ISSN:0975-9646

Cryptography Using Cellular Automata

Harsh Bhasin, Ramesh Kumar, Neha Kathuria
CSE,
MDU, India

***Abstract*-- Cryptography is one of the most essential ingredients of network security. The cryptographic algorithm based on mathematical premise can be backtracked and the text can be decoded. On the other hand the artificial life approaches render backtracking impossible. The work proposes a cryptographic algorithm based on Cellular Automata and Genetic Algorithm. The technique has been implemented and the results so far indicate that the technique is good enough to compete with AES. The work is being enhanced and tested for larger data samples the analysis is carried out using standard tests. The work promises to open the door of Cellular Automata to cryptography.**

***Keywords*-- Cellular Automata, Cryptography, Genetic Algorithms, Coefficient of auto-correlation, Key strength.**

I. INTRODUCTION

Importance of network security is growing day by day. The evolution of programming has crossed the stages of correctness and usability to reach the point where the importance of security has been rightly realized. The essence of the above is development of a key which cannot be guessed. It will be ideal situation whereby even the brute force algorithms are unable to decide the key in a fixed time. This calls for development of a technique which is nature inspired and is statistically sound; still, it should not be a mathematical procedure. The work intends to use Cellular Automata for the above purpose. This work is extension of earlier work [11]. A cellular automaton is chosen for the work as it correctly imitates the sociological behavior. The importance of cellular automata can be judged from the facts that it is used to manage the Water Distribution System [12] and even the pattern of migration in world.

Cryptography is the art of conversion of a plain text to a text which is hard to decode without a key. The key in this case can be produced in two ways: the first method is mathematical in nature. AES and DES fall in this category. These algorithms are statistically sound but can be broken if mathematics behind them is known. Second types of algorithms are nature based and hard to decode. But they require rigorous statistical analysis.

The aim of the work is as follows:-

1) To develop a technique of public key cryptography using cellular automata.
2) To verify the technique statistically.

The rest of the paper has been organized as follows. Section two explains Genetic Algorithms. Section three of the paper explains the concepts and principles of cellular automata. Section four of the paper describes the technique of public key cryptography Section five of the paper presents the proposed technique and exemplifies it. Section six discusses the result and the last section concludes discussing future scope.

II. GENETIC ALGORITHM

Genetic algorithms are heuristic search algorithms based on the concept of natural selection [2, 6, 7, 8]. The principle behind them is natural genetics. These algorithms have ethos of human search. They are not just random algorithms rather sophisticated search algorithms which make use of historical data to generate new data. The technique is adaptive and inspired by nature.

The main aim of genetic algorithms is robustness [6]. As human follow biological method to accomplish a goal, same type of imitation is mocked in genetic algorithms. The various operations used in genetic algorithms try to achieve robustness. But it may be noted that, along with robustness, optimization is also important. An important characteristic of genetic algorithms is encoding of parameter set, which implies that genetic algorithms do not work on parameter itself. It may also be noted that the transition rules applied by genetic algorithms are probabilistic not deterministic [3]. What is needed in genetic algorithms is the optimal value of objective function which can be obtained by problem reduction approach of algorithm analysis and design.

Crossover and mutation are two basic operators of GA [1]. Performance of GA depend on their type and implementation. There are many ways of implementing crossover and mutation [1].

A. 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 [1].

10101111 + 111**00110** =10100110

2) *Two point crossover-*

Here two crossover points 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 [1].

10101**111** + 111**00**110 =10100111

3) Uniform crossover-

In this method bits are randomly copied from the first or from the second parent [3].

11001011 + 11011101 = 11011111

B. Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population to the next [7]. It is similar to the biological mutation [1, 7].

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 [7, 10].

C. 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 to select the best chromosomes, Roulette Wheel Selection is one of them[1,2,9].

1) Roulette Wheel Selection

Parents are selected according to their fitness [1]. The better the chromosomes are, the more chances to be selected they have [1]. Imagine a roulette wheel where are placed all chromosomes in the population, every chromosome has its place big accordingly to its fitness function [1, 4].

