INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

New Symmetric Key Cryptographic Algorithm Using Combined Bit Manipulation And MSA Encryption Algorithm: NJJSAA Symmetric Key Algorithm

Mr. Rangaswamy.D.A
PG Student, VLSI Design and Embedded System,
B.G.S. Institute of Technology, B.G.Nagar, Karnataka, India
Mr. Punithkumar.M.B
Assistant Professor, Dept. of Electronics and Communication Engineering,
B.G.S. Institute of Technology, B.G.Nagar, Karnataka, India

Abstract:
In the present work the authors have introduced a new advanced symmetric key cryptographic method called NJJSAA. The authors introduced new bit manipulation method for data encryption and decryption of any file. Nath et al already developed some symmetric key methods [1, 2, 3, 4] where they have used some randomized key matrix for encryption and decryption methods. In the present work the authors have used a bit manipulation method which include bit exchange, right shift and XOR operation on the incoming bits. To exchange bits the authors used a randomized key matrix of size (16x16) using the method developed by Nath et al (1). The present method allows the multiple encryption and multiple decryption. To initiate the encryption process a user has to enter a text-key which may be maximum of 16 characters long. From the text-key the authors have calculated randomization number and the encryption number. The method used was developed by Nath et al (1). A slight change in the text-key will change the randomization number and the encryption number quite a lot. Multiple encryption using bit exchange, bit right shift and XOR operations makes the system very secured. The present method is a block cipher method and it can be applied to encrypt data in sensor network or in mobile network. The advantage of the present method is that one can apply this method on top of any other standard algorithm such as DES, AES or RSA. The method is suitable to encrypt any large or small file. There is a scope to further enhance the present method of encryption.

Key words: *MSA Algorithm, Encryption, Decryption, XOR, Right Shift, Left Shift.*

1.Introduction

The massive development in internet technology in the last few years now it is a real challenge for the sender to send confidential data from one computer to another computer. There is no guarantee that between sender and receiver there is no one is intercepting those confidential data provided the data is not encrypted or properly protected. The security originality of data has now become a very important issue in data communication network. One cannot send any confidential or important message in raw form from one computer to another computer as any hacker can intercept that confidential message or important message. Now it is a common practice that the teachers are sending question papers over the mail. Now it is not a difficult job for a hacker to intercept that mail and retrieve the question paper if it is not encrypted. This may be very dangerous when someone is sending some confidential matter over the mail such as Bank transaction, Bank statement or any other confidential matter. There is no guarantee that the message will not be intercepted by anyone. This may be further worse during e-banking or ecommerce where the real data should not be intercepted by any hacker. When a client is sending some confidential matter from client machine to another client machine or from client machine to server then that data should not be intercepted by someone. The data should be protected from any unwanted intruder otherwise any massive disaster may happen all on a sudden. The disaster may happen if a sales manager of a company is sending some crucial data to his Managing Director related sales issue over the email. Suppose some intruder has intercepted that data from the internet and pass it to some other rival company. It is possible when the data moving from one computer to other computer is totally unprotected or not encrypted. To get rid of this problem one has to send the encrypted text or cipher text from client to server or to another client. Because of this hacking problem now a days network security and problem now a days network security and cryptography is an emerging research area where the people are trying to develop some good encryption algorithm so that no intruder can intercept the encrypted message. The so called classical cryptographic algorithm can be classified into two categories: (i) symmetric key cryptography where one key is used for both encryption and decryption purpose. (ii) Public key cryptography where two different keys are used one for encryption and the other for decryption purpose. The merits of symmetric key cryptography is that the key management is very simple as one key is used for both encryption as well as for decryption purpose. In case of symmetric key cryptography the key must be secret. In public key cryptography the

encryption key remains as public but the decryption key should be kept as secret key. The public key methods have got both merits as well as demerits. The problem of Public key cryptosystem is that one has to do massive computation for encrypting any plain text. Moreover in some public key cryptography the size of encrypted message may increase. Due to massive computation the public key crypto system may not be suitable in security of data in sensor networks. So the security problem in sensor node is a real problem. The present work we are proposing a symmetric key method called NJJSAA method which can be applied in sensor network, mobile network, ATM network.

