DESIGN AND IMPLEMENTATION OF NETWORK SECURITY USING GENETIC ALGORITHM

Soumya Paul[1], Inadyuti Dutt[2], S.N. Chaudhuri[3]

[1]Associate Professor & Head, [2]Asst. Professor, Dept. of Computer Application, B.P. Poddar Institute of Management & Technology, West Bengal, India, [3]Director, Kanad Institute of Engineering & Management, West Bengal, India,
soumya.paul2000@gmail.com**, **inadyuti&gmail.com, satya.chaudhuri@rediffmail.com

Abstract

Over the last few years, Secured transmission of data has been a major issue in data communication. This project mainly concerns about the security of confidential information and data transmission using public key cryptography with Genetic Algorithm in order to provide confidentiality, authentication, integrity and non-repudiation of the messages. First, an algorithm is developed and implemented to generate a key pair (Private and public Key).A plain text is encrypted using the Public Key of receiver to produce an intermediate cipher. The intermediate cipher is again encrypted using genetic algorithm to produce final cipher. The final cipher first decrypted to produce the intermediate cipher which in turn decrypted to get the plain text using the Private key of the receiver or vice versa.

***Index Terms**: Network Security, Genetic Algorithm*

----------***----------

1. INTRODUCTION

1.1 Network Security:

Network security consists of the provisions and policies to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned a password or other authenticating information that allows them access to information and programs within their authority. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals.

1.2 Genetic Algorithm:

Genetic algorithms are inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved.

Algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce.

This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.

1.3 Outline of Remaining Section:

The paper is organized as follows Section 2 states the security of files that contain certain confidential information. The detail description of the proposed encryption decryption algorithms with illustration of examples are explained in section 3. Finally, the paper concludes in section 4.

2. PROBLEM STATEMENT

Encryption method follows as: A pair of key (Public and Private) generated. In Asymmetric-key cryptography, it is obvious that when a text is encrypted with a public key, it is decrypted with corresponding private key and vice-versa. The plain text is encrypted using public key to produce Intermediate cipher. The Intermediate cipher is further encrypted using Genetic Algorithm to produce the final cipher. Decryption method: The Final cipher is decrypted using Genetic Algorithm to get intermediate cipher which is again decrypted using corresponding private key to get the plain text.

Initial Encryption:
Intermediate cipher = Encrypt (plaintext, public Key) OR
Intermediate cipher = Encrypt (plaintext, Private Key)

Genetic Encryption:
Final cipher = Encrypt (Intermediate cipher)

Genetic Decryption:
Intermediate cipher = decrypt (Final cipher) Final Decryption:
Plaintext = decrypt (Intermediate cipher, Private Key) OR
Plaintext = decrypt (Intermediate cipher, Public Key)

3. PROPOSED ALGORITHM

3.1 Proposed Key Generation Heuristic:

Step 1: Generate two non coprime number (a, b)
Step 2: Set l: =L.C.M (a, b) g: = G.C.D (a, b)
Step 3: Set (a, g) as private key
Step 4: Set x: = ((a-1)/g) + ((a-1) %g)

y: =g

Step 5: Store x, y
Step 6: Set (b, l) as public key
Step 7: Set p: = ((l-1)/b + ((l-1) %b)
q: =b Step 8: Set p, q Step 9: Stop

3.2.1 Proposed Encryption Heuristic Using

Function:-

Input : text, key type, public key (p, q) or private key (x, y)

Output : Intermediate Cipher
Step 1: if key type= private key
Read public key (p, q) from database
Set m: = 2 else
Read private key (x, y) from database
Set m: = 1
Step 2: Set x: = ((x-(y-1))*y) +y
Step 3: Set p: = ((p-(q-1))*q) +q
Step 4: if (x/y=p/q) and (x*q=p*y) then
Set n: = x*q
Step 5: Set key_arr []: =n
Step 6: if (length of text = odd number) Then text: =text + @
Step 7: Set i: =0, c: =" ", j: =0
Step 8: while (i<length of text) Repeat Step 9 to Step 12
Step 9: if i= EVEN number
Set c: =c + text [i] + (m* key_arr[j]) Else
Step 10: Set c: = c+text[i] - (m* key_arr[j]) End if
Step 11: Set j: = j+1
Step 12: if j=key_length then set j: =0
Step 13: Print c
Step 14: Stop

