 ISBN: 978-81-927230-0-6

Symmetric Key Encryption using Genetic Algorithm

Dr. Poornima G. Naik
Assistant Professor
Department of Computer Studies
Chh Shahu Institute of Business Education and Research,
Kolhapur, India
luckysankalp@yahoo.co.in

Mr. Girish R. Naik
Associate Professor
Production Department
KIT's College of Engineering
Kolhapur,India
girishnaik2025@gmail.com

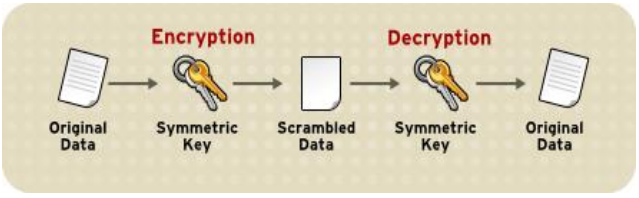
Abstract - Genetic Algorithm (GA) is an invaluable tool for solving optimization problems due to its robustness. It does not break even if the inputs are changed slightly or in the presence of a reasonable noise. GA offers significant benefits over other optimization techniques in searching a large state space or n-dimensional surface [1]. In todays information age information sharing and transfer has increased exponentially. With the ever increasing growth of multimedia applications security has become an important issue in the communication of text and images. Encryption has extensive applications in preserving confidentiality of data in Internet applications [2]. There are various proposed methods for image encryption such as quad tree approach, cellular automata [3, 5]. With the popularization of Internet and exponential increase in e-commerce transactions security has become an inevitable and an integral part of any e-commerce application. Data integrity, confidentiality, authenticity, non-repudiation have gained tremendous importance and have become important components of information security. There are many risks involved in communication of plain text over Internet. Cryptography is a technique of encoding and decoding messages so that they cannot be interpreted by anybody except the sender and the intended recipient. There are wide applications of GA in solving non-linear optimization problems in various domains. But very few papers exist which exploit the randomness in the algorithm for implementation of security. Chaos theory and entropy have large application in secure data communication and the desired disorder is provided by inherent nature of genetic algorithm [6, 8]. Mohammad Sazzadul Hoque et.al have presented an intrusion detection system by applying GA to efficiently detect various types of network intrusions. They have used evolutionary theory to filter the traffic data and thus reduce the complexity [9, 10]. There are several papers related to IDS all of which use GA in deriving classification rules [11, 14]. In this paper we have made an attempt to exploit the randomness involved in crossover and mutation processes for generating a secure one time symmetric key for encryption and decryption of messages. The number of crossover points together with number of mutation points dictate the length of the secret key and hence the strength of the algorithm. The algorithm is further strengthened by making it difficult to break by permuting the symmetric key by a predefined permutation factor agreed upon by both the sender and the intended receiver. The randomness together with permutation makes the algorithm robust and hard to break. Finally, the algorithm is implemented in Java and applied for the encryption and decryption of a text file. The methodology is general and can be applied to any file for secure transmission of data.

I. Index Terms – Genetic Algorithm, Cross-over, Mutation, Cryptography, Encryption, Decryption, Symmetric key.

The paper is organized as follows. The first section gives an introduction to Genetic Algorithm and Cryptography under the heading of Introduction. Section II covers the literature survey and the current scenario of application of soft computing in implementing security. Section III focuses on the proposed method of symmetric key encryption using Genetic Algorithm. Section IV covers implementation of the algorithm in Java. Finally, Section V is devoted for conclusion and scope for future enhancements.

I. INTRODUCTION

Genetic algorithms (GA) are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics. They are based on the principle of Darwinian idea of survival of the fittest and natural genetics. In a symmetric key encryption or secret key encryption only one key is used by both the sender and the intended receiver for both the encryption and decryption of the message. Both the sender and the intended receiver must agree upon the key before any communication begins. Fig. 1. Shows the working of symmetric key encryption. At the sender's end the key is used to encrypt the original message into an encrypted form known as a cipher text. At the receiver's end the same key is used to decrypt the encrypted message and restore a plain text from it.



In practical situations, symmetric key encryption has number of problems. One such problem is that of key agreement and distribution. In this paper, we use Genetic Algorithm for generating a one time symmetric key which is entirely a new approach and is not publicized like RSA and Des algorithms. Hence, even if the key becomes available to an unauthorized user it is difficult to break an algorithm.

