Security of Data over Internet using Genetic Algorithm

Ghanshyam A. Sharma
Research Scholar, MCA
Thakur Institute of Management Studies, Career Development & Research (TIMSCDR)
Mumbai, India

Pankaj Mudholkar
HoD-MCA
Thakur Institute of Management Studies, Career Development & Research (TIMSCDR)
Mumbai, India

Brijesh Pandey
Thakur Institute of Management Studies, Career Development & Research (TIMSCDR)
Mumbai, India

ABSTRACT
Security of the data which flow over internet it highly insecure because every data is crucial to every user and most important to which data is related. There are already many more encryption and decryption algorithm to protect over data over internet from that I had proposed some new technique of the data protection. Since genetic algorithm nowadays mostly used in building robotic, artificial intelligence to way to think and behave like expert human system to take decision so I think to implement genetic algorithm over data across internet to immerge to era of encryption technique in the term of the security.

Keywords
Genetic Algorithm, Network security, Flip over, Crossover, Mutation, Fitness value.

1. INTRODUCTION
In today world in telecommunication, bandwidth, reliability, performance, cost efficiency and most important key aspect is security. Optical fibers almost have all the capability that above as mentioned but compared to the copper-based wire fiber is best in the wireless communication solution. But the biggest disadvantages of fiber optics its cost. With the advancement in fiber optics technology the cost was reduced drastically and in market there is requirement of fast growing efficient technology. But the security issues is still remain has it is. Security is important concept as point of business perspectives. It is nowadays important to protect data on networks by the network attackers and intruders.

A genetic algorithm is a heuristics that mimics the process of natural evolution. Genetic algorithm is based on the Darwin's theory of evolution; the basic rule is "survival of the fittest". The genetic algorithm is used here to determine the fuzzy membership function. The heuristic is routinely used to generate useful solution to optimization the search problem. Genetic algorithm belongs to the larger class of evolutionary algorithms.

2. GENETIC ALGORITHM IMPLEMENTATION PROCESS
Step1: generate initial population by using random number generation.

Step 2: use the tournament selection method to select any two parents.

Step 3: generate the offspring by using the following arithmetic crossover operator.

$$y1=a*y1+(1-a)*y2$$

$$y2=a*y2+(1-a)*y1$$

Step 4: calculated the maximum fitness values by applying this operation separately for each iteration.

Step 5: To print the output of the function

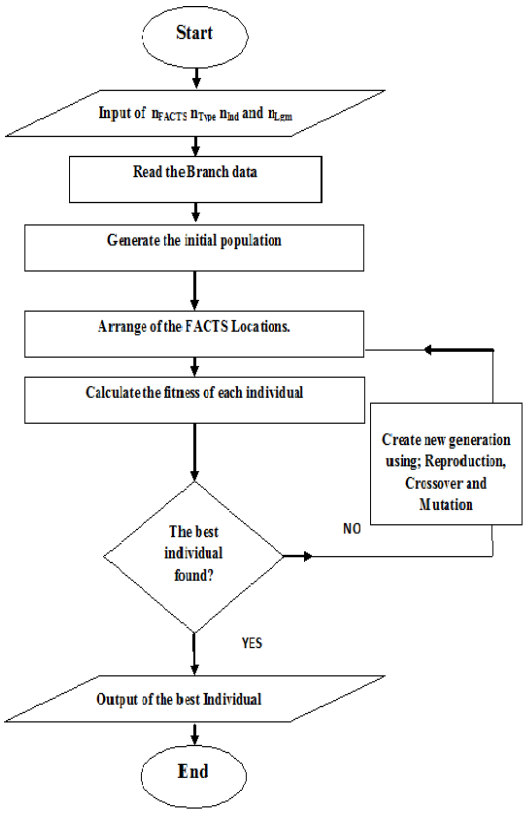


Fig. 1 Flowchart for genetic algorithm

In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next.

Actually crossover is recommendation operator that proceeds in three steps.

1. The reproduction operator elects a random a pair of two individual variables for the mating.
2. A cross site is selected at random along the variable length.
3. Finally, the position values are swapped between the two variable following the cross site.

Below different crossover technique are listed.

Single point crossover: In this type of crossover point of both parent strings are selected. All data beyond that point in either

string is swapped between the two parents. Then the resulting strings are the children.

Two point crossover:-In this type of two crossovers point are chosen and the contents between two mated parents are exchanged between the two mated parents. Thus the contents between these points are interchanged between the parents to produced new child ones for mating in next generation.

The mutation plays the important role of recovering the last genetic materials as well as for randomly distributing genetic information. It is a policy against the irreversible loss of genetic material. Actually there is a random change which tells whether or not and where a particular value will be modified.

These paper presentation is organized to states the security of data that contains confidential information data in order to provide authentication.

3. PROPOSED ALGORITHM

3.1 Encryption Technique using Heuristic Approach with Genetic Algorithm

Input: Accept a plain text from the user.

Output: Generate an encrypted cipher text.

Step 1: Read the input text provided by the user.

Step 2: set a variable where p depends on the position of alphabet.

