Genetic Key Generation for Public Key Cryptography

Sonia Goyat

***Abstract:** The importance of cryptography can be judged by the fact that it is used almost everywhere. It is essential in e transactions, LAN data transfer, in Databases and even while storing data in our own computer. There are many methods of cryptography. Some of that have large complexity and are pretty cumbersome, while other are based on the theory of natural selection. The fact that selecting key for the public key cryptography is a selection process in which various keys can be categorized on the basis of their fitness, makes Genetic Algorithms a good contender for the process to be followed for generating key. Moreover, using Genetic Algorithm we can keep the strength of the key as good as any other key, still make the whole algorithm good enough to have a complexity as low as $O(n^2)$. The work is based on one such approach [1] and modifies the approach to generate keys that have more strength as compared to previous work [1].The work has been implemented and analyzed. The results obtained are good in terms of coefficient of autocorrelation. The samples satisfy the tests including gap test, frequency test etc.*

***Keywords:** Public Key Cryptography, One Time Pad, Genetic Algorithms, Vernam Cipher.*

I. INTRODUCTION

Cryptography:

Key generation in cryptography has been dealt with in many papers but the use of GA in the process has not as yet been explored successfully. It is the most important part of encoding the data .A non repeating key guarantees better results and generates a code that is theoretically impossible to break .Some of the classical techniques used for generating unique keys are OTP and Pseudo random number generators. The work tries to explore use of non-conventional techniques in the process.

Vernam Ciphers:

In this process, the plaintext is converted into cipher text by XORing the binary plaintext with a binary key. The Cipher Text is transferred via a channel and when the receiver receives the cipher text and XORs it with the same key thus getting the plaintext again. It has been proved that if the key, that is One Time Pad, is unique then the cipher text cannot be broken. m1 m2…..mt is operated on by a binary key string k1k2….kt of the same length to produce a cipher text string c1c2…..ct using Eqn 1

$$ci = mi \oplus ki\ ,\ 1 \leq i \leq t \qquad (1)$$

If the key string is randomly chosen and never used again, this cipher is called a one-time system or one-time pad.

GAs are adaptive heuristic search algorithms which are based on the Charles Darwin theory of survival of the fittest. The main idea behind these algorithms was to replicate the randomness of the nature. This required that the algorithm proposed should behave like a natural system. GAs emulate the nature to large extent .GAs produce a population in such a way that the trait which is popular ,that is, has higher fitness value is replicated more, as is done by the nature. This is also the fundamental concept behind evolution. So these algorithms are also referred as the evolutionary algorithms.

II. KEY GENERATION

The papers pertaining to the application of Genetic Algorithms to cryptography have been studied [1], [2]. the strength of these have been tested using various methods. Mostly these methods included various types of attacks. The point which we intend to make in this paper is that if the quality of the random numbers produced by the method is good then the key generated will always be strong. The various types of ciphers have been discussed in the section. This classification helps to compare the technique with the existing ones.

The various types of ciphers include mono alphabetic Substitution Cipher and permutation cipher. In the first case a key consist of all the possible permutation of an alphabet can break the cipher. To tackle this problem poly alphabetic substitution and permutation, transposition cipher were introduced. The permutation cipher is applied to a block of ciphers while columns transposition is applied to the entire text at once.

The work proposed intents to create a key as strong as the vernam cipher. If the key is randomly chosen and never used again, the cipher is called one time pad [1], [2]. The one time pad is theoretically unbreakable [3].

III. PROPOSED WORK

3.1 Population generation:

An initial population of chromosomes is generated in binary form.

3.2 Evaluation:

Each of the chromosomes which is in binary format is converted into the decimal number .The method for converting the binary chromosome which, in this case, is 25 bit into a number is multiplying the bit to 2^{-14} to 2^{15} staring from the first chromosome. After the above conversion randomness tests are performed on the values generated .The values which are produced after the completion of the process, if show a better results than this will prove the validity of the work.

Manuscript received on July, 2012.

Sonia Goyat, Student,M. Tech, Department of computer science and applications ,MDU,Rohtak.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

3.3 Threshold Check:

After the above step those values are selected which are greater than the threshold are selected.

3.4 Crossover:

One point crossover is to be performed on the population which remains after the above step.

3.5 Threshold Check:

The newly generated population after the crossover has few chromosomes which do not cater to our threshold. So a threshold check is required again to eliminate these chromosomes.

3.6 Mutation:

After step 4 some randomly selected bits are changed according to the mutation rate in few of the generated chromosomes which gives us a new population.