The second problem is more serious. Since the same key is used for both encryption and decryption, one key per set of communicating parties is required. This limitation can be overcome by generating a key pair, a public key and a private

key where a public key is freely distributed and the private key is kept confidential known only to the owner of the key pair. Our future work focuses on generating a key pair using GA.

II. CRYPTOGRAPHY

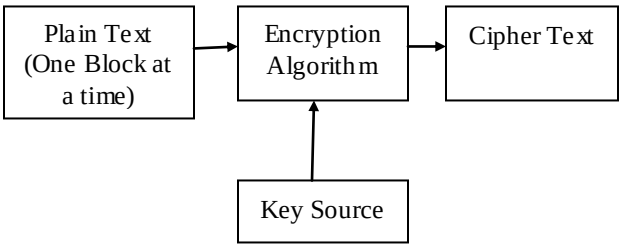
Cryptography plays an important role in network security. Cryptography is the science of writing in secret code. The purpose of cryptography is to protect transmitted information from being read and understood by anyone except the intended recipient. In the ideal sense, unauthorized individuals can never read an enciphered message Cryptographic systems are generally classified among three independent dimensions.

- Types of Operations – All encrypted algorithms are based on two general principles, substitution and transposition. The fundamental requirements are that no information is lost and all operations are reversible.
- Key used – The length of the key determines the strength of the security.

In our current work, GA algorithm transfers 16-byte plain text blocks into 16-byte cipher blocks. Each parent is a 8-byte plain text block On applying various crossover and mutation operations using a randomly generated one time symmetric key the corresponding cipher child blocks are generated. The symmetric key is generated randomly and consists of the following components.

- Randomly generated cross-over points in the range 0-7
- Randomly generated mutation points in the range 0-7.
- Randomly generated permutation factor in the range 1-4.

The strength of the key depends on the number of cross-over points and mutation points. The entire process is depicted in Fig. 2.



Genetic Algorithm

Generally, a Genetic Algorithm consists of three basic operations.

- Selection
- Crossover
- Mutation

The first step consists of searching individuals for reproduction. In our problem, we have selected two vectors of 8 bytes each as parents for reproduction. Since the problem is of encryption, there is no special preference given to any particular selection method. All the vectors are selected sequentially based on their order of appearance in a text file.

Cross-over is the process of taking two parents and producing from them a child. In an optimization problem, crossover operator is applied to the mating pool with the hope that it creates a better offspring. For the problem under consideration, crossover is taken as one of the steps in producing a decrypted vector. We have employed two-point crossover method. In the case of optimization problem, selecting more than two crossover points will result in the disruption of building blocks whereas in the case of encryption larger the disruption better is the algorithm which makes it robust and difficult to break.

After crossover, the vectors are subject to mutation. In optimization problem, mutation prevents the algorithm from being trapped in a local minimum. Mutation plays the role of recovering the lost genetic matter as well as for a randomly distributed genetic information. In encryption problem, mutation is employed for inducing disorder into the vector. It introduces a new genetic structure in the population by randomly modifying some of the building blocks and maintains diversity into the population. We have employed flipping method, in which for a character 1 in mutation chromosome, the corresponding character b in the parent chromosome is flipped from b to (128-b) and corresponding child chromosome is produced. In the following example 1 occurs at two random places of mutation chromosome, the corresponding characters in parent chromosomes are flipped and the child chromosomes are generated.

Parent Chromosome	b0	b1	b2	b3	b4	b5	b6	b7
Mutation Chromosome	1	0	0	0	0	0	0	1
Child Chromosome	128-b0	b1	b2	b3	b4	b5	b6	128-b7

III. LITERATURE SURVEY

In literature to date, many GA based encryption algorithms have been proposed. A. Tragha et.al [1,2] have describe a new symmetric block cipher system namely, ICIGA (Improved Cryptographic Inspired by Genetic Algorithm) which generates a session key in a random process. The block size and key length are variables and can be fixed by the end user in the beginning of the cipher process. ICIGA is an enhancement of the system GIC (Genetic Algorithm inspired Cryptography) [3].