Now we will describe our new advanced symmetric key cryptography. The present method is fully dependent on the text-key which is any string of maximum length 16 characters long. From the text-key we calculate two important parameters (i) Randomization number and (ii) Encryption number. To calculate this two parameters we use the method developed by Nath et al(1). We are giving below how we calculate the above two parameters:

- Step-1: Suppose key=AB Choose the following table for calculating the place value and the power of characters of the incoming key:

Length of Key (n)	Base Value (b)
1	17
2	16
3	15
4	14
5	13
6	12
7	11
8	10
9	9
10	8
11	7
12	6
13	5
14	4
15	3
16	2

TABLE 1: Length Of Key() Vs. Base Value(B)

(i): Calculate Sum = $\sum_{m=1}^{n} \text{ASCII} * b^{m}$----------(1)

where n=number of characters in the input text-key.

Now we calculate the sum for key="AB" using equation (1).

Here n=2, b=16

Sum=65*161 + 66 * 162 = 17936

Now we will show how we calculate 2 parameters from this sum:

(ii) *Randomization number*(n1):

num1=1*1+7*2+9*3+3*4+6*5=84

n1=sum mod num1=17936 mod 84=44

Note: if n1=0 then n1=num1 and if n1>32 then n1=Mod(n1 , 32)

(iii) *Encryption number(n2)*:

num2=6*1+3*2+9*3+7*4+1*5=72

n2=sum mod num2 =17936 mod 72 = 8

Note: if n2=0 then n2=num2 and if n2>32 then n2=Mod(n2 , 32)

- Step-2: Now we will describe how we perform the bit manipulation in the input stream of characters.
- Step2.1: Read 32 bytes at a time from the input file. Convert 32 bytes into 256 bits and store in some 1-dimensional array.
- Step-2.2: Choose the first bit from the bit stream and also the corresponding number(n) from the key matrix. Interchange the 1st bit and the n-th bit of the bit stream.
- Step-2.3: Repeat step-2.2 for 2nd bit,3rd bit,……256-th bit of the bit stream
- Step-2.4: Perform right shift by one bit.
- Step-2.5: Perform bit(1) XOR bit(2), bit(3) XOR bit(4),….bit(255) XOR bit(256)
- Step-2.6: Repeat step-2.4 with 2 bit right, 3bit right,….n2 bit right shift followed by step-2.5 after each completion right bit shift. After performing step-2 we obtain a file which is Encrypted file and it will not be possible to decrypt until and unless one knows the exact number of encryption and exact randomized matrix. One can use this encrypted file for sending data from one terminal to another terminal or from one terminal to one server.
- Step-3: In the present paper we use this encrypted file as the input file to apply MSA encryption algorithm (1) to encrypt this file twice to make the encryption

further strong. In the next section we will describe the MSA encryption algorithm in brief.

2.Meheboob, Saima & Asoke (MSA) Symmetric Key Cryptographic Method

Nath et al.(1) proposed a symmetric key method where they have used a random key generator for generating the initial key and that key is used for encrypting the given source file. MSA method is basically a substitution method where we take 2 characters from any input file and then search the corresponding characters from the random key matrix and store the encrypted data in another file. In present work we have the provision for encrypting message multiple times. The key matrix contains all possible characters (ASCII code 0 to 255) in a random order. The pattern of the key matrix will depend on text_key entered by the user. Nath et al. proposed algorithm to obtain randomization number, encryption number from the initial text_key. We have given a long trial run on text_key and we found that it is very difficult to match the three above parameters for 2 different text_key which means if someone wants to break our encryption method then he/she has to know the exact pattern of the text_key otherwise it will not be possible to obtain two sets of identical parameters from two different text_key. We have given several trial runs to break our encryption method but we found it is almost unbreakable. For pure text file we can apply brute force method to decrypt small text but for any other file such any binary file we cannot apply any brute force method and it does not work. In TABLE-I we have described how we can calculate the randomization number and the encryption number. After we calculate the above two parameters we then create the original key matrix which is of size 16x16 which contains all characters from 0-255 ASCII codes shown in TABLE-2. Then we make the key matrix random by applying some simple function calls one after the other:

