computer Sc. &Engineering*
*JIS College of Engineering,*
*Kalyani, West Bengal, India.*

Monalisa Dey
*Dept. of Computer Sc.& Engineering,*
*JIS College of Engineering,*
*Kalyani, West Bengal, India.*

***Abstract— In these modern globally connected world, with the internet growing continually, the exchange of information over the web has made the data quite vulnerable. The need for data security is thus increasing exponentially day by day. Cryptography is the science of devising methods that allow for information to be sent in a secure form in such a way that the only person able to retrieve this information is the intended recipient. Encryption is based on algorithms that scramble information into unreadable or non-discernable form. Decryption is the process of restoring the scrambled information to its original form. In this project we shall use Genetic Algorithm (GA) for information security. GA is adaptive heuristic adaptive algorithms based on the evolutionary ideas of natural selection and genetics. We use the concept of Crossover. For network security, we shall implement Stair Case Algorithm for generating intermediate cipher. In this paper; a novel approach for information security is introduced. The encryption technique is achieved using two techniques, firstly using Staircase substitution and lastly using Cut and Splice crossover. Moreover the entire process is done on alphabetical data thus increasing the scope of its implementation.***

***Keywords— Cryptography, Genetic Algorithm, Crossover technique, Staircase substitution method, Cut and Splice method.***

## I. INTRODUCTION

**Cryptography** is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analysing protocols that block adversaries; various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography. Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means.

In our proposed work we use Staircase substitution combined with Genetic Algorithm where we use Cut and Splice crossover technique on plain text for encryption to get cipher text. While decryption we do the opposite of what we have done during encryption.

## II. STAIRCASE SUBSTITUTION

In this proposed model the ASCII values of all the characters in the input stream (plain text) is kept on the points of the staircase as shown in the diagram below :

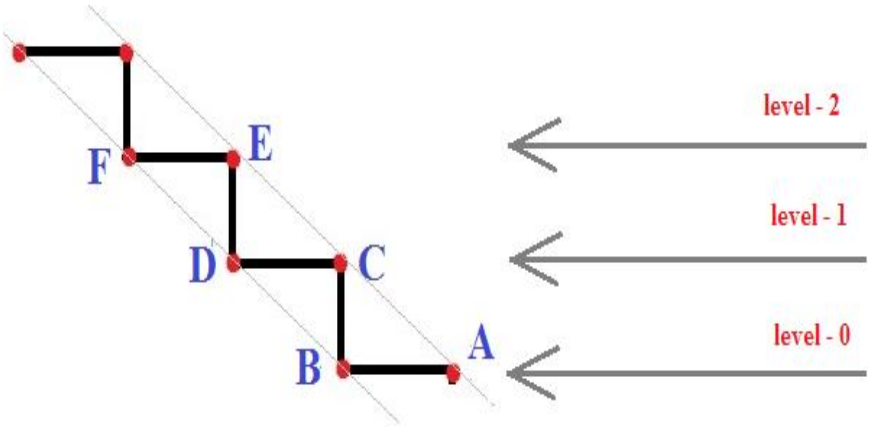



Figure 1: Staircase Model

As we can see that each stair has two points for the characters to be placed. We now choose a non-prime random number which will be our primary key in this algorithm, and modulate it by 256. The number, thus obtained is our secondary key and is added to the ASCII value of the characters of the plain text in accordance with the steps below:

**Level 0:** This is the ground level or $0^{th}$ level of the staircase. The ASCII values of first two characters of the plain text are placed on the $0^{th}$ level of the staircase. In this level the character's ASCII value will only be added with the secondary key obtained.

**Level 1:** This is the $1^{st}$ level of the staircase. The ASCII value of the characters will be added with the secondary keyand the number of points that we have come across along with the ASCII value of the character which lies in the same straight line with it that has already been traversed.

For example, for substitution of the letter C as given in the figure ASCII value of C will be added to the ASCII value of A and for substitution of the letter D, ASCII value of D will be added to the ASCII value of B. Here the modulated random number is incremented by 2 for each subsequent character. And the process goes on in the similar manner for the next higher levels with the modulated random number being incremented by (n+1) during **encryption** where n=level.

During **decryption**, we do the exact opposite to what we do during encryption. We subtract the ASCII values to the secondary key in ground level and in higher levels we subtract from previous level's ASCII of the character which lies along the straight line and number of steps from ground level.

