Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 6, June 2015, pg.815 – 821

RESEARCH ARTICLE

Implementation of Network Security Using Neural Network Architecture

Dr. Soumya Paul[1], Dr. Sudipto Bhattacharyya[2]

[1]Associate Professor and HOD, Department of Computer Applications, BPPIMT, WBUT, India

[2]Associate Professor and HOD, Department of IT, ILEAD, WBUT, India

[1] soumya.paul2000@gmail.com; [2] sdipto@gmail.com

Abstract*— This paper concerns about the security of confidential information and data transmission through Neural Network using symmetric key in order to provide confidentiality, authentication, integrity and non-repudiation of the messages. First, an encryption algorithm is developed and implemented to achieve the aforesaid purpose. It is basically a program that takes any number and a private key as input from the user and produces a cipher text which is sent to the source node of the neural network. Now the cipher text is again encrypted and fragmented in successive layers of neural network architecture to produce the final cipher text. Finally all the fragments are collected and decrypted using the same private key at destination node to get the cipher text which was input to the source node. Then the cipher text is decrypted to get the original text at the other end of the network.

***Keywords*— Symmetric-Key, Artificial Neural Network, Hidden Layer, Network Security, Cipher Text**

I. INTRODUCTION

The main focus of this paper is to develop and strengthen the techniques in order to achieve a point where the intruder fails to fulfill his or her objective which paves a way full of integrity, availability and confidential message transmission. Our three goals of security (integrity, availability and confidentiality) can be threatened by security attacks. 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. The actual implementation of security goals needs some techniques. One of the important techniques is cryptography. Although in past cryptography referred only to the encryption and decryption of messages using secret keys, today it is defined as involving two distinct mechanism: symmetric-key encryption & asymmetric-key encryption. Symmetric-Key encipherment uses a single secret key for both encryption & decryption. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an Artificial Neural Network is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data.

