www.ijsrnsc.org

Available online at www.ijsrnsc.org

IJSRNSC

Volume-1, Issue-2, June- 2013
Research Paper

Int. J. Sci. Res. in Network Security and Communication

ISSN: 2321-3256

Cryptography Automata Based Key Generation

Harsh Bhasin*[1], Neha Kathuria[2]

*[1]*Delhi Technological University, India*

[2]*MDU, India*

Received: 28 March 2013 Revised: 10 May 2013 Accepted: 12 June 2013 Published: 30 June 2013

Abstract-- Cellular Automata is one of the most fascinating disciplines. It is being used in many practical applications but it's used in cryptography is still not being explored. A technique of generating key for public key cryptography has been proposed in the work. The work also applies different tests to the data generated by the application developed to generate cryptography key. It is found that the key is generated follows the gap test and frequency test .Moreover, the coefficient of autocorrelation of the data generated is also apt. Therefore, the method can be safely used in public key cryptography. The strength of the key generated is being tested against AES.

Keywords-- Cryptography, Key Generation, Cellular Automata, Gap Test, Frequency Test.

I. INTRODUCTION

Cryptography is perhaps one of the most important tasks in computing. It is extensively used and hence is immensely important. There are many methods of generating the key. The range from core statistical algorithmic method to nature inspired technique. It's a our firm believe that nature inspired technique will present better results in lesser time as compared to core algorithmic technique. Moreover, it would be practically impossible to decode the text encrypted by such keys. The present work is an extension of our earlier work which proposed a Cellular Automata based technique to generate the key. The present work statistically analyzes the technique using various tests i.e. gap test, frequency test and scatter diagrams. The numbers generated by the application are good enough to be used as a key for cryptography. The key generated an order to encrypt the text should possess that a good random sequence possesses. In one of our earlier implementations random numbers were generated using blood samples. The technique was, however, extremely complicated. In another implementation Genetic Algorithm was used to generate the key. The technique was simple but it is being verified statistically. The present work is intended to fill the gaps in the earlier technique. The new technique is bound to be path breaking, if the comparison with AES yields positive results.

The rest of the paper has been organized as follows: section-2 represents literature review; section-3 explains genetic algorithms,
section 4 presents the background of the work, section 5 presents statistical analysis and the last section concludes.

II. LITERATURE REVIEW

Literature Review is an integral part of the research as it explores the existing methodologies and guides the researchers in proposing new methodologies. The literature review has been carried out as per the guidelines proposed by Kitchenham . The papers have been selected in accordance with the importance and verification techniques. The review has been unbiased and has been extremely helpful in proposing this technique. A team of lecturers and Assistant Professors across colleges was constituted to select the papers. Table1 presents the review briefly. The second column gives the name of the author, the third column presents the technique used and the fourth column presents the verification.

Sr. No	Author	Technique Used	Verification	Ref. No.
1.	Harsh Bhasin et. al.	Cellular Automata, Genetic Algorithms	Gap Test, Frequency Test	1
2.	Harsh Bhasin	Algorithm which has a complexity of $O(n^2)$	Gap Test, Frequency Test	2
3.	Harsh Bhasin	Cellular Automata	Coefficient of auto-correlation	5
4.	Sonia Goyat	Genetic Algorithm	Coefficient of auto-correlation	6
5.	Benjamin Jun et. al.	Hardware-based Intel random number generator	FIPS 140-1 randomness tests	7

Corresponding Author: *Harsh Bhasin*

6.	K.J. Jegadish Kumar et. al.	Cellular Automata	MATLAB 7.7	8
7.	Sonia Goyat	Genetic Algorithm	Frequency Test, Gap Test	9
8.	Uttam Kr. Mondal et. al.	Symmetric key block Cipher algorithm	Frequency Distribution Test, Bit Ratio Test, Non-Homogeneity Test,	10
9.	Farhat Ullah Khan et. al.	Genetic Algorithm	Gap Test, Frequency Test	11

Table1: Techniques of Producing Keys/ Random Numbers

III. GENETIC ALGORITHMS

A genetic algorithm is a search heuristic that mimics the process of natural evolution used to generate useful solutions to optimization and search problems. Genetic Algorithms are a subset of what we call evolutionary algorithm which solves optimization problem using techniques inspired by natural evolution such as inheritance, mutation, selection, and crossover .

John Holland from the University of Michigan started his work on genetic algorithms at the beginning of 60s. A first achievement was the publication of Adaption in Natural and Artificial System in 1975. Holland has two aims, first to improve the understanding of natural adaption process, second to design artificial systems having properties similar to natural systems. Holland method considers the role of mutation and also utilizes genetic recombination that is crossover to find the optimum solution.

Crossover and mutation are two basic operators of GA. Performance of GA depend on them. Type and implementation of operators depends on encoding and also on a problem.

There are many ways of how to do crossover and mutation.

4.1. Crossover

1. Single point crossover -

In this case one crossover point is selected, binary string from beginning of chromosome to the crossover point is copied from one parent, and the rest is copied from the second parent.

11001011+1101**1111** = 11001111

2. Two point crossover -

Here two crossover point are selected, binary string from beginning of chromosome to the first crossover point is copied from one parent, the part from the first to the second crossover point is copied from the second parent and the rest is copied from the first parent.

110010**11** + 110**111**11 = 11011111

3. Uniform crossover -

In this method bits are randomly copied from the first or from the second parent.

11001011 + 11011101 = 11011111

4.2. Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of algorithm chromosomes to the next. It is similar to biological mutation.

