International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Encryption and Decryption of a Message Involving Genetic Algorithm

Ayush Mittal, Ravindra Kumar Gupta

Abstract: ***The aim of this paper is to establish an algorithm for encryption and decryption of a message based on symmetric key cryptosystem involving Genetic Algorithm. In the proposed algorithm we use substitution algorithm, genetic crossover and mutation technique.***

Key Words: ***Encryption, Decryption, Genetic Algorithm, Crossover, Mutation and substitution.***

I. INTRODUCTION

To provide information security we generally use Cryptography, which is a set of mathematical techniques. The key concepts of cryptography are Encryption and decryption [2]. The privacy of the data is protected by encrypting when data is send to receiver i.e. data is converted to illegible form. This encrypted data can be decrypted by receiver into its original form. To encrypt and decrypt the message an encrypting and decrypting key is required. When encryption and decryption process use the same key then it is called symmetric key or public key cryptography, while process is called asymmetric key or private key cryptography if we use different keys.

Substitution and transposition [7] are two types of cryptographic techniques. In substitution technique each plaintext symbol is replaced by another symbol, while in the plaintext transpose of a symbol is known as transposition technique.

An evolutionary algorithm which is based on the notion of natural selection [6] is known as Genetic algorithm. Since Genetic algorithm does not apply the natural numbers directly, therefore it is secure algorithm. Two basic functions i.e. crossover and mutation [4] are involved in Genetic algorithm. On taking more than one parent chromosomes, the child chromosome is produced in crossover function. Single point crossover, two point crossover, Uniform crossover and three parent crossover [1] are various crossover techniques. There are various types of mutation techniques such as Gaussian mutation flipping of bits, uniform mutation, non-uniform mutation, boundary mutation and inversion mutation.

II. RELATED WORK

Genetic Algorithm based symmetric and asymmetric key cryptosystem for encryption and decryption of data have discussed by Dutta [3], Veetil [8], Nagde [5] and others.

Revised Manuscript Received on December 30, 2019.

* Correspondence Author

Mr. Ayush Mittal, Pursuing Ph. D, Department of Computer Science and Engineering, SRK University, Bhopal, (MP.) India.

Dr. Ravindra kumar Gupta, Associate Professor, Department of Computer Science and Engineering, RKDFIST, Bhopal, (M.P.) India.

Taking into account the obligation and weight of symmetric and asymmetric key cryptosystem for encryption and decryption of data involving Genetic Algorithm, we propose to establish a new encryption and decryption techniques of a message involving Genetic Algorithm (crossover and mutation techniques).

III. METHODOLOGY

The following steps are used in the proposed algorithm:

(i) Matrix Subtraction:

(a) Plain text is converted into the text matrix.

(b) User input (key) is converted into the key matrix.

(c) Subtractive matrix is generated by subtraction of text matrix and key matrix.

(ii) Substitution Algorithm:

To produce the intermediate cipher, linear substitution function is applied on the subtractive matrix. The form of substitution algorithm is $y = C(x) = (ax + b)(\text{mod } n)$, where x is the numerical equivalent of the given plaintext letter. a and b are chosen integers. The decryption form is $C^{-1}(y) = a^{-1}(y - b)(\text{mod } n)$. Here we use n = 31. The following conversion table for the English alphabets is used for substitution.

Table I Conversion Table

alphabet/ symbol	numerical value	alphabet/ symbol	numerical value
#	0	P	16
A	1	Q	17
B	2	R	18
C	3	S	19
D	4	T	20
E	5	U	21
F	6	V	22
G	7	W	23
H	8	X	24
I	9	Y	25
J	10	Z	26
K	11	@	27
L	12	&	28
M	13	%	29
N	14	&	30
O	15		

(iii) Genetic crossover and mutation technique:

To generate the final cipher text, the Generating Algorithm functions (crossover and mutation) are applied on the intermediate cipher. In this paper we use uniform crossover techniques. In uniform crossover technique, we select the two parents for crossover.

