



# International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

(Open Access, Double Blind Peer-reviewed, Refereed and Indexed Journal)

[www.iasir.net](http://www.iasir.net)

## Shared Key Generation in Conventional Cryptography using GA

Dr. Vikas Thada, Mr Utpal Shrivastava

Department of Computer Science and Engineering,  
Amity University, Gurgaon, Haryana, INDIA.

**Abstract:** Cryptography is an art and science of securing data. Genetic Algorithm (GA) is a class of evolutionary algorithms for optimization and search problems. This paper presents an approach for generation of secret keys using GA. The research work carried out in this paper is based on selecting symmetric key for shared key cryptography using GA in a random manner and considering the weights of different characters that are part of the generated key. The paper is proposing a new approach for the generation of secret key using GA. The quality of the key is determined using the fitness function which takes into account alphabets, digits and special characters. Further the GA parameters probability of crossover and probability of mutation also affect the fitness of generated key. The implementation was carried out on MATLAB 9 and results were quite satisfactory.

**Index Terms:** Encryption, Decryption, Cryptography, Genetic algorithm, Crossover, Mutation, Selection, Secret key cryptography

### I. INTRODUCTION

Cryptography is art and science of securing data. Genetic Algorithm (GA) is a class of evolutionary algorithms for optimization and search problems. This paper presents application of GA in the field of cryptography. The research work carried out in this paper is based on selecting symmetric key for shared key cryptography using GA in a random manner and considering the weights of different characters that are part of the generated key. The paper is proposing a new approach for the generation of secret key using GA. The quality of the key is determined using the fitness function which takes into account alphabets, digits and special characters.. Although there are many tools available for the generation of symmetric key a new approach which make use of genetic algorithm is used. The algorithm produces shared key of any desired size consisting of alphabets, digits and special characters. All the keys generated are always unique. The keys generated can also be used as randomly generated password by varying the length of the key generated.

The paper is organized as follows: next section gives an introduction of Genetic Algorithm, this is followed by basic idea of cryptography, next section discusses about experiments & result, and finally conclusion is presented.

### II. GENETIC ALGORITHM

The GA is a search based on the mechanics of natural selection. The core thinking behind this is notion of adaptability to some kind of environment, the way normal natural system of human and animals work. This generally means that GA is based on the principal of survival of fittest. In GA reproduction of an individual is promoted by discarding the weaker individual and survival of stronger one.

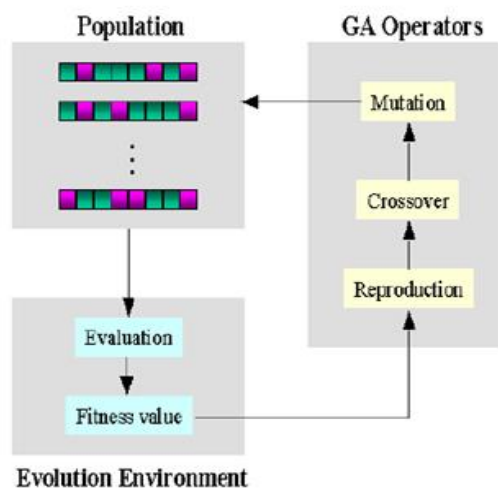


Figure 1: Genetic algorithm

Basically, a GA starts with a population (set of individuals) which may be randomly generated or collected, it depends upon problem at hand which is optimized. Following the generation of population the genetic algorithm directly enters in a loop. At the end of every round known as iteration, a newly population has been produced by applying a certain number of stochastic operators (selection, crossover, mutation) to the old population. That type of iteration is known as a generation. The GA uses two reproduction operators - crossover and mutation. In the crossover operation some of the genes from two chromosomes is swapped. Result of this swapping operations is birth of two new offspring. The type of swapping is determined by type of crossover operation like one point crossover, two point crossover, random crossover etc. The parameter that controls crossover is known as probability of crossover (ProC). It determined what % of total population goes through crossover operation. By combining traits of two different chromosomes GA forms an optimum solution