In this paper we have made an attempt, to device a new approach to data encryption based on symmetric key. The randomness involved in crossover and mutation is exploited in generation of a one time symmetric key. A permutation factor is randomly generated and applied to the successive blocks of text to make the algorithm more unpredictable for the intruder. Finally, the algorithm is implemented in Java and applied to a text file.

IV. *PROPOSED* METHOD

The flow chart for encryption process using GA is depicted in Fig 3.1

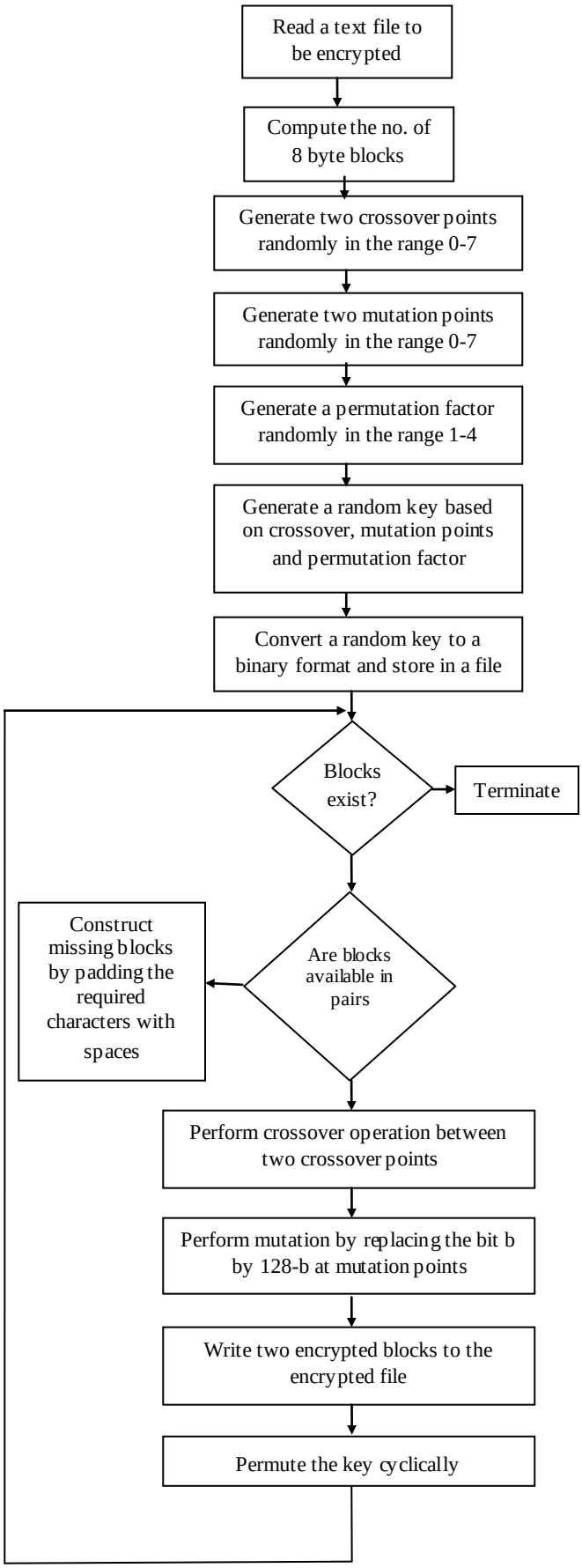


Figure 3.1 Flowchart for Encryption of a text file using GA

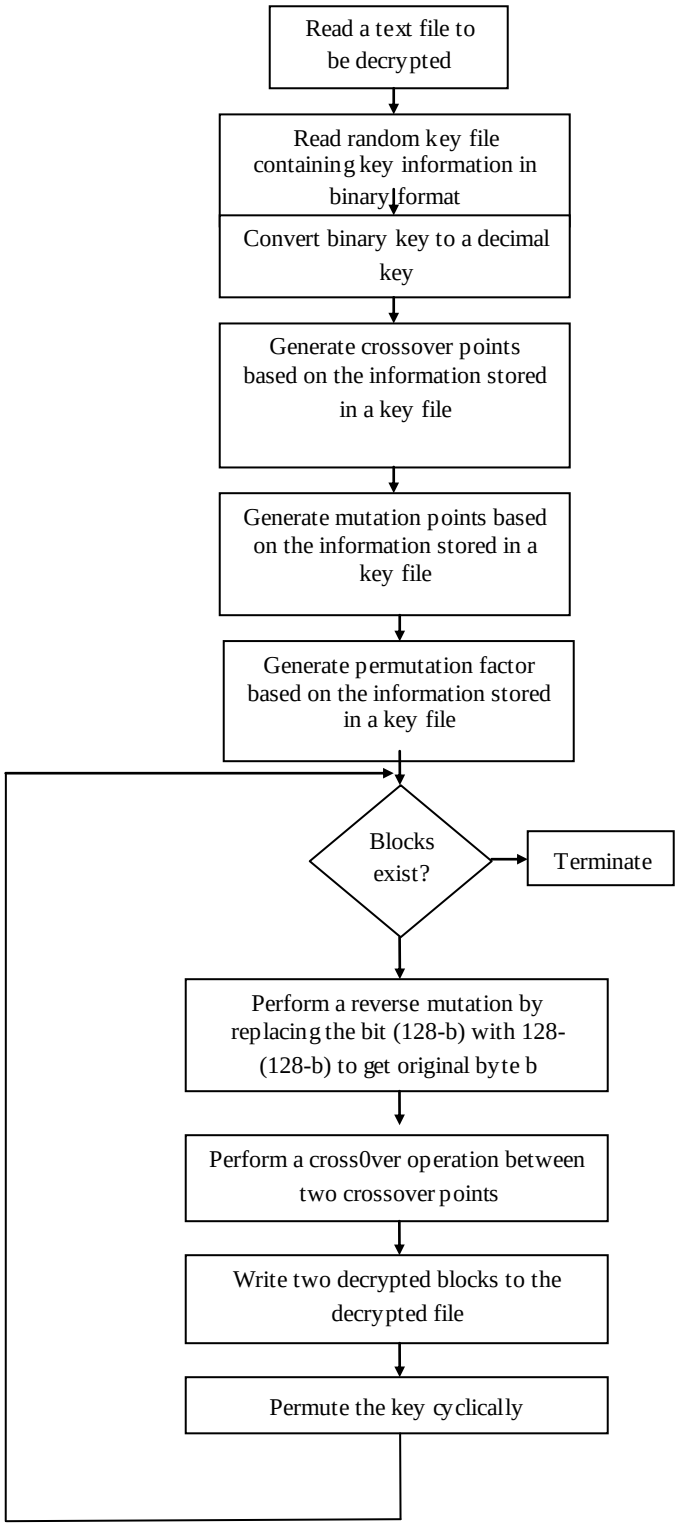


Figure 3.2 Flowchart for Decryption of a text file using GA

Step 1: Extract two 8-byte blocks from the text file to be encrypted. Let the two blocks be represented by

b0	b1	b2	b3	b4	b5	b6	b7

c0	c1	c2	c3	c4	c5	c6	c7

where each bi and ci is a character in a file.

Step 2 : Perform crossover operation.
Generate two random numbers in the range 0-7. Let the two random numbers generated be 2 and 5.

Perform the crossover between two crossover points generated above.

b0	b1	b2	b3	b4	b5	b6	b7

c0	c1	c2	c3	c4	c5	c6	c7

The blocks after performing crossover operation are

b0	b1	c2	c3	c4	b5	b6	b7

c0	c1	b2	b3	b4	c5	c6	c7

Step 3: Perform mutation operation.
Generate two random numbers in the range 0 to 7. Let the two random numbers generated be 1 and 7.

Hence, Mutation Point1 = 1
Mutation Point2 = 7

Perform mutation operation on two blocks obtained in Step 1.

b0	128-b1	c2	c3	c4	b5	b6	128-b7

c0	128-c1	b2	b3	b4	c5	c6	128-c7

Step 4: Generate the permutation factor randomly in the range 1-4 Let the permutation factor be 3.

Step 5: Generate a random key based on crossover points, mutation points and crossover factor generated above.
Hence the symmetric key in an octal form is

2	5	1	7	2

Each octal digit in a symmetric key can be represented using 3 bits. Hence a symmetric key in a binary format is given by

In the above example, c = m = 2.
Hence Key Length = 15.

