Application of Genetic Algorithm in Communication Network Security

Somalina Chowdhury, Sisir Kumar Das, Annapurna Das
Assistant Professor, Dept. of C.A, Guru Nanak Institute of Technology, Kolkata, India.
Professor and Dean – Research & Administration, Guru Nanak Institute of Technology, Kolkata, India.
Professor & Principal, Guru Nanak Institute of Technology, Kolkata, India.

ABSTRACT: In this paper an Encryption and Decryption algorithm have been designed for communication network using Genetic Algorithm (GA) to make the total cryptography process much faster, robust, and highly secure. Here block cipher and one point crossover is used for simplification. GA is used as a part of Encryption and Decryption technique. It is a complex and highly secured concept to break by intruders, but easy to implement in the design.

KEYWORDS: Encryption, Decryption, Selection, Crossover, Mutation, key, Fitness Value, XOR, Permutation, block cipher.

I. INTRODUCTION

Online transaction is nowadays most common practice in our daily life. So network security has become one of the most concern era. Cryptography is one of the techniques to communicate through insecure network securely. At the same time Genetic Algorithm (GA) is one of the newly emerging problem solving techniques which is gaining popularity due to its robustness. GA offers significant benefits over other optimization techniques in searching a large state space.

Cryptography is the art of processing information secretly. The basic element of cryptography is Algorithm for encryption and decryption of information and the other one is the key used.

Geneticalgorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as mutation, selection, and crossover. Here we have used two point crossover. And the fitness function is used for taking decision of further movement.
II. RELATED WORK

In literature to date, there are many popular encryption technique are present namely DES, RSA etc.[1] And many GA based encryption algorithms have also been proposed. A. Tragha et.al [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]. There are various proposed methods for image encryption such as quad tree approach, cellular automata [4, 5]. There are wide applications of GA in solving non-linear optimization problems in various domains [6]. 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 [7, 9]. Mohammad Sazzadul Hoque et.al [10] 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 [11]. There are several papers related to IDS all of which use GA in deriving classification rules [12, 14].

In this paper we have made an attempt, to build an encryption algorithm inside which we have introduced genetic algorithm. It’s Strength lie in its key size i.e., 64 bit.

III. PROPOSED ALGORITHM

A. Proposed Encryption Algorithm

Step 1
Divide the plain text in 64 bits block each and encrypt each block.

Step 2
Arrange these 64 bits into 4 x 16 table format. For example:

```plaintext
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64
```

Step 3
Four bit shifting per row in a cyclic fashion as highlighted

Step 4
Now divide into two block (32 bits per block). Left part(LP) and Right part(RP).

LP =

RP =

Step 5
Perform Expansion permutation with RP. As shown:
Here some bits from the input are duplicated at the output; e.g. the fifth bit of the input is duplicated in both the sixth and eighth bit of the output. Thus, the 32-bit half-block is expanded to 48 bits.

**Step: 6**
Now generate key of 48 bit (Explained later).

**Step: 7**
Apply XOR with 48 bit key and 48 bit RP to get new RP of 48 bit.

**Step: 8**
Now with the result apply genetic algorithm operations like selection, crossover and mutation.

**Fitness Function:**
From new 48 bit RP make 4 x 12 table. Let the table be:

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
<th>A9</th>
<th>A10</th>
<th>A11</th>
<th>A12</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td>B5</td>
<td>B6</td>
<td>B7</td>
<td>B8</td>
<td>B9</td>
<td>B10</td>
<td>B11</td>
<td>B12</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>C4</td>
<td>C5</td>
<td>C6</td>
<td>C7</td>
<td>C8</td>
<td>C9</td>
<td>C10</td>
<td>C11</td>
<td>C12</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td>D7</td>
<td>D8</td>
<td>D9</td>
<td>D10</td>
<td>D11</td>
<td>D12</td>
<td></td>
</tr>
</tbody>
</table>

Fitness will be calculated for each row of the above table. The fitness function will count the number of ones in each row. If the number of one’s is 6 then fitness is 6 for that row. Similarly calculate for other rows. Then finally count the total fitness. Now perform selection.

**Selection:**
Consider 1st and 3rd row as 1st pair of parent and other as 2nd pair of parent.
Let the new RP of 48 bit in 4 x 12 table as:

Before selection:

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
<th>A9</th>
<th>A10</th>
<th>A11</th>
<th>A12</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td>B5</td>
<td>B6</td>
<td>B7</td>
<td>B8</td>
<td>B9</td>
<td>B10</td>
<td>B11</td>
<td>B12</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>C4</td>
<td>C5</td>
<td>C6</td>
<td>C7</td>
<td>C8</td>
<td>C9</td>
<td>C10</td>
<td>C11</td>
<td>C12</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td>D7</td>
<td>D8</td>
<td>D9</td>
<td>D10</td>
<td>D11</td>
<td>D12</td>
<td></td>
</tr>
</tbody>
</table>

After selection:

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
<th>A7</th>
<th>A8</th>
<th>A9</th>
<th>A10</th>
<th>A11</th>
<th>A12</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td>B5</td>
<td>B6</td>
<td>B7</td>
<td>B8</td>
<td>B9</td>
<td>B10</td>
<td>B11</td>
<td>B12</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>C4</td>
<td>C5</td>
<td>C6</td>
<td>C7</td>
<td>C8</td>
<td>C9</td>
<td>C10</td>
<td>C11</td>
<td>C12</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td>D7</td>
<td>D8</td>
<td>D9</td>
<td>D10</td>
<td>D11</td>
<td>D12</td>
<td></td>
</tr>
</tbody>
</table>

**Crossover (Two point crossover):**
We will randomly choose any two numbers from 1 to 12 as cross over point and implement crossover. Here the points are (3,9).