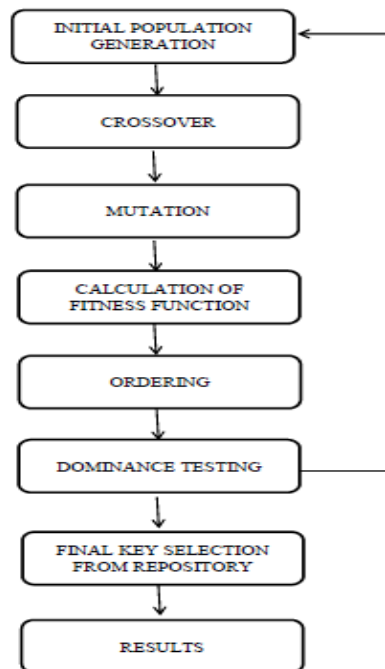


**Figure 2: Step by step implementation flowchart**

### III. CALCULATING FITNESS FUNCTION

For getting an optimal value among many different solutions fitness function is used. The function is written depending upon problem in consideration to judge the fitness of a solution i.e. how optimal it is among other possible solutions. In the proposed solution, all the keys which consists of digits, special characters, and alphabets are converted into decimal representation to perform following test on them.

#### A. Frequency Test

The purpose of frequency test is to determine randomness of the generated keys. In this test individual members of the calculated to check repetition of chromosomes [10].

#### B. Gap Test

The reason for conduction of this test to see the effects of the interval between recurrences of the same character. This is to find out how far are the two repeating characters in the chromosome.[11].

#### C. Dominance Testing

Following the tests population is arranged in descending order of their fitness value. For performing dominance testing key are selected randomly. Pairing of the key with highest fitness value is done with other keys (randomly chosen). After pairing hamming distance between each pair is calculated. Now from these pair one key will be chosen randomly so that it dominates over others and that serves as final key.

#### D. Final Key Selection from Repository

Selection operation is followed by fitness evaluation operation. Motivation behind selection process is to keep strongly fitter candidates in the population. Thus higher fitness valued chromosomes goes through reproduction and lower fitness valued chromosomes are discarded. The idea is to give preference to better individuals. In the

selection operation fitter chromosomes are duplicates and chromosomes with poor fitness values are eliminated [13].

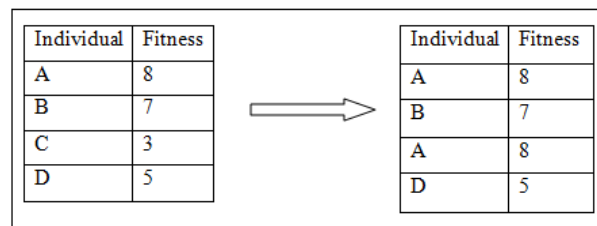


**Figure 3: Selection Operator on a Population of 4 Individuals [9].**

The research work uses roulette wheel selection method as selection operator (Wikipedia 2014). It is also known as fitness proportionate selection method. These steps are repeated N number of times till there start improvements in the new population.

#### IV. CRYPTOGRAPHY

The art of secret writing is cryptography. In encryption plain readable text is converted in a form that is not readable and known as cipher text. The cipher text can back be converted using decryption. Keys once used are discarded usually. The key containing unique characters or less frequent characters does not give any clue to the attacker and considered as strong and reliable. The field of cryptography encompasses within it plenty of ciphering techniques and algorithms. The process of generating a key in cryptographic algorithms is of utmost importance and secrecy. The generation of key using GA has not been explored by many researchers and same work is presented in this paper. Cryptography techniques are best suited for sharing of private or confidential messages over insecure communication channels. There exist two main types of cryptography: (a) Symmetric key cryptography (b) Asymmetric key cryptography

In asymmetric key cryptography two keys are used. The public key is used for performing encryption and other key (private) is used for performing decryption.

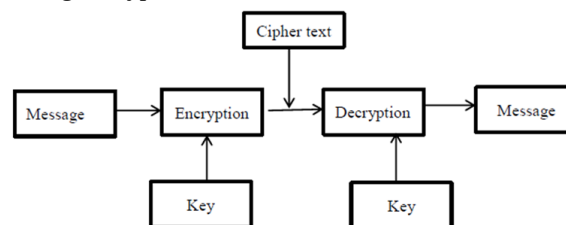


**Figure 4: Symmetric Key Encryption**

Shared key /symmetric/ cryptography is uses one single key both for encryption and decryption. The major concern in symmetric key cryptography is of key exchange.