Hence, Crossover Point1 = 2
Crossover Point1 = 5

0	1	0	1	0	1	0	0	1	1	1	1	0	1	0

Hence the length of the key is 15 bits in our case which depends on the number of crossover points and mutation points. The length of the symmetric key can be computed using a general formula

Key Length = 3 (c + m + 1)

Where c-> No. of Crossover points
m-> No. of Mutation points

The same operation is repeated for the next set of blocks after permuting the digits to the left of the permutation factor of a symmetric key by a permutation factor.

The new key generated is

1	7	2	5	2

Applying the same operation on the next two blocks , we get,

Step 1: Extract next two 8-byte blocks from the text file to be encrypted. Let the two blocks be represented by

d0	d1	d2	d3	d4	d5	d6	d7

f0	f1	f2	f3	f4	f5	f6	f7

Step 2: Perform crossover operation. New crossover points are,

Crossover Point1 = 1
Crossover Point1 = 7

d0	d1	d2	d3	d4	d5	d6	d7

f0	f1	f2	f3	f4	f5	f6	f7

d0	f1	f2	f3	f4	f5	f6	d7

f0	d1	d2	d3	d4	d5	d6	f7

Step 3 : Perform mutation operation. New mutation points are,

Mutation Point1 = 2
Mutation Point2 = 5

d0	f1	128-f2	f3	f4	128-f5	f6	d7

f0	d1	128-d2	d3	d4	128-d5	d6	f7

The process is repeated for all pairs of blocks in a text file.

V. IMPLEMENTATION IN JAVA

Files used in the program.

Filename	Description
ga.txt	Contains the text to be encrypted
Encrypt.txt	Contains the encrypted data
Decrypt.txt	Contains the decrypted data
Key.txt	Contains randomly generated symmetric key.

Contents of the various files are given below:

Ga.txt

this file will be encrypted using genetic algorithm and then will be decrypted again.

Encrypt.txt

th i lile s`f be sypte e ring l otic e erith n`wd th nill -t decry e`d a

Decrypt.txt

this file will be encrypted using genetic algorithm and then will be decrypted again.

Key.txt

25242

VI. REFERENCES

1. *David. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Pearson Education, 1989, ISBN-13: 978-020115767.*
2. *X. F. Liao, S. Y.Lai and Q. Zhou. Signal Processing. 90 (2010) 2714–2722.*
3. *H. Cheng and X. Li. IEEE Transactions on Signal Processive. 48 (8) (2000) 2439–2451.*
4. *O. Lafe. Engineering Applications of Artificial Intelligence. 10 (6) (1998) 581–591.*
5. *R. J. Chen and J. L. Lai. Pattern Recognition. 40 (2007) 1621–1631*
6. *S. Li, G. Chen and X. Zheng. Multimedia security handbook. LLC, Boca Raton, FL, USA: CRC Press; (2004) [chapter 4].*
7. *Y. Mao and G. Chen. Handbook of computational geometry for pattern recognition, computer vision, neural computing and robotics. Springer; (2003).*
8. *H. S. Kwok, W. K. S. Tang, Chaos Solitons and Fractals, (2007) 1518–1529.*
9. *Mohammad Sazzadul Hoque, Md. Abdul Mukit and Md. Abu Naser Bikas, An Implementation of Intrusion Detection System Using Genetic Algorithm, International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012*
10. *L.M.R.J Lobo, Suhas B. Chavan, Use of Genetic Algorithm in Network Security, International Journal of Computer Applications (0975 – 8887)Volume 53– No.8, September 2012*
11. *W. Lu, I. Traore, "Detecting New Forms of Network Intrusion Using Genetic Programming". Computational Intelligence, vol. 20, pp. 3, Blackwell Publishing, Malden, pp. 475-494, 2004.*
12. *M. M. Pillai, J. H. P. Eloff, H. S. Venter, "An Approach to Implement a Network Intrusion Detection System using Genetic Algorithms", Proceedings of SAICSIT, pp:221-228, 2004.*
13. *S. M. Bridges, R. B. Vaughn, "Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection", Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp. 109-122, 2000.*
14. *M. Middlemiss, G. Dick, "Feature selection of intrusion detection data using a hybrid geneticalgorithm/KNN approach", Design and application of hybrid intelligent systems, IOS Press Amsterdam, pp.519-527, 2003.*