Bit and Byte Level Generalized Modified Vernam Cipher Method with Feedback

Prabal Banerjee
Department of Computer Science
St. Xavier's College (Autonomous), Kolkata
Kolkata, India

Asoke Nath
Department of Computer Science
St. Xavier's College (Autonomous), Kolkata
Kolkata, India

ABSTRACT
In the present paper the authors have introduced a new symmetric key cryptographic method where they have clubbed both bit level and byte level generalized modified vernam cipher method with feedback. Nath et al already developed methods bit level encryption standard(BLES) Ver-I and Ver-II where they have used extensive bit level permutation, bit exchange, bit xor and bit shift encryption method. Nath et al also developed a method bit level generalised vernam cipher method. In the present study the authors have used both bit level generalized vernam cipher method and after that byte level vernam cipher method using feedback. Due to introduction of feedback in both bit level as well as byte level vernam cipher method the common attacks such as differential attack or known plain text attack is not applicable in the present method. In the present paper the authors have used random key generator to construct the keypad for vernam cipher method. The authors have introduced a special bit manipulation method so the encryption algorithm will work even for all characters with ASCII Code 0 or all characters with ASCII Code 255. The most of the standard algorithm will fail to encrypt a file where all characters are ASCII '0' or all characters with ASCII '255' where as the present method will be able to encrypt a file where all characters are ASCII '0' or all characters are ASCII '255'. The present method will be effective for encrypting short message, password, confidential key etc.

General Terms
encryption, bit-level

Keywords
generalized vernam cipher method, bit exchange, random key, feedback, differential attack

1. INTRODUCTION
The growth in data communication and network in the last few years now it is a real problem for the sender to send confidential data from one computer to another computer. If we are sending data in open Internet then there is no guarantee that between the sender and the receiver there is no one intercepting those confidential data especially if the data is not encrypted or properly protected. The security or the originality of data has now become a very important issue in data communication network. Any confidential or important data should not be sent from one computer to another One cannot send any confidential or important message in original form to a computer from another computer as any hacker can intercept that confidential message or important message. It is now a common practice to send marks, attendance or question papers, bank statement over e-mail. But this method is not fully secured as anybody can intercept the data from internet and misuse it. It is not at all difficult task for a hacker to intercept an e-mail and retrieve the confidential data especially if it is not encrypted. It must be ensured that in e-business, air or railway reservation system or in credit card or debit card system the data should not be tampered or intercepted by an unauthorized person. Any confidential data must be protected from any unwanted intruder to avoid any disaster. The disaster may happen in any business house if some higher official in that house make some important data out through e-mail. The disaster may happen when the data is sent from one computer to other computer in an unprotected manner. To overcome this problem one has to send the encrypted text or cipher text from client to server or to another client. To protect data from intruder or hacker now network security and cryptography is an emerging research area where the programmers are trying to develop some strong encryption algorithm so that no intruder can intercept the encrypted message. These methods are called classical cryptographic algorithm and those methods can be divided 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 merit 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 symmetric key algorithm the key is called secret key and it should be known to sender and receiver both and no one else. The public key methods have 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 a case like sensor networks where the computation time is very important. In sensor networks symmetric key cryptosystem is preferred rather than public key cryptosystem. In the present work the authors are proposing a symmetric key method where they have used bit level modified generalized vernam cipher method using feedback method after that byte level generalized modified vernam cipher method was applied with feedback. The present method can be applied in corporate sectors, academic institutions, Defense network etc.

The key element is the bit exchange depending on the randomized matrix from which the actual key is extracted depending upon the plaintext size. The key is then shuffled by generating all the anagrams possible. As all the characters extracted from the randomized matrix is unique, hence the anagrams generated are free of any repetitions. As the applied key generated from actual key is considerably randomized, the data finally gets shuffled to such an extent that without knowing the process and key, it would be impossible to

decrypt. The authors have implemented the bit-wise exchange method as follows:

Firstly, they begin with initial transformation where the data is broken down to its corresponding bits and stored in a file.