## III. GENETIC ALGORITHM

Genetic algorithm is basically a computer algorithm that simulates evolution process. It adapts natural selection processof living thing in preserving their lives by creating better offspring than its parents. Two concepts that have to be introduced in discussing genetic algorithm are chromosome and reproduction. Chromosome is a macromolecule structure that contains DNA which represents genetic information. Its role is to pass genetic characteristics from parents to its offspring during reproduction process. Reproduction is the phase of creating offspring from two individual parents, during reproduction the chromosome from both parents will be crossed to create new offspring. Computer genetic algorithm adapted these two concepts to search the possible better value from certain initial function values.

In genetic algorithm, **crossover** is a genetic operator used to vary the programming of a chromosome from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Cross over is a process of taking more than one parent solutions and producing a child solution from them. There are methods for selection of the chromosomes.

In our approach we have used the technique of Cut and Splice crossover which is documented below.

### *A. Cut and Splice Crossover*

The crossover process uses cut and splice approach, which has these characteristics:

- Each parent string has a separate choice of crossover point.

- Results a change in length of the children string.

### *B. Example of cut and splice crossover:*

**Before crossover**

*Parent 1:*1001010111010111
*Parent 2:*0010101010101100

Let the crossover points for parent1 be 5 and parent2 be 10.

**After crossover**

*Child 1:***10010**101100
*Child 2:*00101010**10111010**111

## IV. ALGORITHM

### A. *Encryption*

***STEP 1:*** Initially plain text is taken, primary key is decided, and secondary key is calculated.
***STEP 2:*** Each character of plain text is converted into ASCII values and **Staircase Substitution** is applied to get intermediate cipher.
***STEP 3:*** Every character of intermediate cipher is converted to 16 bit binary. Now, every character's binary equivalent is merged to get binary data (say binary data1) and thereby dividing into two equal parts to get parent 1 and parent 2.
***STEP 4:*** We divide the obtained binary data1 to parent 1 and parent 2.
***STEP 5:*** We apply **Cut and Splice genetic algorithm** on parent 1 and parent 2 using two crossover points and obtain child 1 and child 2 and combine to binary data (say binary data2).
***STEP 6:*** We divide binary data2 into binary blocks of 16 bit each, wrap it, and convert into equivalent ASCII to symbol which is our cipher text. This cipher text is the encrypted form of plain text and sent to receiver.

### B. *Decryption*

***STEP 1:*** On the receiver end we need to convert the cipher text to its 16 bit binary block and merge all the converted binary block to binary data (say binary data 3).
***STEP 2:*** Binary data3 is divided into two equal parents - parent1 and parent2.Now, with the help of crossover points we do **Cut and Splice genetic algorithm** on parent1 and parent2 to obtain child1 and child2.
***STEP 3:*** Obtained child1 and child2 is merged into binary data (say binary data 4). Then divide binary data4 to 16 bits each to produce 16 bit equivalent binary data block.
***STEP 4:*** In this step we apply **Staircase substitution** to the binary data block obtained from step 3.
***STEP 5:*** Results of step 4 is converted into ASCII values which in turn is converted to plain text. This decrypted plain text is the actual text intended for receiver.

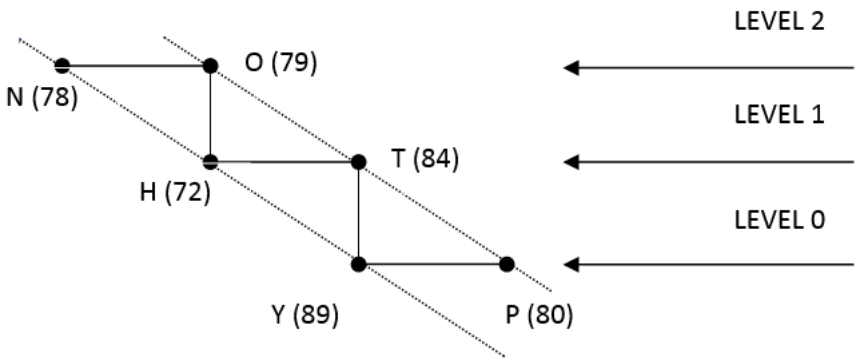
## V. IMPLEMENTATION

### A. *Encryption*

Let us work with,
Plain Text=**PYTHON**
Non-prime random no. as Primary key = **25879658.**
Secondary key = 25879658 % 256 = **106.**

*ASCII values of plain text are,* ***P*** *= 80,* ***Y*** *= 89,* ***T*** *= 84,* ***H*** *= 72,* ***O*** *= 79,* ***N*** *= 78.*

In the first step, we convert each character of plain text to equivalent ASCII values and apply staircase substitution technique to get intermediate cipher.