III. CELLULAR AUTOMATA

The history of CA dates back to the forties. It was started by Stanislaw Ulam. His aim was to study the evolution of graphic construction generated by simple rules. The base of the construction was a two-dimensional space divided into cells. Each of these cells had two states: ON or OFF. Starting from given pattern the new generations were formed according to the neighborhood rules. Ulam found that this mechanism permitted to generate complex figures and these figures may self-replicating. Simple rules were made to build very complex pattern. The question that need to be analyzed is whether the complexity is apparent or real. These rules were studied by John Von Neumann for use in self-productive automata [5]. He worked on the conception of a self-productive machine, the "kinematic". Such a machine was supposed to be able to reproduce any machine described in its programs; including a copy of itself. This led to libration from real physical constraints to work in an extremely simplified universe that was able to generate high complexity. Van designed 29 states cells, containing a universal replicator, a description of itself and a Turing machine for supervision. CA left laboratories in 1970 with the now famous Game of Life of John Horton Conway.

The cellular automaton is depicted in Figure1.

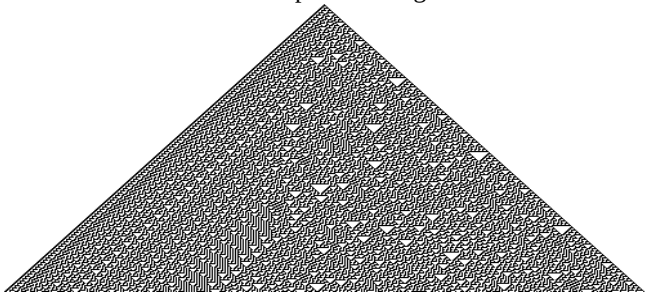

Figure 1: Cellular Automata

IV. PROPOSED WORK

The technique uses the concept of cellular automata and genetic algorithms to produce a key which can be used for public key cryptography. The steps to generate the key have been explained as follows.

Step 1: Cellular Automata is used to generate 256 patterns which are stored in 256 cells and having dimension in 100x100. For example, rule 18 is one of the most interesting rules of the cellular automata. The rule depicted in Figure 2.

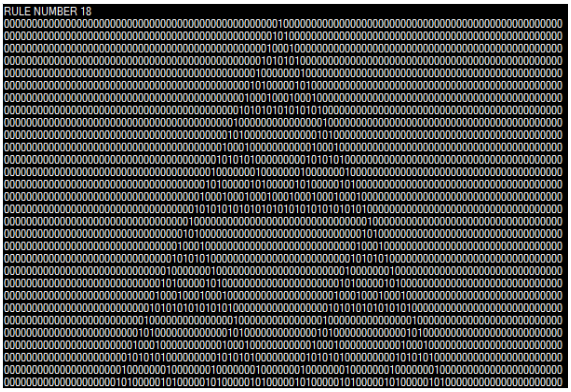

Figure 2: Cellular Automata Pattern (Rule 18)

Step 2: These patterns are depicted by chromosomes of genetic population. Each chromosome contains 22 cells, 8 cells represent pattern number remaining 14 cells represent the row and column of the array.

Step 3: Initial population is generate as per the following code:

```
for(i=0;i<n;i++)
{
        for(j=0;j<n;j++)
        {
                if(rand()%100>50)
                {
                        pop[i][j]=1;
                }
                else
                {
                        pop[i][j]=0;
                }
        }
}
```

Step 4: After population is generated crossover is performed. The crossover used in the work is single point crossover. Crossover is done as per following procedure:

```
n1=rand()%n
n2=rand()%n
x=rand()%100
```

The process of crossover is depicted in Figure 3.

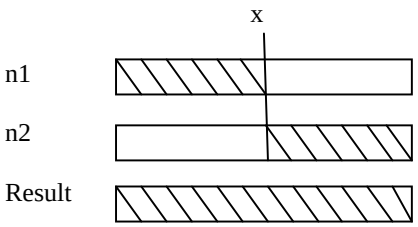

Figure 3: Crossover

In above procedure n1, n2 are two numbers which are randomly selected and x is a crossover point. Number of crossovers is calculated by the following formula.
No. of Crossover= (Crossover rate*row*column)/100

Step 5: Mutation is carried out as per following procedure. The mutation operator is depicted in Figure 4.

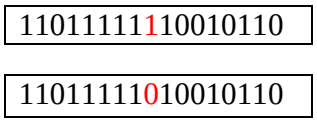

Figure 4: Mutation Operator

Step 6: Finally fitness is evaluated for each chromosome and fittest chromosome is selected. That chromosome depicts that element of the pattern to be selected, 256 such elements are generated and hence a key is formed.
The process has been depicted in Figure 5.