Method given in most of the sources including Wikipedia:

An example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified. This mutation procedure, based on the biological point mutation, is called single point mutation. Other types are inversion and floating point mutation. When the gene encoding is restrictive as in permutation problems, mutations are swaps, inversions and scrambles.

The purpose of mutation in GAs is preserving and introducing diversity. Mutation should allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other.

4.3. Reproduction and Selection

Chromosomes are selected from the population to be parents to crossover. According to Darwin's evolution theory the best ones should survive and create new offspring. There are many methods how to select the best chromosomes, for example roulette wheel selection.

1. Roulette Wheel Selection

Parents are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have. Imagine a roulette wheel where are placed all chromosomes in the population, every chromosome has its place big accordingly to its fitness function.

IV. BACKGROUND:

In the earlier paper, we have used the combination of two technique called Cellular Automata and Genetic Algorithms. Cellular Automata was used to generate 256 patterns which were stored in 256 arrays having dimensions 100×100. This was followed by the encoding of patterns in the Genetic Algorithm Population. Each chromosome had 22 cells in the population. Of these 14 cells represent rows and columns in an array and remaining 8 cells depicted pattern number of cellular automata. To the initial population Genetic Operators like Crossover, Mutation and Selection were applied to craft new chromosomes. The type of crossover used was single point crossover in which two random numbers and a crossover point were generated. The number of times crossover was carried out is given by the following formula:

Number of Crossovers = (Crossover rate* number of rows*number of columns)/100

This was followed by the mutation operator being applied on crossover population. . After the mutation the fittest chromosome was calculated. The chromosome depicted the element of the pattern to be selected. 256 such elements were generated and finally a key was formed. The technique was verified by frequency and gap test.

V. VERIFICATION

The above technique has been implemented and samples have been generated. The implementation has been done in C#. Samples have been collected and analyzed in Microsoft Excel, Various tests have been applied on the sample and most of them give satisfactory results.

Since, around a 3000 values were analyzed and no repetition was obtained therefore frequency test was not applied. The coefficient of autocorrelation was calculated for k = 1 to k = 19. The result for k = 1 was 0.031, thus indicating a good random sample. Coefficient of auto-correlation is given by following formula.

$$r_k = \frac{\sum_{i=1}^{N-k} (Y_i - \bar{Y})(Y_{i+k} - \bar{Y})}{\sum_{i=1}^{N} (Y_i - \bar{Y})^2}$$

In a next test, a time series has also been considered and Karl Pearson Coefficient of correlation has been calculated, also giving satisfactory data.

In the analysis of data, correlogram have been plotted which is an image of correlation statistics. The randomness is ascertained by computing autocorrelations for data values at varying time lags. If random, such autocorrelations should be near zero for any and all time-lag separations. If non-random, then one or more of the autocorrelations will be significantly non-zero. Such correlogram has also been plotted for our sample data.

The majority was calculated by taking a group of 10 cells. If more samples are needed then a set of 5 cells can also be taken. The whole process is being enhanced and analyzed.

VI. CONCLUSION

The technique proposed has been verified and tested for various values of k. The gap test, frequency test, values of autocorrelation instills the confidence in the validity of the technique. It may also be stated that the use of cellular automata in cryptography is sure to be path breaking, as only nature inspire techniques can complete with AES.

REFERENCES

[1] Harsh Bhasin, Ramesh Kumar, Neha Kathuria, 'Cryptography using Cellular Automata', International Journal of Computer Science and Information technologies, Vol. 4 (2), 2013.

[2] Umesh Kumar Singh, Shivlal Mewada, Lokesh Laddhani and Kamal Bunkar, "An Overview & Study of Security Issues in Mobile Ado Networks", International Journal of Computer Science and Information Security (IJCSIS) USA, Vol-9, No.4, pp (106-111), April 2011.

[3] Harsh Bhasin, "Corpuscular Random Number Generator", International Journal of Information and Electronics Engineering, Vol. 2 (2), 2012.

[4] Harsh Bhasin, Nakul Arora, " Cryptography Using Genetic Algorithm", Reliability Infocom Technology and Optimization,2010

[5] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, Stephen Linkman, "Systematic literature reviews in software engineering – A systematic literature review", Journal Information software technologies", Vol. 51 (1), 2009.

[6] Harsh Bhasin, "Use of Cellular Automata Patterns in Cryptography" in National Conference on Advances in Computational Intelligence, 2011.

[7] Sonia Goyat, "Cryptography using Genetic Algorithm", IOSR Journal of Computer Engineering, Vol. 1 (5), 2012.

[8] Benjamin Jun, Paul Kocher, "The Intel Random Number Generator", Cryptography Research, Inc. White Paper Prepared For Intel Corporation, 1999.

[9] K. J. Jegadish Kumar, K. Chenna Kesava Reddy, S. Salivahanan, "Novel and Efficient Cellular Automata based Symmetric Key Encryption Algorithm for Wireless Sensor Networks", International Journal of Computer Applications Vol. 13 (4), 2011.

[10] Sonia Goyat, "Genetic Key Generation For Public Key Cryptography", International Journal of Soft Computing and Engineering (IJSCE), Vol. 2 (3), 2012.

[11] Uttam Kr. Mondal, Satyendra Nath Mandal, J. Pal Choudhury, J. K. Mandal , "Frame Based Symmetric Key Cryptography", International Journal Advanced Networking and Applications 762 Vol. 2 (4), 2011.

[12] Farhat Ullah Khan, Surbhi Bhatia, "A Novel Approach To Genetic Algorithm Based Cryptography", International Journal of Research in Computer Science, Vol. 2 (3), 2012. Introduction