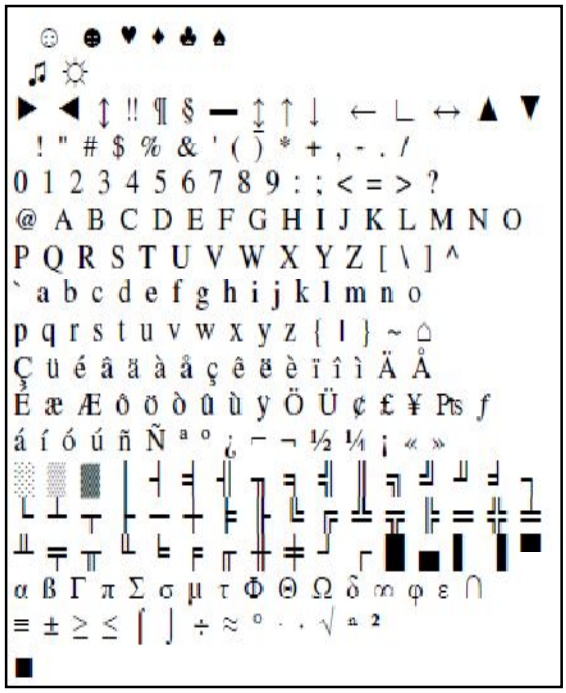


Table 2: The Original Key Matrix

Now we apply the following randomization methods one after another in a serial manner:

- Step-1: call Function cycling()
- Step-2: call Function upshift()
- Step-3: call Function downshift()
- Step-4:call Function leftshift()
- Step-5:call Function rightshift()

For detail randomization methods we refer to the done by Nath et al(1).

We will now show how the above randomization works on a particular key matrix.

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Table 3: Original Key Matrix

L	A	B	C
I	G	K	D
M	F	J	H
N	O	P	L

Table 4: Key Matrix Calling Function Cycling()

M	E	F	A
P	K	L	D
N	I	O	G
C	J	B	H

Table 5: Key Matrix Calling Function Upshift()

H	M	N	E
I	F	O	A
G	P	C	K
J	L	B	D

Table 6: Key Matrix Calling Function Rightshift()

D	O	B	A
H	G	M	P
N	C	E	K
I	J	F	L

Table 7: Key Matrix Calling Function Downshift()

B	K	A	I
D	C	O	E
M	L	P	N
H	J	G	F

Table 8: Key Matrix Calling Function Leftshift()

Now we will describe how we perform the encryption process using MSA algorithm (1):

We choose a 4X4 simple key matrix:

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Table 9: Key Matrix (4x4)

- Case-I: Suppose we want to encrypt FF then it will take as GG which is just one character after
- F in the same row.
- Case –II: Suppose we want to encrypt FK where F and K appears in two different rows and twodifferent columns. FK will be encrypted to KH (FK->GJ->HK->KH).
- *Case-III*: Suppose we want to encrypt EF where EF occurs in the same row. Here EF will be converted to HG.

After encrypting 2 bytes we write the encrypted bytes on a new output file. The entire encryption method we apply multiple times and the encryption number will be determined by the process we have described in TABLE-1.

2.1.Decryption Method

The decryption method will be the just the reverse process of encryption method as mentioned below:

(i) Apply MSA decryption algorithm on the encrypted message.

(ii) Apply Bit manipulation decryption algorithm that means applying leftshift of bits, XORING bits and then interchanging bits.

(iii) We repeat the same number of decryption process so that we get back the original file.

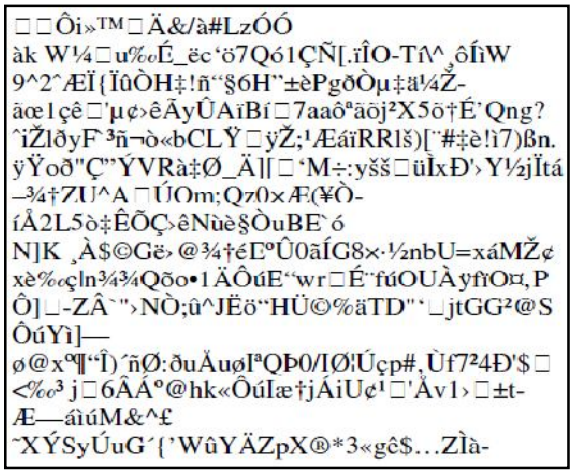
If there is any change in decryption process then we will not be able to get back the original file.