3.7 Threshold Check:

Mutation might lead to population which might not cater to our threshold. So, a threshold check is required again. Finally after this procedure we have fit population in accordance to our threshold. This population is stored in a file .The process is repeated 'n' times. The above steps leads to 'n' sets of populations .The randomness tests are performed on all these samples and the samples are ranked accordingly. The best sample is selected and the deviation of each chromosome value; from the Coefficient of autocorrelation (CC) of that sample; is calculated. This value will be denoted by Є. Now fitness value of each of the chromosome is calculated using the Eqn 2. The chromosomes are now arranged in decreasing order with respect to Fit (i).

$$\text{Fit}(i) = 1/(1+e\,\text{Є}) \qquad (2)$$

3.8 Roulette Wheel Selection

Replication is carried out by roulette wheel selection. Those values which are of high fitness are replicated in a greater frequency as compared to others.

3.9 Crossover and Mutation:

The process of Crossover and mutation is performed again and the final population is checked for threshold. The chromosome having the highest fitness value is selected.

Flow Diagram Of Genetic Random Key Generator

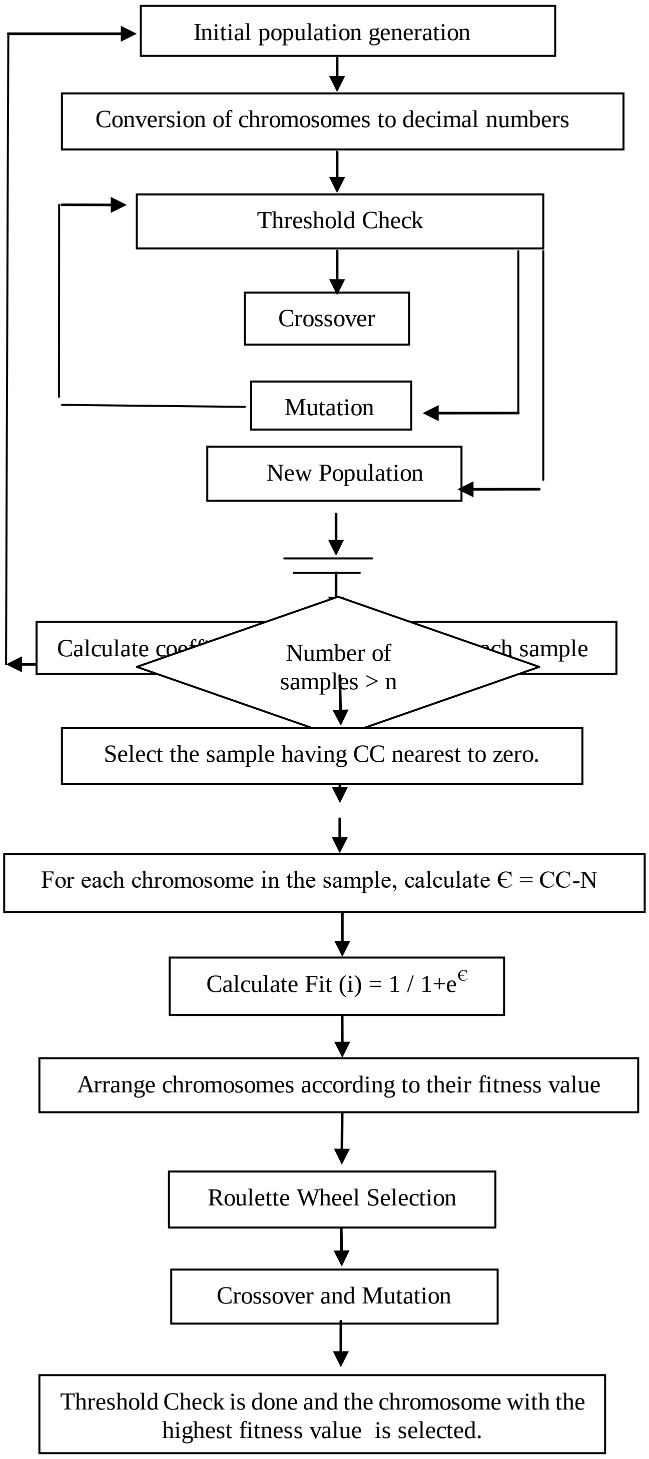
Initial population generation
Conversion of chromosomes to decimal numbers
Threshold Check
Crossover
Mutation
New Population
Calculate coefficient ... each sample
Number of samples > n
Select the sample having CC nearest to zero.
For each chromosome in the sample, calculate Є = CC-N
Calculate Fit (i) = 1 / 1+e^Є
Arrange chromosomes according to their fitness value
Roulette Wheel Selection
Crossover and Mutation
Threshold Check is done and the chromosome with the highest fitness value is selected.