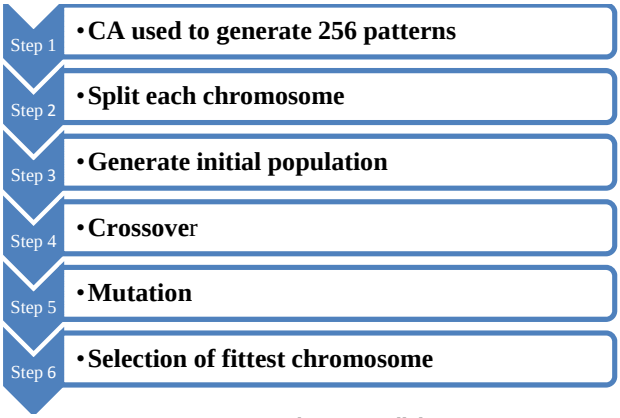

Figure 5: Cryptography using Cellular Automata

V. RESULTS AND CONCLUSION

The proposed work has been implemented using C#, .NET Framework 4.0. The application developed generates cellular automata patterns of different rules and is capable of storing the patterns in text format.
Figure-4 depicts a snapshot of the application. The technique has been tested for 3000 values. Frequency test and gap test have been applied giving satisfactory results. The coefficient of auto-correlation for K=1 is 0.2 for the values obtained so far. Coefficient of auto-correlation is given by following formula.

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

The technique seems sound enough to compete with the existing methodologies. The importance of Cellular Automata is being realized in every sphere, from migration of population to Biology. Cellular Automata is being use extensively. The above work is intends to amalgamate this fascinating technique with one of the most essential procedures in computer science i.e. cryptography to develop a technique which is bound to be pivotal in the discipline.

REFERENCES

[1] H. Bhasin, S. Bhatia, (2011), "Application of Genetic Algorithms in Machine Learning", *Harsh Bhasin et al, International Journal of Computer Science and Information technologies (IJCSIT)*, Vol. 2 (5), pp. 2412-2415.
[2] Harsh Bhasin, (2011), "Use of Genetic Algorithms for finding Roots of Algebraic Equation", *International Journal of Computer Science and Information technologies* (*IJCSIT*), Vol. 2 (4), pp. 1693-1696.
[3] Harsh Bhasin, Supreet Singh, (2012), "Genetic Algorithms correlation based Rule Generation for Expert System", *Harsh Bhasin et al, International Journal of Computer Science and Information technologies* (*IJCSIT*), Vol. 3 (2), pp. 3733-3736.
[4] Harsh Bhasin, Neha Singla, (2012), "Modified Genetic Algorithms Based Solution To Subset Sum Problem", *International Journal of Advanced Research in Artificial Intelligence* (*IJARAI*) Vol. 1 (1), pp. 38-41.
[5] Harsh Bhasin, Neha Singla, (2012), "Harnessing Cellular Automata and Genetic Algorithms to Solve Travelling Salesman Problem", *International Conference on Information, Computing and Telecommunications (ICICT)*, pp. 72-77.
[6] Sonia Goyat, (2012), "Cryptography using Genetic Algorithms", *IOSR Journal of Computer Engineering (IOSRJCE)*, Vol.1, Issue 5, pp. 06-08.
[7] Harsh Bhasin, Neha Singla, (2012), "Genetic based algorithms for N-Puzzle Problem", *International Journal of Computer Applications (IJCA)*, Vol. 51, No.22, pp. 44-50.
[8] Harsh Bhasin, Rohan Mahajan, (2012), "Genetic Algorithms Based Solution To Maximum Clique Problem", *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 4, No.18, pp. 1443-1448.
[9] Harsh Bhasin and Gitanjali, (2012), "Harnessing Genetic Algorithms for Vertex Cover Problem", *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 1, Issue 2, pp. 218-223.
[10] Harsh Bhasin et al, Regression Testing Using Coupling and Genetic, Algorithms (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (1) , 2012, 3255 – 3259.
[11] Harsh Bhasin, (2012) Cryptography using Genetic Algorithms, Reliability, Infocom, Technology and Optimization Conference.
[12] Keedwell, E. and Khu, S. (2006). "Novel Cellular Automata Approach to Optimal Water Distribution Network Design." J. Comput. Civ. Eng., 20(1), 49–56.