II. **FLOW DIAGRAM**

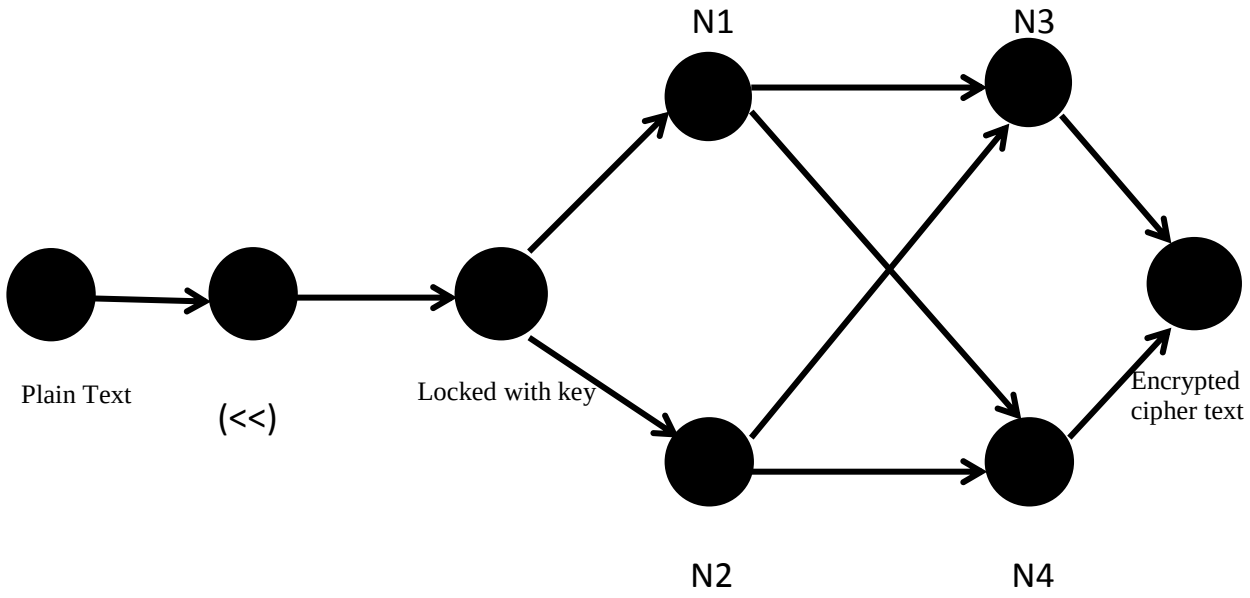

Fig. 1 Encryption Flow diagram

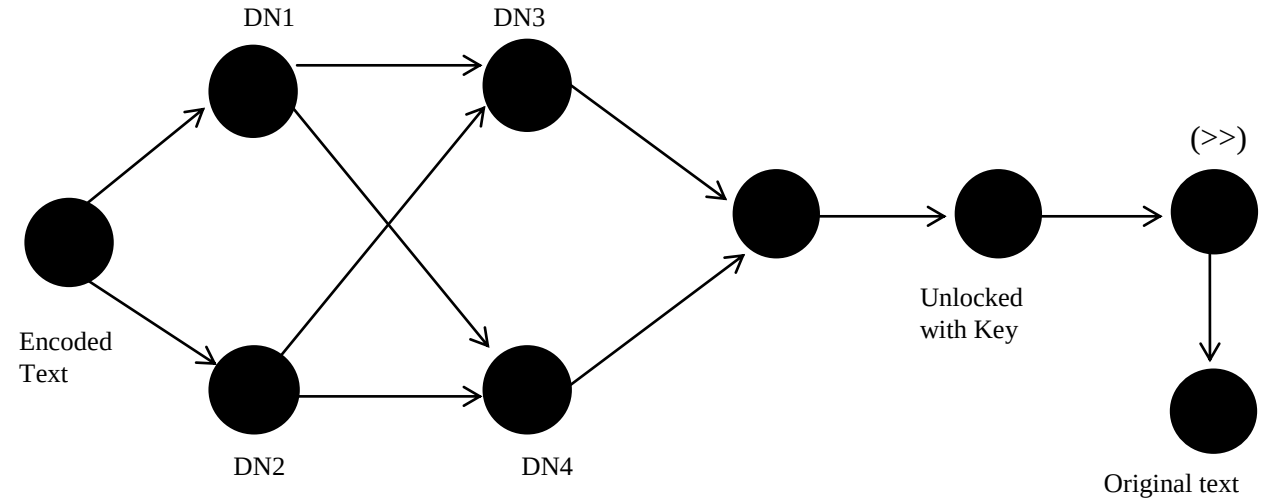

Fig. 2 Decryption Flow diagram

III. **ALGORITHM**

A. Proposed Encryption – Decryption Heuristic

Input: A plain text and key is taken as input.
Process: Plain text is converted into a cipher text using a key and transmitted over network.
Output: Encoded cipher text is decoded to get back original plain text at the receiving end of the network.

Step 1: Call secret key generation algorithm.
Step 2: Input a plain text for encryption

Step 3: Call encryption algorithm using Neural Network Architecture.
Step 4: Call decryption algorithm using Neural Network Architecture (output).
Step 5: Stop.

1) Algorithm for encoded secret key generation:

Step 1: Take a 4x4 matrix with the numbers from 0-9.
Step 2: Take a numeric secret key (fixed).
Step 3: Interchange the row and column positions of the secret key values.
Step 4: Transpose the positional matrix which we got after step 3.
Step 5: Stop

2) Algorithm for input matrix to be encrypted:

Step 1: Place the input values sequentially in the 4x4 matrix in row-major order.
Step 2: Operate left shift (<<) operation on the ASCII values of the elements of the input matrix.
Step 3: Subtract the encoded secret key form the left shifted matrix got from step 2.
Step 4: Convert the 4X4 matrix into a single dimension array by row-major order.
Step 5: Split the array equally into two single dimensional arrays (called nodes) containing 8 elements each according to their even and odd positions. Send the even positional elements to N1 and the odd positional elements to N2.
Step 6: Take a single dimensional array of prime numbers (starts from 2 and increasing sequentially) of equal size of the two halves got from Step 5.
Step 7: Add the elements of the prime number array with the corresponding elements of N1.
Step 8: Subtract the elements of the prime number array with the corresponding elements of N2.
Step 9: Send the even positional elements of N1 & N2 to N3 alternatively from N1 & N2.
Step 10: Send the odd positional elements of N1 & N2 to N4 alternatively from N1 & N2.
Step 11: Concatenate the elements of N3 & N4 in a 1-D array.
Step 12: Convert the 1-D array into a 2-D 4x4 matrix.
Step 13: Stop