3.2.2 Proposed Encryption Heuristic Using

Genetic Algorithm:-

Input : Intermediate Cipher(c)

Output : Final Cipher
Step 1: Set st1:= substring of c (1 to c/2)
step2:= substring of c (c/2 to c) Step 2: Set i=1
Step 3: while (i<length of text) Repeat Step 4 to Step 8
Step 4: perform crossover at mate point i with st1 (0, i) & st2 (i, n)
& st2 (0, i) & st1 (i, n)
Step 5: Set st: = st1+Reverse (st2)
Step 6: Reverse st & set gst= st
Step 7: compare c & gst to find out fit value (no of character position remain unchanged)
Step 8: select the gst having minimum no of fit value
Step 9: Return gst
Step 10: Stop

3.3 Proposed Decryption heuristic using

Genetic Algorithm:-

Input : Final Cipher(c)

Output : Intermediate Cipher

Step 1: Set st: = reverse (cipher)
Step 2: Set n=length of cipher
Step 3: Set st1:= substring of c (1 to n/2)
st2:= substring of c (n/2 to n)
Step 4: Set st2: =Reverse of st2
i:=1
Step 5: while (i<length of st2) Repeat Step 6 to Step 9
Step 6: perform crossover at mate point i
with st1 (0, i) & st2 (i, n)
and st2 (0, i) & st1 (i, n)
Step 7: Set gst: =st1+st2
Step 8: compare gst & cipher to find out fit value
(no of character position
Remain unchanged)
Step 9: select the gst having minimum no of fit value
Step 10: Return gst
Step 11: Stop

3.4 Proposed Decryption Heuristic using

Function:-

Input : Intermediate cipher, keytype, public key
(p, q) or private key (x, y)

Output : Plain Text
Step 1: if key type= public key
Set m: = 2 & Read private key (x, y) from database

else
Set m: = 1 & Read public key (p,q) from database

Step 2: Set x: = ((x-(y-1))*y) +y
Step 3: Set p: = ((p-(q-1))*q) +q

Step 4: if (x/y=p/q) and (x*q=p*y) then
Set n: = x*q
Step 5: Set key_arr [] : =n
Step 6: Set i=0, c=" ", j=0
Step 7: while (i<length of plmcipher) Repeat Step 4 to Step 7
Step 8: if i= EVEN number then
Set c: =c + plmcipher[i] - (m* key[j]) Else
Step 9: Set c: = c+plmsiphert[i]+(m* key[j]) End if
Step 10: Set j: = j+1
Step 11: if j=length of key [] then Set j: =0
Step 12: Print c

3.5 Proposed Cryptographic Algorithm:

Step 1: Start
Step 2: Call Proposed Encryption Heuristic
Step 3: Call Proposed Encryption Heuristic using Genetic Algorithm
Step 4: Call Proposed Encryption Heuristic using Genetic Algorihm
Step 5: Call Proposed Decryption Heuristic
Step 6: Stop

3.6 Illustration of Proposed Algorithm with Example:

:

A. Key Generation

Two Non-Co prime number generated: a=39, b=453
l:=L.C.M (a, b)=L.C.M(39,453)=5889 g:= G.C.D (a, b)=G.C.D(39,453)=3
x: = ((a-1)/g) + ((a-1) %g)
=((39-1)/3)+((39-1)%3)=14 y: =g=3
p: = ((l-1)/b + ((l-1) %b)
=((5889-1)/453+((5889-1)%453)=464 q: =b=453
Publish (x,y) as private key and (p,q) and public key

B. Encryption

STEP 1: Encryption using Public Key:

Input Public key(p,q)=(464,453) Input plain text: "Good Morning"
Read Private Key(x,y)=(14,3) from Database x: = ((x-(y-1))*y) +y=((14-(3-1))*3)+3=39
p: = ((p-(q-1))*q) +q
=((464-(453-1))*453)+453=5889 (x/y=p/q=13) and (x*q=p*y=17667)
n: = x*q=39*453=17667 key_arr[]={7,6,6,7,1} m:=1 and c="" when i=1 ie, ODD
c: =c + ASCII(text [i])+ (m* key_arr[j])
=ASCII('G')+(1*7)=N
when i=2 ie, EVEN
c: = c+ ASCII(text[i])- (m* key_arr[j])
= N+(ASCII('o')-(1*6))=Ni