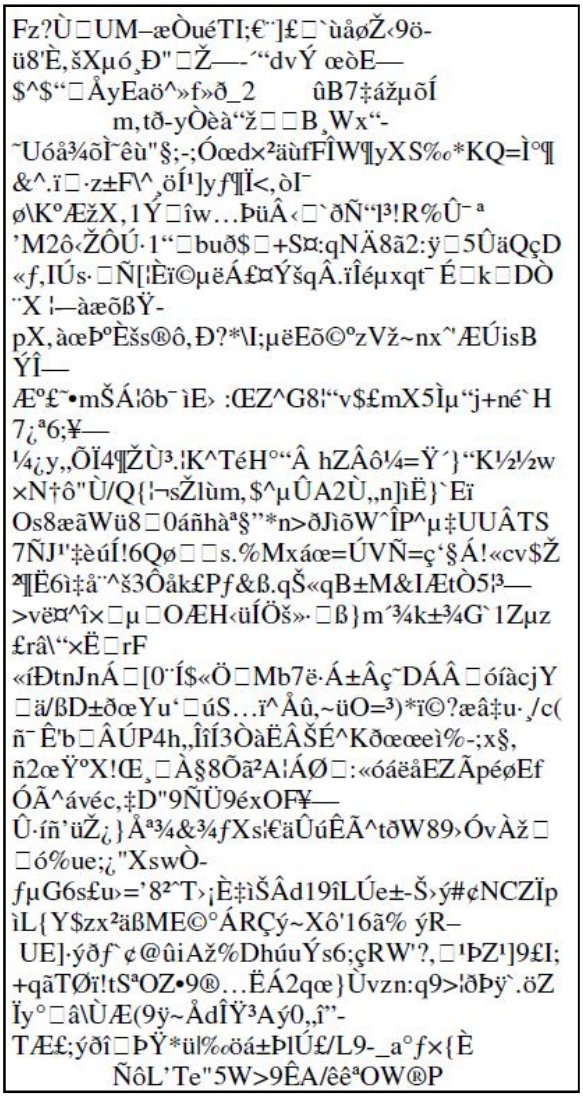
3.Results and Discussion

Welcome to CSNT 2011

The International Conference on Communication Systems and Network Technologies (CSNT- 2011) is organized to address various issues to prosper the creation of intelligent solutions in future. The aim is to bring together worldwide leading researchers, developers, practitioners and educators interested in advancing the state of the art in computational intelligence and communication networks for exchanging knowledge that encompasses a broad range of disciplines among various distinct communities. It is expected that researchers will bring new prospect for collaboration across disciplines and gain idea facilitating novel breakthrough. The theme for this conference is innovating and inspiring the researchers to adopt the outcome for implementation. The conference will provide an exceptional platform to the researchers to meet and discuss the utmost solutions, scientific results and methods in solving intriguing problems with people that actively involved in these evergreen fields. The 3-day conference commencing from 03 June, will feature prominent keynote speakers, tutorials and paper presentation in parallel sessions. All accepted papers will appear in conference proceedings published by the Conference Publishing Services (CPS) and papers will be available on IEEE Explore. A first of its kind in the historical Jammu, CSNT 2011 will no doubt be proven to be exciting and educative. The General Chairs, along with the entire team cordially invite you to take part in this upcoming event and together we flourish it into a most memorable experience. Organizing committee will also plan for various tours to various historical Places around Jammu City. G.S.Tomar, Ajith Abraham & Vipan Kakkar CSNT-2011.