3) Decryption algorithm:

Step 1: Convert the 2-D 4x4 cipher text matrix received in a single 1-D array in row-major order.
Step 2: Break the 1-D array into 2 equal halves.
Step 3: Put the values of 1st half to even cell positions of DN1 & DN2 alternatively.
Step 4: Put the values of 2nd half to odd cell positions of DN1 & DN2 alternatively.
Step 5: Take a single dimensional array of prime numbers (starts from 2 and increasing sequentially) of equal size of the two halves got from Step 1.
Step 6: Subtract the elements of the prime number array with the corresponding elements of DN1.
Step 7: Add the elements of the prime number array with the corresponding elements of DN2.
Step 8: Merge the values of DN1 & DN2 alternatively in a 1-D array.
Step 9: Convert the 1-D array into a 4X4 matrix.
Step 10: Add the encoded secret key with the matrix got from Step 9.
Step 11: Operate right shift (>>) operation on the ASCII values of the elements of the resultant matrix.
Step 12: Extract the input string whenever the special character is found.
Step 13: Stop.

IV. **EXAMPLE ILLUSTRATION**

A. Example Illustrating Encryption Algorithm

Initial matrix is taken like

	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	?	?
3	?	?	?	?

Input the SECRET KEY as 786

Step 1: Search the elements of the secret key in the matrix. If found swap by interchanging the row & column positions

	0	1	2	3
0	0	1	8	3
1	4	5	9	?
2	2	6	?	?
3	?	7	?	?

We found 7 at (1, 3). So we swapped with? (3, 1). Similarly all other elements in the secret key are also swapped to get the positional matrix.

Step 2: Transpose the positional matrix.

	0	1	2	3
0	0	4	2	?
1	1	5	6	7
2	8	9	?	?
3	3	?	?	?

Step 3: Find the corresponding ASCII values to get the key matrix.

	0	1	2	3
0	48	52	50	63
1	49	53	54	55
2	56	57	63	63
3	51	63	63	63

Step 4: Input the plain text i.e. 631257

Step 5: Converting it into 4x4 matrix & replacing white spaces by "?"

	0	1	2	3
0	6	3	1	2
1	5	7	?	?
2	?	?	?	?
3	?	?	?	?

Corresponding ASCII value matrix is

	0	1	2	3
0	54	51	49	50
1	53	55	63	63
2	63	63	63	63
3	63	63	63	63

Step 6: Left shift operation (<<) on the matrix by 1

	0	1	2	3
0	108	102	98	100
1	106	110	126	126
2	126	126	126	126
3	126	126	126	126

Step 7: Subtract the encoded key matrix from the resultant matrix

	0	1	2	3
0	60	50	48	37
1	57	57	72	71
2	70	69	63	63
3	75	63	63	63

Step 8: Corresponding character matrix

	0	1	2	3
0	<	2	0	%
1	9	9	H	G
2	F	E	?	?
3	K	?	?	?

Step 9: Converting the matrix elements into a 1-D array

<	2	0	%	9	9	H	G	F	E	?	?	K	?	?	?

Step 10: Placing the even positional values at N1 & the odd positional values at N2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<	2	0	%	9	9	H	G	F	E	?	?	K	?	?	?

N1

<	0	9	H	F	?	K	?

N2

2	%	9	G	E	?	?	?