**Initially, the secondary key is 106.**
P →80 +106→ 186.
Y → 89 +106→195.
**Secondary key incremented by 2, it becomes 106 + 2 = 108.**
T→ 84 + 80 + 108 → 272.
H→ 72 + 89 + 108 → 269.
**Secondary key incremented by 4, it becomes 108 + 4 = 112.**
O→ 79 + 84 + 80 + 112 →355.
N→ 78 + 72 + 89 + 112 → 351.

In the step 2, every character of intermediate cipher is converted to 16 bit binary which is illustrated in Table 1.

TABLE 1. INTERMEDIATE CIPHER TO 16 BIT EQUIVALENT BINARY.

| CHARACTER | CONVERTED INTERMEDIATE CIPHER (USING STAIRCASE SUBSTITUTION) | 16 BIT EQUIVALENT BINARY |
|---|---|---|
| **P** | **186** | **0000000010111010** |
| **Y** | **195** | **0000000011000011** |
| **T** | **272** | **0000000100010000** |
| **H** | **269** | **0000000100001101** |
| **O** | **355** | **0000000101100011** |
| **N** | **351** | **0000000101011111** |

Now, every character's binary equivalent is merged to get binary data1 and thereby dividing into two equal parts to get parent 1 and parent 2.

In step 3, we divide the obtained binary data 1 to parent 1 and parent 2.

Binary data1 = **000000001011101000000000110000110000000100010000000000010000110100000001011000110000000101011111**
Size of binary data1 = No. of character in plain text * 16 bits = 6 * 16 = **96 bits**.
So, size of each parent = Size of binary data1 / 2 = 96 / 2 = **48 bits**.
Parent 1 = **000000001011101000000000110000110000000100010000**
Parent 2 = **000000010000110100000001011000110000000101011111**

In step 4, we apply **Cut and Splice genetic algorithm**,

Crossover point for parent 1 = **3**
Crossover point for parent 2 = **6**
Parent 1 = **000**000001011101000000000110000110000000100010000
Parent 2 = 000000**010000110100000001011000110000000101011111**
Obtained,

Child 1 = **000010000110100000001011000110000000101011111**
Child 2 = 000000000001011101000000000110000110000000100010000

Now, we combine the child 1 and child 2 to obtain binary data2,
Binary data2 = **000010000110100000001000011010000000101011111**000000000001011101000000000110000110000000100010000

In step 5, we divide binary data2 into binary blocks of 16 bit each, wrap it, and convert into equivalent ASCII to symbol which is illustrated in Table 2,

TABLE 2: CONVERSION OF BINARY DATA2 TO SYMBOL

| BLOCK | 16 BIT BINARY | INTEGER EQUIVALENT | ASCII VALUE AFTER WRAPPING UP(WITHIN 256) | SYMBOL (IBM EXTENDED CHARACTER SET) |
|---|---|---|---|---|
| 1 | 0000100001101000 | 2152 | 104 | h |
| 2 | 0000100001101000 | 2840 | 24 | ↑ |
| 3 | 0000101011111000 | 2808 | 248 | ° |
| 4 | 0000000010111010 | 186 | 186 | ‖ |
| 5 | 0000000011000011 | 195 | 195 | ├ |
| 6 | 0000000100010000 | 272 | 16 | ► |

So, the final **cipher text** is **h↑° ‖ ├►**

*B. Decryption*

In first step, the receiver need to convert the cipher text to its 16 bit binary block and merge all the converted binary block to binary data 3.

TABLE 3: CONVERSION OF CIPHER TEXT TO EQUIVALENT 16 BIT BINARY BLOCK

| SYMBOL | ASCII VALUE | CONVERTED INTEGER | 16 BIT BINARY BLOCK |
|---|---|---|---|
| h | 104 | 2152 | 0000100001101000 |
| ↑ | 24 | 2840 | 0000100001101000 |
| ° | 248 | 2808 | 0000101011111000 |
| ‖ | 186 | 186 | 0000000010111010 |
| ├ | 195 | 195 | 0000000011000011 |
| ► | 16 | 272 | 0000000100010000 |

**Binary data3 =** 000010000110100000001000011010000000101011111000000000001011101000000000110000110000000100010000
Size of binary data3 = **96 bits**.
In the step 2, we do **Cut and Splice** on parent1 and parent2 to obtain child1 and child2.