Fig 1:

IV. RESULTS AND CONCLUSIONS

The work has been implemented and analyzed. The implementation has been done in C#. Samples have been collected and analyzed using Excel. Some common tests for random numbers have been applied on the sample . The sample seemingly satisfies almost all of them. Around a 500 values were analyzed and no repetition was obtained. The coefficient of autocorrelation was calculated. The result for k = 1 was 0.028, thus indicating a good random sample. The implementation proposed can generate good random sequences varying length and it is fast. 214 values were tested. Longer sequences haven't been tested. Testing of 214 values gave a good value as per the chi squared test. To test the randomness of the sample other tests can also be done but the time factor did not permit more tests. The various graphs obtained are as follows:

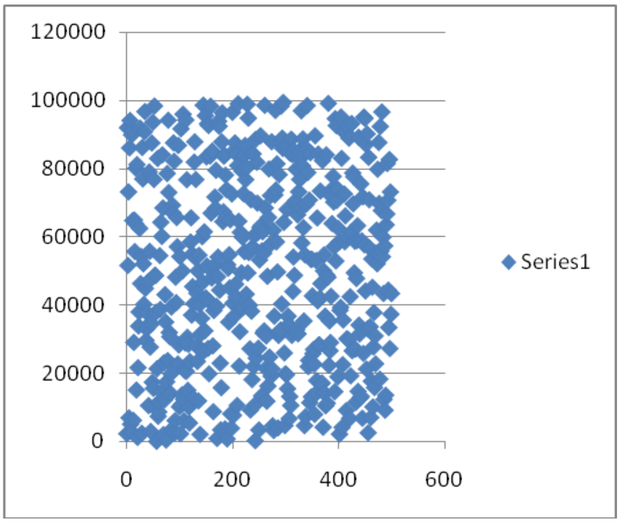

Figure 1: Sample

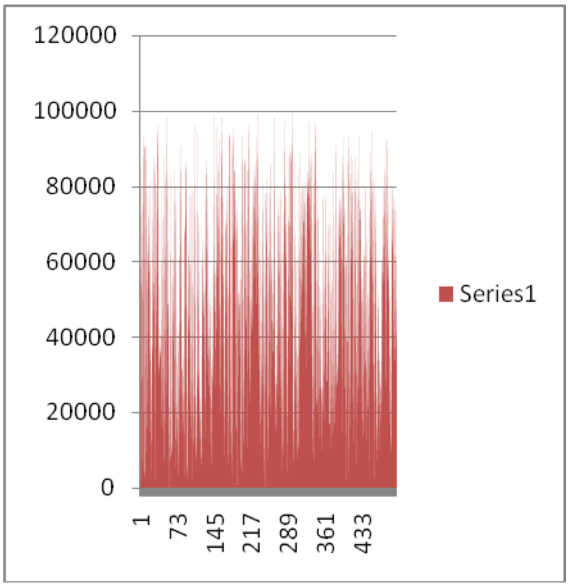

Figure 2: Sample division

REFERENCES

1. Harsh Bhasin, Nakul Arora, Reliability Infocom Technology and Optimization 2010, Conference Proceedings pages 226- 230.
2. Bethany Delman, Genetic Algorithms in Cryptography, MS Thesis 2004.
3. Norman D. Jorstad, CRYPTOGRAPHIC ALGORITHM METRICS, January 1997
4. ABDELSALAM ALMARIMI et al, A NEW APPROACH FOR DATA ENCRYPTION USING GENETIC ALGORITHMS, Published in: Proceeding CERMA '10 Proceedings of the 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference
5. Menezes, A., van Oorschot, P., & Vanstone, S. (1997). Handbook of Applied Cryptography Boca Raton: CRC Press
6. Harsh Bhasin, Supreet Singh, GA-Correlation Based Rule Generation for Expert Systems, IJCSIT, Volume 3, Issue 2, Pages 3733-3736
7. Harsh Bhasin, Surbhi Bhatia, Application of Genetic Algorithms in Machine learning, IJCSIT, Volume 2, Issue 5, Pages 2412-2415
8. Harsh Bhasin, Surbhi Bhatia, Use of Genetic Algorithms for Finding Roots of Algebraic Equations, IJCSIT, Volume 2, Issue 4, Pages 1693-1696
9. Harsh Bhasin, Gitanjali, Harnessing Genetic Algorithm for Vertex Cover Problem, International Journal on Computer Science and Engineering (IJCSE), Volume 4, Issue 2, 218 - 223.
10. A NOVEL APPROACH TO GENETIC ALGORITHM BASED CRYPTOGRAPHY Farhat Ullah Khan, 2012-04