Finally we get the intermediate cipher :
Niu]!Fuluhua

STEP 3: Encryption using Genetic Algorithm

Input: Intermediate Cipher (st): Niu]!Fuluhua n:=length of st
st1:=substring(0,n/2)= Niu]!F

st2:=substring(n/2,n)= uluhua

Iteration 1: Cross Over

N	i	u	h	u	a
u	l	u	]	!	F

Mutation: NluhuaF!]uiu

Inversion(gst): u i u] ! F a u h u l N (st)
:N i u] ! F u l u h u a

Fit value: Comparing st and gst to find out no. of characters whose position remained unchanged.
fit(st,gst)=5

Iteration 2: Cross Over

N	I	u	]	!	F
u	l	u	h	u	a

=

N	l	u	h	u	a
u	i	u	]	!	F

:

Mutation: NiuhuaF!]ulu

Inversion(gst): u l u] ! F a u h u i N (st)
:N i u] ! F u l u h u a

Fit value: Comparing st and gst to find out no. of characters whose position remained unchanged.
fit(st,gst)=4

Iteration 3: Cross Over

N	i	U	]	!	F
u	l	U	h	u	a

=

N	i	u	h	u	a
u	l	u	]	!	F

Mutation: NiuhuaF!]ulu

Inversion(gst): u l u] ! F a u h u i N (st): N i u] ! F u l u h u a

Fit value: Comparing st and gst to find out no. of characters whose position remained unchanged.

fit(st,gst)=4

Iteration 4: Cross Over

N i u] ! F
u l u h u a

=

N	i	u	]	!	a
u	l	u	h	u	F

Fit value: Comparing st and gst to find out no. of characters whose position remained unchanged. fit(st,gst)=2

So, fittest gst having minimum fit value is : u l u h u
F a !] u i N

C. Decryption

STEP 1: Decryption using Genetic Algorithm

Input: Final Cipher (st):
u l u h u F a !] u i N Inversion: N i u] ! a F u h u l u
Mutation(mst): N i u] ! a u l u h u F
n:=length of st st1:=substring(0,n/2)= N i u] ! a
st2:=substring(n/2,n)= u l u h u F
Fit value: Comparing st and gst to find out no. of characters whose position remained unchanged. fit(st,gst)=4
Iteration 1: Crossover

N i u] ! a
u l u h u F

=

N	i	u	]	u	a
u	l	u	h	!	F

Mutation: Niu]uaF!hulu

Inversion(gst): u l u h ! F a u] u i N
(st):N i u] ! F u l u h u a

Fit value: Comparing st and gst to find out no. of characters whose position remained unchanged. fit(st,gst)=3

Iteration 5: Cross Over

gst= N l u h u F u i u] ! a
st= u l u h u F a !] u i N

Fit value: Comparing st and gst to find out no. of characters whose position remained unchanged. fit(st,gst)=5

Iteration 2: Crossover

N	i	u	]	!	a
u	l	u	h	u	F

gst= N i u h u F u l u] ! a st= u l u h u F a !] u i N

Mutation: Niu]!aFuhulu

Inversion(gst): u l u h u F a !] u i N (st):N i u] ! F u l u h u a

Iteration 3: Crossover

=

N	i	u	]	!	a
u	l	u	h	u	F

N	I	u	h	u	F
U	L	u	]	!	a

gst= N i u h u F u l u] ! a
st= u l u h u F a !] u i N

Fit value: Comparing st and gst to find out
no. of characters whose position remained unchanged. fit(st,gst)=4

Iteration 4: Crossover

=

N	i	u	]	!	a
U	l	u	h	u	F

N i u] u F
u l u h ! a

N	l	u	h	u	F
u	i	u	]	!	a

gst= N i u] u F u l u
h ! a
when i=1 ie, ODD
c: =c + ASCII(text [i])- (m* key_arr[j])=ASCII('N')-(1*7)=G
when i=2 ie, EVEN
c: = c+text[i]+ (m* key_arr[j])= N+(ASCII('i')+(1*6))= Go

st= u l u h u F a !] u i N

Fit value: Comparing st and gst to find out no. of characters whose position remained unchanged.
fit(st,gst)=3