It creates two child offspring of n genes selected from both of the parents uniformly. In this paper, we use inversion mutation. In this technique we select a subset of genes; we merely invert the entire string in the subset instead of shuffling the subset.

Here we also use the following EBCDIC Character Table:

Table II

Dec	Code	Dec	Code	Dec	Code
64	Space	212	M	233	Z
193	A	213	N	240	0
194	B	214	O	241	1
195	C	215	P	242	2
196	D	216	Q	243	3
197	E	217	R	244	4
198	F	226	S	245	5
199	G	227	T	246	6
200	H	228	U	247	7
201	I	229	V	248	8
209	J	230	W	249	9
210	K	231	X	123	#
211	L	232	Y	124	@
80	&				

IV. ALGORITHM

4.1 Encryption Algorithm:

(i) Generate a key matrix.
(ii) Convert the chosen plain text into a text matrix.
(iii) By subtracting key matrix from plain text matrix, produce a subtractive matrix.
(iv) To generate an intermediate cipher, apply substitution algorithm on the subtractive matrix.
(v) To generate the cipher text, generic function (uniform crossover, inversion mutation) is applied on the intermediate cipher.
(vi) Key matrix, block size, substitution key, cross over and mutation points are send to the receiver to generate the plaintext back.

4.2 Decryption Algorithm:

(i) To produce the intermediate cipher, apply the reciprocal of cross over and mutation technique.
(ii) To generate the original plaintext, reverse the substitution algorithm and matrix addition operation.

Example:

Let block size n = 4
Choose input key be CRYPTOGRAPHY KEY
The EBCDIC equivalent for the chosen key are:
195 217 232 215 227 214 199 217 193 215 200 232 64 210 197 232
Binary equivalent of the above EBCDIC equivalent are:

11000011 11011001 11101000 11010111
11100011 11010110 11000111 11011001
11000001 11010111 11001000 11101000
01000000 11010010 11000101 11101000

Now apply the right shift operation by 2 bits on the above binary streams, we get as follows:

00110000 00110110 00111010 00110101
00111000 00110101 00110001 00110110
00110000 00110101 00110010 00111010
00010000 00110100 00110001 00111010

Now obtained the decimal equivalent of the above binary sets, the Key Matrix is:

$$K = \begin{bmatrix} 48 & 54 & 58 & 53 \\ 56 & 53 & 49 & 54 \\ 48 & 53 & 50 & 58 \\ 16 & 52 & 49 & 58 \end{bmatrix}$$

A. Encryption:

(i) Consider the plain text is ENCRYPTION BLOCK
(ii) Convert the plain text into block of size n, we get their EBCDIC equivalent text matrix as given below:

$$E = \begin{bmatrix} 197 & 213 & 195 & 217 \\ 232 & 215 & 227 & 201 \\ 214 & 213 & 64 & 194 \\ 211 & 214 & 195 & 210 \end{bmatrix}$$

(iii) Subtract the key matrix from text matrix, we get following subtractive matrix:

$$A = \begin{bmatrix} 149 & 159 & 137 & 164 \\ 176 & 162 & 178 & 147 \\ 166 & 160 & 14 & 136 \\ 195 & 162 & 146 & 152 \end{bmatrix}$$

(iv) To produce intermediate cipher, we apply the substitution algorithm on the above subtractive matrix. Here we use C(x) = (ax + b)(mod 31), where a = 5, b = 2.

C(149) = (5×149 + 2)(mod 31) = 747 (mod 31) = 3 = C (By using Table I)

Similarly we get
VEPNFXXZ@J#PFSR
Finally we get the intermediate cipher, given below
CVEPNFXXZ@J#PFSR

(v) Write equivalent EBCDIC code for the above intermediate cipher, we get:
195 229 197 215 213 198 231 231 233 124 209 123 215 198 226 217

(vi) Convert the above EBCDIC code to their 8 bit binary equivalent as follows:

B_1 = 11000011 B_2 = 11100101 B_3 = 11000101
B_4 = 11010111 B_5 = 11010101 B_6 = 11000110
B_7 = 11100111 B_8 = 11100111 B_9 = 11101001
B_{10} = 01111100 B_{11} = 11010001 B_{12} = 01111011
B_{13} = 11010111 B_{14} = 11000110 B_{15} = 11100010
B_{16} = 11011001

(vii) Now above binary streams is divided into two segments, as follows:
1100001111100101110001011101011111010101110001101110011111100111
1110100101111100110100010111101111010111110001101110001011011001

(viii) Now applying the uniform crossover on the set of 8 bits after each 8 bit, we get
1100001101111100110001010111101111010101110001101110011111011001
1110100111100101110100011101011111010111110001101110001011100111

(ix) For mutation operation here we use inversion mutation technique on each 8 bit group from 3rd bit to 6th bit, therefore we have
1100001101111100111000010101111111101001111000101110011111011001
1101010111100101110010011110101111101011111000101100011011001111
(x) Now divide above binary bits into the set of 8 bits and convert them into their equivalent hexadecimal, which is the final cipher as follows:
C3, 7C, E1, 5F, E9, E2, E7, D9, D5, E5, C9, EB, EB, E2, C6, E7
(xi) Now this cipher text (C37CE15FE9E2E7D9D5E5C9EBEBE2C6E7) is send to the receiver for decryption along with CRYPTOGRAPHY KEY452CM.
Here user input is CRYPTOGRAPHY KEY, block size is 4, the prime numbers which are used in the substitution algorithm are 5 & 2, C is the uniform crossover applied on the set of 8 bits after each 8 bit and M is the inversion mutation on each 8 bit set from 3rd bit to 6th bit.

B. Decryption:

1. Consider the cipher text
C3, 7C, E1, 5F, E9, E2, E7, D9, D5, E5, C9, EB, EB, E2, C6, E7
2. Convert it into 8 bit binary equivalent and divide the binary streams into two segments, we get
1100001101111100111000010101111111101001111000101110011111011001
1101010111100101110010011110101111101011111000101100011011001111
3. Apply inversion mutation operation on each 8 bit group from 3rd bit to 6th bit, we get
1100001101111100110001010111101111010101110001101110011111011001
1110100111100101110100011101011111010111110001101110010111001111
4. Applying the uniform crossover on the set of 8 bits after each 8 bit, we get
1100001111100101110001011101011111010101110001101110011111100111
1110100101111100110100010111101111010111110001101110010111011001
5. Now converting the above into 8 bit group i.e.
B_1 = 11000011 B_2 = 11100101 B_3 = 11000101
B_4 = 11010111 B_5 = 11010101 B_6 = 11000110
B_7 = 11100111 B_8 = 11100111 B_9 = 11101001
B_{10} = 01111100 B_{11} = 11010001 B_{12} = 01111011
B_{13} = 11010111 B_{14} = 11000110 B_{15} = 11100010
B_{16} = 11011001
6. Write their EBCDIC decimal equivalent, we get
195 229 197 215 213 198 231 231 233 124 209 123 215 198 226 217
7. After converting above into equivalent EBCDIC code, we get intermediate cipher as follows:
CVEPNFXXZ@J#PFSR
8. Now find there numeric values using table I, we get
3 22 5 16 14 6 24 24 27 10 0 16 6 19 18
9. After applying the reverse substitution function $C^{-1}(y) = a^{-1}(y - b)(\text{mod } 31)$, we move on the following matrix of 4 × 4 order as follows:

$$A = \begin{bmatrix} 149 & 159 & 137 & 164 \\ 176 & 162 & 178 & 147 \\ 166 & 160 & 14 & 136 \\ 195 & 162 & 146 & 152 \end{bmatrix}$$

10. Add the key matrix K with A, we get

$$E = \begin{bmatrix} 197 & 213 & 195 & 217 \\ 232 & 215 & 227 & 201 \\ 214 & 213 & 64 & 194 \\ 211 & 214 & 195 & 210 \end{bmatrix}$$