If position of v=even then v= (pos-1);

If position of v=odd then v= (pos+1);

(Where v is alphabet)

Step 3: The Function is defined as

F(y) = y2-y+1 when 1=y<10[if f(y) =31 then f(y) =f(y) +1];

F(y) = y-1 when y=10;

F(y) =2y+1 when y>10 and odd [if f(y) = 43, then

f(y) = f(y) +2];

F(y)= 3y+2 when y>10 and even

[Note: using these above function we generate unique value for all alphabetic A to Z]

Step 4: Block of character taken as input string that was provided by the user.

Step 5: From a y, (y, y) matrix for strong the storing the values of Add(y), where S = 1, 3,5,7,9.

Step 6: Store the value of f(y) in an array of length "n".

Step 7: Set the value of Add(y)

Such that Add(y)= f(y) + &Add(n)ij = s[i]+f[j];

Where s[] = array for key & f[] = array for f(y);

Step 8: From the table Minimum value of the row[i] is selected. Where I is range from 0 to n .

Step 9: Allocate the value and delete the corresponding row and column.

Swap the key value according to the given sequence.

New key[start]=oldkey[last];

New key[intermediate]=oldkey[unsed];

New key[second last]=oldkey[first];

New key[last]=oldkey[second];

Step 10: Repeat the step 8 and step 9 until the number of allocated cell is equal to the order of the matrix.

Step 11: Allocate the key values are called and its corresponding column value.

Step 12: Print the encrypted data text.

Step 13: G=65;

Step 14: G=M[i] +G-1;

Step 15: M[i] is encrypted by the character of corresponding ASCII value "G".

Step 16: If "i" is less than or equal to n then go to the step 14 and 15 else go to step 17.

Step17: Print the encrypted value.

Step 18: Initialize population size =5, Maximum no of generation=100, Crossover probability=0.99.

Step 19: Input encrypted string are placed in to a square matrix whose size is next intermediate square value of the string length.

Step 20: Then calculated in each row [i] how many times a values is repeated such di.

Where "i" is range from 0 to n;

Step 21: Calculated the sum of the entire di such that

Sum (di) = d1+d2+d3+d4+…. +dn.

Step 22: Fitness of the value f1=sum(di).

Step 23: Apply the crossover technique such that to remove duplication of value in each row.

Step 24: Again apply step 20, step 21 and step 22 and calculated the new fitness value f2

Step 25: if (f1< =f2) then

{

Select f1 and print the encrypted text.

}

Else

{

Select f2 and print the encrypted text.

}

Step 26: Pass the encrypted cipher text on the network.

Step 27: Stop:

3.2 Decryption Technique using Heuristic Approach with Genetic Algorithm

Input: An encrypted cipher text accepted as an input.

Output: Generated Plain text as an output.

Step 1: Read the Input Encrypted cipher text.

Step 2: Then Allocated the string in the matrix S based on the population.

Step 3: Apply the reverse crossover technique that we had applied in the encryption: Such that repetition of string will occur.

Step 4: Calculated the fitness value such that

If (f1>=f2) then

{

Select f1 and print the decrypted text.

}

Else

{

Select f2 and print the decrypted text

}

Step 5: Taking four characters sequentially at a time from the decrypted text.

Step 6: Match each group of characters with their immediate allocated values which was selected at the time off encryption in the 1st encryption matrix.

Step 7: subtract the character value from each group with their corresponding allocated cell value. Let it be x.

Step 8: Get the corresponding character for each x.

Step 9: Now we have to match the column sequence with the four character of each group from step 8.

Step 10: Sort the column number in linear order and then arrange the character that we associate with the column number accordingly.

Then the finally the resultant string set will be the final original plain text.

Step 11: End.

4. ILLUSTRATION OF PROPOSED ALGORITHM WITH THE HELP OF EXAMPLE

Working example of security of data over network

Initial position	1	2	3	4	5	6	7	8	9
Alphabetic	A	B	C	D	E	F	G	H	I
Value	2	1	4	3	6	5	8	9	10
Initial position	10	11	12	13	14	15	16	17	18
Alphabetic	J	K	L	M	N	O	P	Q	R
Value	9	12	11	14	13	16	15	18	17
Initial position	19	20	21	22	23	24	25	26	
Alphabetic	S	T	U	V	W	X	Y	Z	
Value	20	19	22	21	24	23	26	25	

Applied function on the given alphabet to generate unique value for each.