3.1.Name of the Encrypted File: Output.txt





We apply our encryption and decryption algorithm on various other files and we found in every case it was working perfectly ok. In TABLE-10 the different type of files are given below:

S.No.	Original File(size of File in byte)	Encrypted File(size of File in byte)	Decrypted File(size of File in byte)	Remarks
1.	Sample.doc (22,016 bytes)	Sample1.doc (22,016 bytes)	Sample2.doc (22,016 bytes)	Original file and the decrypted file identical
2.	Sample.avi (82,944 bytes)	Sample1.avi (82,944 bytes)	Sample2.avi (82,944 bytes)	Original file and the decrypted file identical
3.	Sample.bmp (1,440,054 bytes)	Sample1.bmp (1,440,054 bytes)	Sample2.bmp (1,440,054 bytes)	Original file and the decrypted file identical
4.	Sample.com (69,886 bytes)	Sample1.com (69,886 bytes)	Sample2.com (69,886 bytes)	Original file and the decrypted file identical
5.	Sample.exe (1,26,976 bytes)	Sample1.exe (1,26,976 bytes)	Sample2.exe (1,26,976 bytes)	Original file and the decrypted file identical
6	Sample.jpeg (7,189 bytes)	Sample1.jpeg (7,189 bytes)	Sample2.jpeg (7,189 bytes)	Original file and the decrypted file identical
7.	Sample.mp3 (5,184 bytes)	Sample1.mp3 (5,184 bytes)	Sample2.mp3 (5,184 bytes)	Original file and the decrypted file identical
8.	Sample.pdf (76,864 bytes)	Sample1.pdf (76,864 bytes)	Sample2.pdf (76,864 bytes)	Original file and the decrypted file identical
9.	Sample.wav (9,01,164 bytes)	Sample1.wav (9,01,164 bytes)	Sample2.wav (9,01,164 bytes)	Original file and the decrypted file identical
10.	Sample.xls (20,480 bytes)	Sample1.xls (20,480 bytes)	Sample2.xls (20,480 bytes)	Original file and the decrypted file identical
11.	Sample.ppt (37,84,192 bytes)	Sample1.xls (37,84,192 bytes)	Sample2.xls (37,84,192 bytes)	Original file and the decrypted file identical
12.	Sample.txt (2,992 bytes)	Sample1.txt (2,992 bytes)	Sample2.txt (2,992 bytes)	Original file and the decrypted file identical

Table 10: Consolidated Report On Njjsaa Encryption And Decryption Algorithm

4.Conclusion

In the present work we use the maximum encryption number=32 and maximum randomization number=32. The key matrix of size 16x16. This key may be generated in 256! ways. The present method uses two distinct methods in bit manipulation method we use block cipher method and in MSA method we use stream cipher method. The present method uses first bit manipulation and then MSA encryption method. The present method may be applied in serial manner to increase the strength of the encryption as well as decryption method. There is lot of scope to modify the present method. The merit of this method is that it is almost impossible to break the encryption algorithm without knowing the exact key matrix. We propose that this encryption method can be applied for data encryption and decryption in banks, in defense, in government sectors for sending confidential data. The present algorithm may be used for database encryption also.

5.Reference

1. Symmetric key cryptography using random key generator, A.Nath, S.Ghosh, M.A.Mallik, Proceedings of International conference on SAM-2010 held at Las vegas(USA) 12-15 July,2010, Vol-2,P-239- 244

2. Advanced Symmetric key Cryptography using MSA method: DJSSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Soumitra Mondal, Suvadeep Dasgupta and Asoke Nath, Journal of Computing, Vol 3, issue 2, Page 66-71(Feb 2011).

3. Data Hiding and Retrieval, A.Nath, S.Das, A.Chakrabarti, Proceedings of IEEE International conference on Computer Intelligence and Computer Network held at Bhopal from 26-28 Nov, 2010.

A new Symmetric key Cryptography Algorithm using extended MSA method: DJSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Suvodeep Dasgupta and Asoke Nath, Accepted for publication in CSNT-2011 to be held at SMVDU 03/06/2011 to 05/06/2011.

4. Symmetric key Cryptography using modified DJSSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, Accepted for publication in WORLDCOMP-2011 to be held at Las Vegas,USA 18-21 Jul 2011

5. Cryptography and Network, William Stallings, Prectice Hall of India.

6. Modified Version of Playfair Cipher using Linear Feedback Shift Register, P. Murali and Gandhidoss Senthilkumar, UCSNS International journal of Computer Science and Network Security, Vol-8 No.12, Dec 2008.