11. The EBCDIC character code equivalent of this matrix gives the original plain text i.e.
ENCRYPTION BLOCK

V. RESULT AND DISCUSSION

In the proposed algorithm we use right shift, matrix subtraction, substitution algorithm, modulo operation and genetic operations (uniform crossover, inversion mutation) as key operations. Due to this reason the extraction of original plain text is difficult. Modulo function has the larger order of growth among the key operation. Here the order of growth will be $O(n^2)$, if a data is encrypted by the proposed algorithm. Also, Brute force attack cannot be able to break the cipher because here we use 4×4 matrix as key length. Explanation of result is as follows:

SN	Name of attack	Opportunity of attack	Explanation
1	Known plain text attack	Difficult	Due to the chosen genetic operation and key length
2	Chosen plain text attack	Difficult	Due to the chosen genetic operation and key length
3	Adaptive chosen plain text attack	Difficult	Due to the chosen genetic operation and key length
4	Cipher text attack	Very Difficult	Due to the chosen genetic operation and secret key
5	Chosen cipher text attack	Very Difficult	Due to the chosen genetic operation and secret key
6	Adaptive chosen cipher text attack	Very Difficult	Due to the chosen genetic operation and secret key

VI. CONCLUSION

In this paper the describe algorithm is simple and easy for implementing in cryptographic scheme. For transferring the data there are good security is provided by key generation process and intermediate cipher algorithm. To ensure secrecy in networks, here we use symmetric key substitution algorithm. To provide additional security, we combine symmetric key substitution algorithm with the help of genetic functions.

REFERENCES

1. Bhasin Harsha, Kumar Ramesh, Kathuria Neha: "Cryptography using Cellular Automata". International Journal of Computer Science and Information Technology, Vol. 4(2), 355-357, 2013.
2. Douglas, R. Stinson: "Cryptography – Theory and Practice ", CRC Press, 1995.

3. Dutta Suvajit, Das Tanumay, Jash Sharad, Patra Debasish, Paul Pranam: A Cryptography Algorithm Using the Operations of Genetic Algorithm & Pseudo Random Sequence Generating Functions, International Journal of Advances in Computer Science and Technology, Volume 3, No.5, May 2014, pp. 325-330.
4. Mitchell M.: "An Introduction to Genetic Algorithms," The MIT Press, Cambridge, USA, 1999.
5. Nagde Deepak, Patel Raviraj, Kelde Dharmendra: New Approach for Data Encryption using Two Way Crossover, International Journal of Computer Science and Information Technologies, Vol. 4 (1), 2013, pp. 58 - 60.
6. Sivanandan S. N., Deepa S. N.: "Introduction to Genetic Algorithm", Springer Verlag Berlin Heidelberg, 2008.
7. Stallings William: Cryptography and Network Security Principles and Practices, Prentice Hall, 2005.
8. Veetil Amritha Thekkumbadan: An Encryption Technique Using Genetic Operators, International Journal of Scientific & Technology Research, Vol. 4, Issue 07, July 2015, pp. 202-203.

AUTHORS PROFILE

First Author

Mr.Ayush Mittal did his post-graduation in M.Tech(Master of Technology) in Computer Science and Engineering from Indian Institute of Information Technology and Management(IIITM),Gwalior, MP in 2015. He has also received gold medal in post-graduation degree. Currently, he is pursuing his Phd in Computer Science and Engineering from SRK university, Bhopal, MP. His area of interest includes robotics, cryptography and embedded systems. He has published 2 research papers in International Journals.

Second Author

Dr. Ravindra kumar Gupta received his M.Tech (Master of Technology) degree in Computer Science & Engineering from Sri Satya Sai Institute Of Science & Technology, RGPV Bhopal, In 2010 M.P., Ph.D in Computer Science & Enginnering From Barkatullah University Bhopal India. Presently he is Associate Professor Of Computer Science and Engineering Department in RKDFIST, BHOPAL,M.P. India. He is having 12 Yrs of teaching experience .He has published 54 papers in referred International/National Journal & conference also a Member of Easy Chair Conference System.