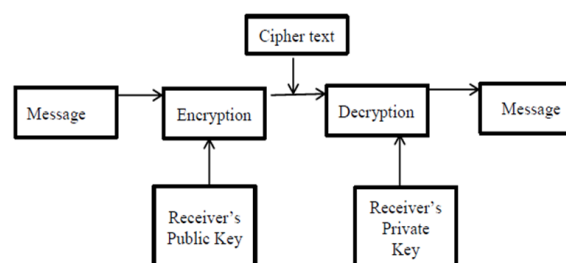


**Figure 5: Asymmetric Key Encryption**

#### V. EXPERIMENTATION

The research work starts with a randomly generated population of candidate solutions. The individual's fitness is calculated their weights. The weights have been assigned to digits, alphabets and special characters. Highest weight is given to special characters, followed by digits and then alphabets. An initial population is randomly generated composed of several chromosomes which are either binary or hex depending on the type of population. The complete process of GA comprising fitness calculation, selection operator (roulette wheel),

crossover, mutation is applied for different iterations. Based on the fitness chromosomes are selected to engender the next generation having fitness value better than the previous generation. The experiment was carried out using different probability of crossover and mutation for different iterations. The result obtained are presented in tables 1-3

**Table 1: Fittest key generation for Pc=0.9 and Pm=0.01**

| CROSSOVER RATE = 0.9, MUTATION RATE = 0.01 |                         |                         |
|--------------------------------------------|-------------------------|-------------------------|
| ITERATIONS                                 | FIVE FITTEST KEYS       | FITTEST KEY AFTER 5 RUN |
| 50                                         | 205, 225, 210, 220, 225 | K@5I3GGH3X&G@Z          |
| 100                                        | 195, 190, 195, 175, 180 | GM9I%*0M8B92ZCK3        |
| 150                                        | 210, 185, 215, 205, 195 | IL/%D%VH5Q+CNRL         |
| 200                                        | 230, 200, 220, 220, 195 | ^&A-N9*OR*@XK#SJ        |
| 250                                        | 190, 200, 220, 235, 190 | I&0NW#HZV*QYV1--        |
| 300                                        | 205, 215, 220, 190, 200 | A^NYD+K#5-^D*3Z&        |
| 350                                        | 200, 255, 220, 200, 205 | ^NVY-5CD+O-%&&-+        |
| 400                                        | 205, 205, 185, 255, 195 | *%LK#++##5-T2DNV*       |
| 450                                        | 245, 190, 215, 195, 200 | #2*O&@/^XBKQQ+K^        |
| 500                                        | 150, 235, 250, 180, 180 | +R+&+I3JL*G^TO&%        |

**Table 2: Fittest key generation for Pc=0.7 and Pm=0.01**

| CROSSOVER RATE = 0.7, MUTATION RATE = 0.01 |                         |                         |
|--------------------------------------------|-------------------------|-------------------------|
| ITERATIONS                                 | FIVE FITTEST KEYS       | FITTEST KEY AFTER 5 RUN |
| 50                                         | 235, 195, 200, 180, 215 | OZ-UH#^ZU^!&&^S2        |
| 100                                        | 200, 195, 185, 220, 260 | H^*-%E***^#KI%G%        |
| 150                                        | 175, 210, 185, 230, 200 | I9-%^0^9@Y-VX%IE        |
| 200                                        | 215, 180, 215, 200, 200 | &CMXJ%D6H9L^BMMN        |
| 250                                        | 180, 210, 225, 195, 215 | E@QPURR%8+&-M^IOZ       |
| 300                                        | 205, 215, 175, 175, 240 | L%ZBM^E-t@%N5VJ         |
| 350                                        | 185, 240, 210, 220, 215 | F&Z@-P-OUNLYV*4*8       |
| 400                                        | 215, 195, 225, 170, 195 | U@-8#-L7D&#VN@3Q        |
| 450                                        | 195, 240, 205, 210, 170 | DWOMJ^K^*^@*JEA^        |
| 500                                        | 175, 185, 195, 185, 180 | VTG+ZUACA8*DS+&H        |