ASCII values of the characters at N1

60	48	57	72	70	63	75	63

ASCII values of the characters at N2

50	37	57	71	69	63	63	63

Step 11: Taking a 1D array of prime numbers starts from '2' & increasing sequentially& of equal size of the arrays of N1 and N2

2	3	5	7	11	13	17	19

Step 12: Adding the array of prime numbers with the elements of N1 & subtracting the same from the elements of N2

N1

62	51	62	79	81	76	92	82

N2

48	34	52	64	58	50	46	44

Step 13: Placing the even positional elements of N1 & N2 to N3 alternatively from N1 & N2

N3

62	48	62	52	81	58	92	46

Step 14: Placing the odd positional elements of N1 & N2 to N4 alternatively from N1 & N2

N4

51	34	79	64	76	50	82	44

Step 15: Concatenating the elements of N3 & N4 in a 1-D array.

62	48	62	52	81	58	92	46	51	34	79	64	76	50	82	44

Corresponding character array

>	0	>	4	Q	:	\	.	3	"	O	@	L	2	R	,

The encrypted final cipher text is

	0	1	2	3
0	>	0	>	4
1	Q	:	\	.
2	3	"	O	@
3	L	2	R	,

B. Example Illustrating Decryption Algorithm

The encrypted cipher text received is converted in a 1-D array

>	0	>	4	Q	:	\	.	3	“	O	@	L	2	R	,

Step 1: Breaking the cipher text into 2 equal halves.

DN1

>	0	>	4	Q	:	\	.

DN2

3	“	O	@	L	2	R	,

ASCII values of the characters at DN1

0	1	2	3	4	5	6	7
62	48	62	52	81	58	92	46

ASCII values of the characters at DN2

0	1	2	3	4	5	6	7
51	34	79	64	76	50	82	44

Step 2: Placing the values of 1st half to even cell positions of DN1 & DN2 alternatively at DN3 & the values of 2nd half to odd cell positions of DN1 & DN2 alternatively at DN4

DN3

0	1	2	3	4	5	6	7
62	51	62	79	81	76	92	82

DN4

0	1	2	3	4	5	6	7
48	34	52	64	58	50	46	44

Step 3: Taking a 1-D array of prime numbers starts from ‘2’ & increasing sequentially & of equal size of the arrays of DN1 & DN2.

2	3	5	7	11	13	17	19

Step 4: Subtracting the array of prime numbers from the array elements of DN3 & adding the same with the array elements of DN4.

DN3

60	48	57	72	70	63	75	63

DN4

50	37	57	71	69	63	63	63

Step 5: Merging the values of DN3 & DN4 alternatively in a 1-D array

60	50	48	37	57	57	72	71	70	69	63	63	75	63	63	63

Step 6: Converting the array into a 4X4 matrix

	0	1	2	3
0	60	50	48	37
1	57	57	72	71
2	70	69	63	63
3	75	63	63	63

Step 7: Adding the encoded secret key with the matrix

	0	1	2	3
0	108	102	98	100
1	106	110	126	126
2	126	126	126	126
3	126	126	126	126

Step 8: Right shift operation (>>) on the matrix by 1

	0	1	2	3
0	54	51	49	50
1	53	55	63	63
2	63	63	63	63
3	63	63	63	63

Corresponding character matrix

	0	1	2	3
0	6	3	1	2
1	5	7	?	?
2	?	?	?	?
3	?	?	?	?

Step 9: Extract the input string whenever the special character is found except '?'
Step 10: Plain text 631257 is received at destination end.
Step 11: Stop

V. CONCLUSIONS

The paper emphasizes a procedure whereby a plain text containing numbers are sent from one end of a network to another end by enciphering the text with the means of Neural Network architecture thereby ensuring network security. The encoding is done with the help of the hidden layer concept of Artificial Neural Network and symmetric-key cryptography. Thus the integrity, availability and confidentiality of the message passed over a network is secured and is not compromised.

REFERENCES

[1] William Stallings, *Cryptography and Network Security*, 2nd edition.
[2] Soumya Paul, et.al *"Design and Implementation of Network Security using Neural Network Architecture"*, Journal of Computing Vol 4, Issue: 8 August 2012, pp 115-120, ISSN: 2151-9617.