Crossover for 1st two parents:

P11 -

P12 -

Result or child of crossover:

C11 =

C12 =
Similarly the other two parent pair will also create another two child.

**Mutation:**
We will randomly choose any one numbers from 1 to 12 as mutation point and implement. Here the point is 6(let).

Shown the result after mutation.

Now combine C11, C12, C21 and C22 to get RP.

Now again calculate Fitness same way. If the new fitness is lower then go to next step else again perform mutation, selection, and crossover.

**Step: 9**
Perform XOR operation with LP of step 4 and the result of step 8 which is new RP.

XOR gate work as:

![Key XOR with RP](image)

**Step: 10**
And the RP of step is taken as LP.

**Step: 11**
Now recombine LP of step10 and RP of step 9 to transform in 16 x 4 table.

**Step: 12**
Four bit shifting per column in a cyclic fashion as given below.

Recombine set before shifting

After Shifting the result is

The resultant is the original cipher text.

**B. Proposed Key Generation Technique:**

**Step: 1**
At first random 64 bit are taken. And arrange them into table of 8 x 8 format.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
</tr>
<tr>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
</tr>
<tr>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
</tr>
<tr>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
</tr>
</tbody>
</table>

**Note:**
For 1st block of 64bit plain text the key creation will discard 8th column for next 64 bit it will discard 7th column so as on. And from 9th block again start from 8th column and continue.

**Step: 2**
Now truncate the 8th column of the table, the result will be of 56 bit key.
Step: 3
Arrange 56 bit into table of 8 x 7 format. And truncate mid or 4th row.

Step: 4
Now the key of 48 bit is ready to use. Shown in 4 x 12 format.

C. Implementation in JAVA
We can implement the above algorithm in JAVA to get the result. The files uses for the algorithm are:

<table>
<thead>
<tr>
<th>Filename</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain.txt</td>
<td>Contains the text to be encrypted</td>
</tr>
<tr>
<td>Encrypt.txt</td>
<td>Contains the encrypted data</td>
</tr>
<tr>
<td>Decrypt.txt</td>
<td>Contains the decrypted data</td>
</tr>
<tr>
<td>Key.txt</td>
<td>Contains randomly generated symmetric key</td>
</tr>
</tbody>
</table>
Here in this paper an encryption technique is introduced which will prove to be a strong encryption technique for the unauthorized party to break the code. Here the key needed to share very efficiently as strength lie in key mostly. In near future it can prove to be a popular genetic-encryption technique.

Here the randomness of genetic algorithm and secrecy of cryptography has made this algorithm strong one. And the key transformation process also improved its efficiency. Although many of its steps and key generation techniques are inspired from DES algorithm but flavour of GA makes it unique.

REFERENCES


BIOGRAPHY

Somalina Chowdhury, Obtained BCA and MCA degree in 2008 &2011 respectively, from West Bengal University of Technology, INDIA. She is the member of Computer Society of India. She is presently working as Assistant Professor in GNIT, Kolkata. Her areas of working interest are Genetic algorithm, Network Security, Cryptography and Wireless Sensor Network.

Sisir Kumar Das, Obtained B.Tech, M.Tech and Ph.D degree from Calcutta University, IIT kharagpur and Anna University, respectively in India. He was faculty in Delhi University during 1977-1980. Dr. Das led EMC evaluation and design of electronics products manufactured by the industry meeting International Standards and Electromagnetic Research projects in the country and abroad for 28 years under the ministry of communication and IT, Govt. of India, during 1980-2007. Presently he is Prof. and Dean – Research & Administration, GNIT, Kolkata.He is co-author of Engineering Text Book “Microwave Engineering”, published by Mc-Graw Hill, USA, Singapore and India. He is the author of the text book “Antenna and Propagation”; published by Mc-Graw Hill India. He has written three chapters in the book “Engineering EMC”, Published by IEEE press. He has nearly 120 research publications in journal and conference proceedings.

Copyright to IJIRCCE 10.15680/ijircce.2015.0301057
society of EMC Engineers (India) highest award 2002 in recognition of his contribution to the EMI/EMC Solutions for Indian Industrial Products.

**Annapurna Das** obtained M.Sc. degree in physics from University of Calcutta, M.Tech degree in Microwave Electronics and Ph.D degree in Electrical Engineering from the University of Delhi. She worked in the Department of ECE, Anna University during 1985-2007 as Professor. Presently she is Principal of GNIT, Kolkata. She is the author of Engineering Text Book “Microwave Engineering”, published by Mc-Graw Hill, USA, Singapore and India and co-author of the text book “Antenna and Propagation”, published by Mc-Graw Hill Education. She is the life member of Society of EMC Engineers (India) and ISTE. Her current interests are microwaves, EMI/EMC and Microstrip Antenna.