Iteration 5: Crossover
=

N	i	u	]	!	a
u	l	u	h	u	F

N	i	u	]	!	F
u	l	u	h	u	a

gst=

N i u] ! F u l u h u a
st= u l u h u F a !] u i N

Fit value: Comparing st and gst to find out no. of characters whose position remained unchanged.
fit(st,gst)=2

So fittest string(fst) having minimum fit value(2) = N
i u] ! F u l u h u a

STEP 2: Decryption using Private Key: Input Private Key(x,y)=(14,3)
Input intermediate cipher: Niu]!Fuluhua
Read Public key(p,q)=(464,453)
from Database
x: = ((x-(y-1))*y) +y
=((14-(3-1))*3)+3=39 p: = ((p-(q-1))*q) +q
=((464-(453-1))*453)+453=5889

(x/y=p/q=13) and (x*q=p*y=17667)

n: = x*q=39*453=17667 key_arr[]={7,6,6,7,1} m:=1 and c=""
Finally we get the intermediate
cipher : Good Morning

CONCLUSIONS

In this work, a heuristic for network security in using genetic algorithm in computer networks has been proposed. Here an asymmetric cryptographic approach is implemented to ensure confidentiality in networks, which is again designed and implemented with genetic algorithm.

ACKNOWLEDGEMENTS

The authors of this research would like to thank B. P. Poddar Institute of Management and Technology for providing high-end computing laboratories during their research work.

REFERENCES

[1] Kartalopoulos, S.V. "Differentiating Data Security and Network Security", IEEE Communications, 2008. ICC'08. International Conference on Telecommun. Networking, Univ. of Oklahoma, Tulsa, OK, pp. 1469 – 1473, Issue Date: 19-23, May 2008.
[2] Stamatios V. Kartalopoulos, Williams Professor in Telecommunications Networking, "Optical Network Security: Countermeasures in View of Channel Attacks", Military Communications Conference,
2006. MILCOM 2006. IEEE , The University of Oklahoma, Washington, DC, ISBN: 1-4244-0617-X, pp. 1-5, Issue Date: 23-25 Oct, 2006.
[3] Vulnerabilities and Security strategy for the Next Generation bandwidth Elastic Pon Stamatios V.Kartalopous,Di Jin Ece department,TCOM Graduate program , The University of Oklahoma , 4502 E.41 st Street,Tulsa,OK 74135 USA.
[4] Cryptography and Network Security , by Forouzon.
[5] An Introduction to Neural Network ,Kevin Gurney.

BIOGRAPHIES

Soumya Paul, Assoc. Professor and Head, Department of Computer Application in B. P. Poddar Institute of Management & Technology, Kolkata, has been in teaching and research for over 12 years. He holds a Master's Degree in Technology, Computer Application as well as in Mathematics and has gathered vast experiences in the same. He received his M.Sc. (Mathematics) from Visva Bharati University and stood 1st class 1st. He received MCA from National Institute of Technology, Rourkella and M. Tech (CSE) from AAI-Deemed University and pursuing Ph. D in Computer Science and Engineering. He served as a faculty member and visiting faculty member in various Institutes and Universities like RCCIIT, Visva Bharati University, University of Calcutta, Bardhaman University, West Bengal University of Technology etc. He has delivered numerous lectures across India in the field of his research interest, Optical Networks and Genetic Algorithms. He is an author/co-author of several published articles in International Journals and International Conferences. He has chaired an International Conference technically supported by IEEE communication. He has more than 15 research publications and currently Reviewer and Member, Editorial Board in many conferences and journals like International Journal of Data Modelling and Knowledge Management.

Inadyuti Dutt has been in the field of academics for more than ten years and currently the Assistant Professor of Dept. of Computer Application, B.P.Poddar Institute of Management & Technology, West Bengal, India. Earlier she held various technical positions in National Informatics Centre, Kolkata , Semaphore Computing Network etc. She has earned Master's degree in Computer Application and currently pursuing her research in Computer Science & Engineering. She has more than 20 publications to her laurels and her research interest is specifically in the field of Optical Networking, Security and Genetic Algorithms. She has also been Member, Editorial Board in journal publications

S.N. Chaudhuri, Director Kanad Institute of Engineering & Management, Mankar, Burdwan, West Bengal is a renowned Academician as well as a world famous Scientist. He has working experience for nearly 40 years in different National and International Institutions. As visiting Professor, he visited different foreign Universities. He is having a number of publications to his credit and his name has been included in who is Who Indian Personages.