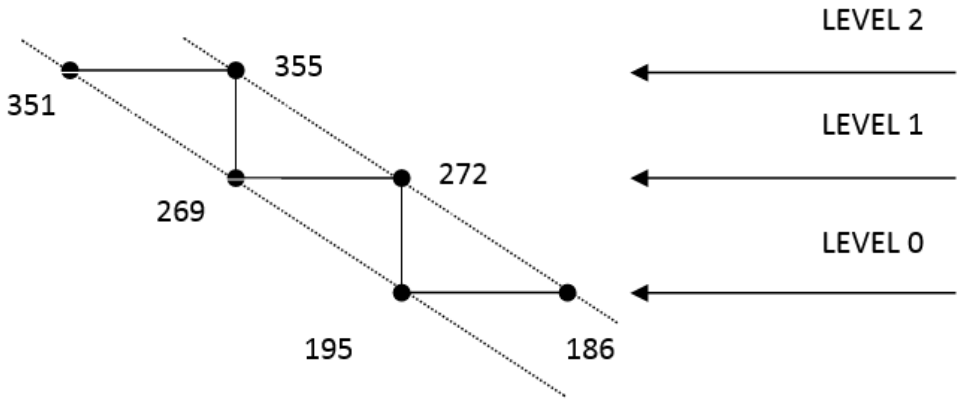
Crossover point for parent 1(p1) = **3**
Crossover point for parent 2(p2) = **6**
So, size of parent1 = p1 + ((size of binary data3 / 2) – p2) = 3 + ((96/2) – 6) = **45 bits.**
Size of parent2 = Size of binary data3 – size of parent1 = 96 – 45 = **51 bits.**
Parent 1 = **000010000110100000001011000110000000101011111**
Parent 2 = **000000000001011101000000000110000110000000100010000**

Therefore, applying Cut and Splice Crossover,
Parent 1 = **000**010000110100000001011000110000000101011111
Parent 2 = 000000**000001011101000000000110000110000000100010000**
Child 1 = **000000001011101000000000110000110000000100010000**
Child 2 = 000000010000110100000001011000110000000101011111
Binary data4 = **000000001011101000000000110000110000000100010000000000010000110100000001011000110000000101011111**

In step 3, we divide binary data4 to 16 bits each to produce 16 bit equivalent binary data block.

In step 4, we use Staircase substitution to these data block obtained from step 3,

Non-prime random no. as Primary key = **25879658.**
Secondary key = 25879658 % 256 = **106.**




**Initially, the secondary key is 106.**
186→186- 106 →80 → P.
195→195- 106 →89→ Y.

**Secondary key incremented by 2, it becomes 106 + 2 = 108.**
272→272- 186 - 2→84→ T.
269→269-195 - 2→72→ H.

**Secondary key incremented by 4, it becomes 108 + 4 = 112.**
355→355- 272 - 4→79→ O.
351→351-269 - 4→78→ N.

Conversion of 16 bit data block to original plain text is illustrated in Table 4.

TABLE 4: CONVERSION OF 16 BIT DATA BLOCK TO PLAIN TEXT.

| BLOCK | 16 BIT BINARY DATA | EQUIVALENT INTEGER | ASCII EQUIVALENT (USING STAIRCASE SUBSTITUTION) | CHARACTER |
|---|---|---|---|---|
| 1 | 0000000010111010 | 186 | 80 | P |
| 2 | 0000000011000011 | 195 | 89 | Y |
| 3 | 0000000100010000 | 272 | 84 | T |
| 4 | 0000000100001101 | 269 | 72 | H |
| 5 | 0000000101100011 | 355 | 79 | O |
| 6 | 0000000101011111 | 351 | 78 | N |

So, the Plain text obtained is **PYTHON**.

## VI. CONCLUSIONS

The algorithm has been implemented and designed on ASCII data. This type of data can be easily converted to binary data. If only binary data is taken, the second approach only can be used to implement security. The key taken are random, thus enhancing security. Last but not the least, the run time of the proposed scheme is very low thus making it more feasible.

## ACKNOWLEDGEMENT

We would like to thank to our Head of Department of Computer Science & Engineering, JIS College of Engineering Mr. Sudarshan Nandy for his immense support and help. We would also like to extend our sincere thanks to our family members, without them we would have been incomplete.

## REFERENCES

[1] URL: https://msdn.microsoft.com/en-us/library/ee276763%28v=bts.10%29.aspx.
[2] 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.
[3] 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.
[4] 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.
[5] M. Mitchell, "An Introduction to Genetic Algorithms," The MIT Press, Cambridge, USA, 1999.
[6] S., N. Sivanandan, S. N. Deepa, "Introduction to Genetic Algorithm",Springer Verlag Berlin Heidelberg, 2008.
[7] A.Tragha , F.Omary, A.Mouloudi ," ICIGA: Improved Cryptography Inspired by Genetic Algorithms" , International Conference on Hybrid Information Technology ,IEEE, 335-341,2006.
[8] S. K Mahata et. al., "Encryption using Double Triangulation and Two Point Crossover", International Journal of Advanced Research in Computer Science and Technology, 2015, pp.37-40.