**Table 3: Fittest key generation for Pc=0.5 and Pm=0.01**

| CROSSOVER RATE = 0.5, MUTATION RATE = 0.01 |                         |                         |
|--------------------------------------------|-------------------------|-------------------------|
| ITERATIONS                                 | FIVE FITTEST KEYS       | FITTEST KEY AFTER 5 RUN |
| 50                                         | 210, 225, 225, 215, 210 | D@M^B^OCG%*J-MK!        |
| 100                                        | 205, 210, 180, 220, 195 | ##%WXRST2L@^3-~&Q       |
| 150                                        | 195, 190, 205, 195, 220 | GGMOLIFZHP%~-N2@        |
| 200                                        | 175, 215, 215, 185, 210 | IS^GC%KO-K8**SD#        |
| 250                                        | 210, 180, 220, 190, 215 | *C*B&%OERAZC1%Y         |
| 300                                        | 205, 185, 190, 190, 195 | 8X^+O@3*VH9-QTC         |
| 350                                        | 210, 200, 205, 185, 210 | #395&W&ELB@-2R+         |
| 400                                        | 190, 210, 195, 200, 190 | &B%#K@LR7%K5U&FI        |
| 450                                        | 245, 210, 200, 200, 225 | -I7IK#L#%G-^+JU-        |
| 500                                        | 190, 245, 210, 215, 215 | ZG*%DL+IS#%R^!@-        |

## VI. CONCLUSION

The entire process of key generation was found to be completely random in nature and keys generated were completely unique. The fitness values of the keys were calculated and analyzed. Keys strength and randomness were judged using the Frequency and Gap Tests. Mutation rate was selected as 0.01 and crossover rate was varied from 0.5-0.7 with gap of 0.2. Various tests were conducted on the sample population generated and the results were quite good. Random samples were created by generating an initial random population of 320 genes of total 20byte chromosomes. An encryption method based on GA which is used to generate keys by pseudorandom number generator is introduced. The work has been implemented and analyzed. Using Genetic Algorithm one can keep the strength of the key to be long, we are still on working to make the complete algorithm better enough. After generating 20 byte chromosomes size, crossover function was applied taking the total population in which each chromosome is of 16 byte, using the crossover rate as 0.5,0.7,0.9.

## ACKNOWLEDGMENTS

The authors wish to thank various authors and researchers whose research work has proved a great source of help for this paper. Authors also thank Google for bringing quality and relevant results.

## REFERENCES

- [1] Y. Mao, G. Chen 2003. "Handbook of computational geometry for, computer vision, pattern recognition, robotics, and neural computing" Springer, 2003.
- [2] A. Kumar, N. Raj pal, A. Tayal 2005, "Security System for Multimedia Data Transmission Using Genetic Algorithms" NCC, 05 Presented in the Kharagpur (IIT),
- [3] A. Tragha, F. Omary, A. Mouloudi 2006. "ICIGA: improved cryptography inspired by genetic algorithm" at IEEE 2006.
- [4] Dr. D. Singh, P. Rani, Dr. R. Kumar 2013, "To design a GA for cryptography to enhance the security", issue 2 April 2013.
- [5] R. Afarin, S. Mozaffari 2013, "Image encryption using genetic algorithm" at MVP and IEEE 2013.
- [6] S.Geetha, Siva S.Sindhu, A. Kennan 2006, "An active rule based approach to audio steganalysis with a genetic algorithm" at IEEE 2006.

- [7] A. Almarimi, A. Kumar, I. Almerhag, N. Elzoghbi 2013, "A new approach for data encryption using GA" at LIBYA. 2013.
- [8] S. Mishra, S. Bali 2013, "Public key cryptography using GA" at IJRTE 2277-3878, volume2, issue 2 May 2013.
- [9] Dr. Poornima G. Naik , Girish R. Naik 2014, "Asymmetric key encryption using GA" at IJLTET, volume3, issue 3 Jan 2014.
- [10] S. Jawaid, Adeeba Jamal 2014, "Generating the best fit key in cryptography using GA" at IJCA (0975-8887) volume 98 no.20, issue July 2014. page-579-583, 28-20 Jan 2005.
- [11] Bethany Delman 2004, "Genetic algorithms in cryptography", 7-1-2004.
- [12] Abdel-karim S.O. Hassan, Ahmed F. Shalash and Naglaa F. 2014, "Modifications on RSA cryptosystem using genetic optimization" at IJRRAS 19 (2) May 2014.
- [13] A Kumar, N Raj pal 2004, "Application of Genetic Algorithm in the Field of Steganography" , in Journal of Information Technology (IJIT), volume2, no.1, Jul-Dec.2004
- [14] P. Singh, G. Gosawi, S. Dubey 4 (2014). "GA: A technique for cryptography real time data transmission" at BJDMAN 37-40.