Secondly, randomization number and encryption number is calculated based on input key and file size. The generated random key and its corresponding anagrams are stored in a file.

Thirdly, key and bits are extracted from their corresponding files, worked on and saved in a third file. This process is executed till encryption number is reached , i.e., until all the bits have been successfully worked on.

Fourthly, the file is reversed, saved and then again worked on in a similar manner.

After bitwise encryption of the plaintext, the cipher text (obtained after bit-level encryption method) is encrypted using byte-level modified generalized Vernam Cipher method. In byte level vernam cipher method the authors have taken one byte at a time each from the text and the key matrix. They added the integer equivalent of the extracted bytes and the feedback which is the result of the previous such operation. The final result is taken modulo 256 and is stored. At the same time it is also sent as the feedback to the next byte operation. This way the text is encrypted twice, first bit level modified generalized Vernam Cipher method and after that Byte level Generalized modified vernam cipher method to make the entire system more secured.

The multiple key generation from a set of random characters and both bitwise and bytewise encoding make the system very secure.

2. ALGORITHM OF MODIFIED GENARALIZED VERNAM CIPHER METHOD USING FEEDBACK

The present method is dependent both on the text-key and the plaintext filesize. From the text-key a randomization matrix is generated using the method developed by Nath et al(1). The algorithm of bit-level and byte level generalized vernam cipher method is given as follows:

Step 1: Call Bitwise_Encrypt() //writes into file C

Step 2: Call Bytewise_Encrypt(File C)

Step 3: Exit

2.1 Function Bitwise_Encrypt ()

Step 1: Input a key string K

Step 2: Generate a 16x16 matrix (mat[][]) using the MSA algorithm for the key string K

Step 3: Input Filename P which is the plaintext on which the encryption is to be applied

Step 4: size=no. of bytes in file P, rand_no=1

Step 5: If size>=factorial of rand_no, rand_no=rand_no+1, repeat step 5

Step 6: Take 'rand_no' amount of characters from mat[][] and put in string buf

Step 7: Find all anagrams of buf and put in file F

Step 8: Call Encrypt_byte(P,F,mat)

Step 9: Reverse the contents of A into which function Encrypt_byte has written

Step 10: Call Encrypt_bit(A,mat)

Step 11: limit=number of bytes in file B

Step 12: i=0

Step 13: if i>=limit/8, goto step 23

Step 14: add=j=0

Step 15: if j>=8, goto step 20

Step 16: Read a character from B and store into ch

Step 17: add=add+(ch-48)*power(7-j)

Step 18: j=j+1

Step 19: Goto step 15

Step 20: Convert add to character and print into file C

Step 21: i=i+1

Step 22: Goto step 13

Step 23: Return control to calling function

2.2 Function Bytewise_Encrypt (File C)

Step 1: limit=number of bytes in File C, k=carry=0

Step 2: if k>limit , goto step 11

Step 3: Read a character from file C and store to ch

Step 4: ch=ch+mat[i][j]+carry

Step 5: Write ch to file D

Step 6: carry= ch % 256

Step 7: j=j+1, k=k+1

Step 8: if j=16, i=i+1 and j=0

Step 9: if i=16, i=0

Step 10: Goto step 2

Step 11: Exit

2.3 Function Encrypt_byte (File P, File F, mat[16][16])

Step 1: Find the number of bytes in the plaintext file P on which the encryption is to be applied. Let it contain no_of_bytes.

Step 2: carry=0

Step 3: Read a character from file F and store to ch

Step 4: Call char_to_bit(ch,key_bit)

Step 5: Read a byte ch from P

Step 6: Call char_to_bit(ch,text_pattern)

Step 7: k=0

Step 8: if k>=8, goto step 16

Step 9: add=text_pattern[k]+key_bit[k]+carry

Step 10: if add=1 or add=3, cipher_bit=1

else cipher_bit=0

Step 11: if add>=2, carry=1

else carry=0

Step 12: If carry=0, carry=cipher_bit

Step 13: Print cipher_bit into file A

Step 14: k=k+1

Step 15: Goto Step 8

Step 16: no_of_bytes=no_of_bytes-1

Step 17: If no_of bytes>0, goto step 3

Step 18: Return control to calling function

2.4 Function Encrypt_bit (File A, mat[16][16])

Step 1: Find the number of bytes in A on which the encryption is to be applied. Let it contain no_of_bytes.