- F(y) = y2-y+1 when 1=y<10[if f(y) =31 then f(y) =f(y) +1];
- F(y) = y-1 when y=10;
- F(y) =2y+1 when y>10 and odd [if f(y) = 43, then f(y) = f(y) +2];
- F(y)= 3y+2 when y>10 and even

F(Y)	F(A)	F(B)	F(C)	F(D)	F(E)	F(F)	F(G)	F(H)	F(I)
Val	3	1	13	7	32	21	57	43	9
F(Y)	F(J)	F(K)	F(L)	F(M)	F(N)	F(O)	F(P)	F(Q)	F(R)
Val	73	38	23	44	27	50	31	56	35
F(Y)	F(S)	F(T)	F(U)	F(V)	F(W)	F(X)	F(Y)	F(Z)	Space @
Val	62	39	68	45	74	70	80	51	2

Key Value: - 1, 3, 5, 7, 9

User entered plain text: - HELLO WORLD

Array: space is denoted by @

We will divide the user enter text into group of four alphabet

So the text will be look like this:

HELL O@WO RLD@

1) HELL

	43	32	23	23
1	44	33	24	24
3	46	35	26	26
5	48	37	28	28
7	50	39	30	30

Cipher text:-7531

2) O@WO

	50	2	74	50
7	57	9	81	57
9	59	11	83	59
1	51	3	75	51
3	53	5	77	53

Cipher text: - 3917

3) RLD@

	35	**23**	**7**	**2**
3	38	26	10	5
5	40	28	12	7
7	42	30	14	9
9	44	32	16	11

Cipher text: - 9753

Complete cipher text is: - 753139179573

Encrypted using ASCII values:-

- 1= (65 + 1)-1=65=A
- 3= (65+ 3)-1=67=C
- 5= (67 + 5)-1=71=G
- 7= (71+7)-1=76=M
- 9= (76+9)-1=85=U

Final cipher text:-MGCA CUAM UMGC

Encryption using heuristic approaches with genetic algorithm

1	M	G	C	A	C	2
2	U	A	M	U	M	2
3	G	C				3
4						5
5						5

F1= 2+2+3+5+5=17;

Applying crossover:

1	M	G			C	2
2	U	A			M	2
3	G	C				3
4			C	A		3
5			M	U		3

F2=2+2+3+3+3=13;

According the rule:

if (f1< =f2) then

{

Select f1 and print the encrypted text.

}

Else

{

Select f2 and print the encrypted text.

}

After the crossover:

Final ENCRYPTED cipher text is:

MG@@CUA@@MGC@@@@@CA@@@MU@

4.1 Decryption Using Genetic Algorithm

Input text: MG@@C UA@@ MGC@ @@@@ CA@@ @MU@

1	M	G			C	2
2	U	A			M	2
3	G	C				3
4			C	A		3
5			M	U		3

F1=2+2+3+3+3=13;

Applying crossover:

1	M	G	C	A	C	2
2	U	A	M	U	M	2
3	G	C				3
4						5
5						5

F2= 2+2+3+5+5=17;

According the rule:

If (f1>=f2) then

{

Select f1 and print the decrypted text.

}

Else

{

Select f2 and print the decrypted text

}

Initial decrypted value is:

MGCA CUAM UMGC

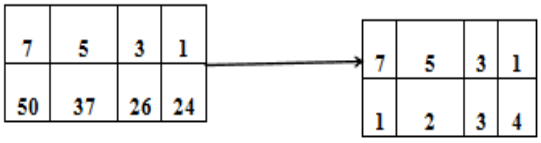
Decrypted using ASCII values:-

- A=(65+1)-65=1
- C=(67+1)-65=3
- G=(71+1)-67=5
- M=(77+1)-71=7
- U=(85+1)-77=9

Decrypted new value is:

7531 3917 9753

1) 7531



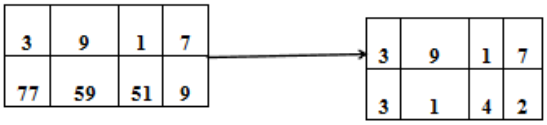
50-7=43=H position:

37-5=32=E HELL

26-3=23=L

24-1=23=L

2) 3917



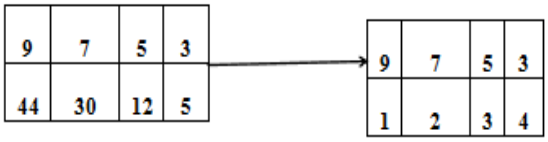
77-3=74=W position:

59-9=50=O O@WO

51-1=50=O

9-7=2=@

3) 9753



44-9=35=R position:

30-7=23=L RLO@

12-5=7=O

5-3=2=@

So the text is: HELLO@WORLD@

After replacing @ with space we get final plain text as mentioned bellowed

"HELLO WORLD"

5. CONCLUSION

The research paper is concerned about sending a message securely to the receiver so that no third party or intruder. Who has the cipher text but can't able to understand what is actually a plain text. It also provides facilities to the sender to enter the complete message and encode using crossover and heuristic technique with genetic algorithm. So the only genuine user can decode the message and get the original plain text by using genetic algorithm. The heuristic technique provides not only optimum solution but also good solution. And genetic algorithm works on the method that today solution of the problem may be not the solution of the problem in future. So in genetic algorithm its keeps on changing based on the best fitness value test.

6. REFERENCES

[1] Principle of soft computing author of the book is S.N. Sivanandam and S.N. Deepa under publication of Willey India Pvt. LTD ISDN-978-81-265-2741-0.

[2] http://scialert.net/fulltext/?doi=ajsr.2008.403.411

[3] Principle of soft computing author of the book is S.N. Sivanandam and S.N. Deepa under publication of Willey India Pvt. LTD ISDN-978-81-265-2741-0.