Step 2: carry=0

Step 3: Read a character from file F and store to ch

Step 4: Call char_to_bit(ch,key_bit)

Step 5: n=0

Step 6: if n>=8, goto step 11

Step 7: Read a char from A

Step 8: text_pattern[n]=ch-48

Step 9: n=n+1

Step 10: Goto step 6

Step 11: k=0

Step 12: if k>=8, goto step 16

Step 13: add=text_pattern[k]+key_bit[k]+carry

Step 14: if add=1 or add=3, cipher_bit=1

else cipher_bit=0

Step 15: if add>=2, carry=1

else carry=0

Step 16: If carry=0, carry=cipher_bit

Step 17: Print cipher_bit into file B

Step 18: k=k+1

Step 19: Goto Step 8

Step 20: no_of_bytes=no_of_bytes-8

Step 21: If no_of bytes>0, goto step 3

Step 22: Return control to calling function

2.5 Function power (integer p) -- Function returns 2 to the power p

Step 1: ans=2

Step 2: if p!=0, return 1

Step 3: p=p-1

Step 4: if p=0, goto step 7

Step 5: ans=ans*2

Step 6: Goto step 4

Step 7: return ans

Step 8: Return control to calling function

2.6 Function char_to_bit (integer c, integer a[]) --Function changes a character to its corresponding bit pattern

Step 1: i=0

Step 2: if i>=8, goto step

Step 3: if ((ch)AND(1<<i))>0, a[7-i]=1

else a[7-i]=0

Step 4: Return control to calling function

3. DECRYPTION ALGORITHM

Step 1: Call Bytewise_Decrypt() //writes into file A

Step 2: Call Bitwise_Decrypt(File A)

Step 3: Exit

3.1 Function Bitwise_Decrypt (File P)

Step 1: Input a key string K

Step 2: Generate a 16x16 matrix (mat[][]) using the MSA algorithm for the key string K

Step 3: size=no. of bytes in file P, rand_no=1

Step 4: If size>=factorial of rand_no, rand_no=rand_no+1, repeat step 4

Step 5: Take 'rand_no' amount of characters from mat[][] and put in string buf

Step 6: Find all anagrams of buf and put in file F

Step 7: Call Encrypt_byte(P,F,mat)

Step 8: Reverse the contents of B into which function Encrypt_byte has written

Step 9: Call Encrypt_bit(B,mat)

Step 10: limit=number of bytes in file B

Step 11: i=0

Step 12: if i>=limit/8, goto step 22

Step 13: add=j=0

Step 14: if j>=8, goto step 19

Step 15: Read a character from B and store into ch

Step 16: add=add+(ch-48)*power(7-j)

Step 17: j=j+1

Step 18: Goto step 14

Step 19: Convert add to character and print into file C

Step 20: i=i+1

Step 21: Goto step 12

Step 22: Exit

3.2 Function Bytewise_Decrypt (File P)

Step 1: Input Filename P which is the plaintext on which the encryption is to be applied

Step 2: limit=number of bytes in File P, k=carry=0

Step 3: if k>limit , goto step 12

Step 4: Read a character from file P and store to ch

Step 5: ch=ch - mat[i][j] - carry

Step 6:if ch<0, ch=ch+255

Step 7: carry= ch , Store ch in File A

Step 8: j=j+1, k=k+1

Step 9: if j=16, i=i+1 and j=0

Step 10: if i=16, i=0

Step 11: Goto step 3

Step 12: Exit

4. RANDOMIXZATION OF MATRIX USING MEHEBOOB, SAIMA & ASOKE(MSA) RANDOMIZATION METHOD

We first create a square matrix of size n x n where n can be 4, 8, 16 and 32. First we store numbers 0 to (n*n-1). We apply the following randomization techniques to create a random key matrix. The detail description of randomization methods is given by Nath et.al[1].

The following Randomization methods were applied on initial key matrix to obtain a randomized key matrix:

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()

5. RESULTS AND DISCUSSION

We have made an extensive study on various plain text files and encrypt using (i) bit-level generalized modified vernam cipher method with feedback and (ii) Bit level followed by Byte level modified generalized Vernam Cipher Method with feedback. The present result shows that the inclusion of byte level Vernam cipher method gives better result than only bit level modified generalized vernam cipher method.

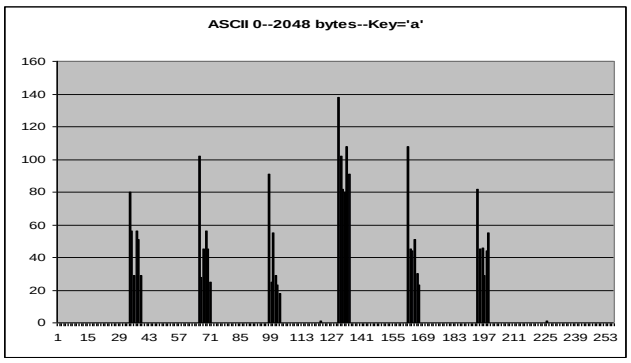

Fig-1: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '0' and Key='a' after bitwise encryption only.

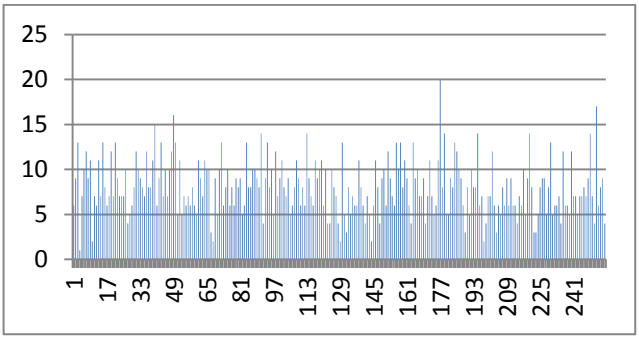

Fig-2: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '0' and Key='a' after bit and bytewise encryption.

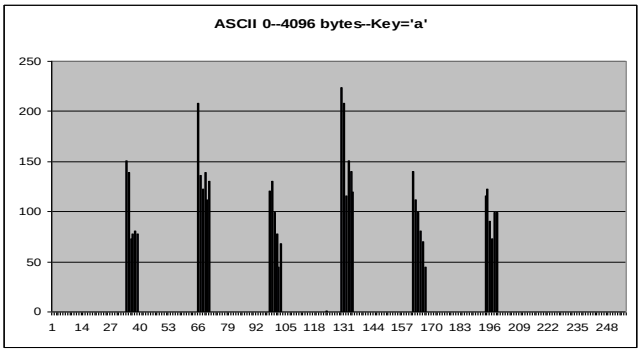

Fig-3: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '0' and Key='a' after bitwise encryption only.

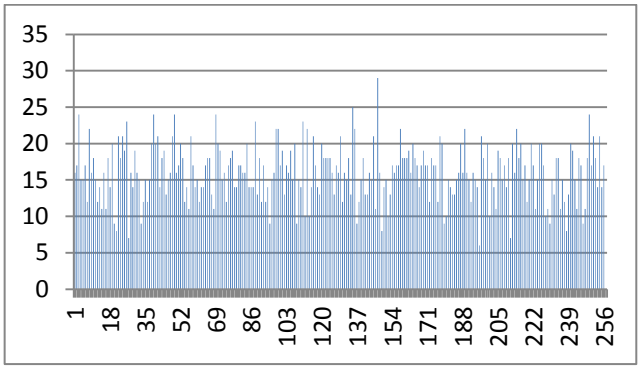

Fig-4: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '0' and Key='a' after bit and bytewise encryption.

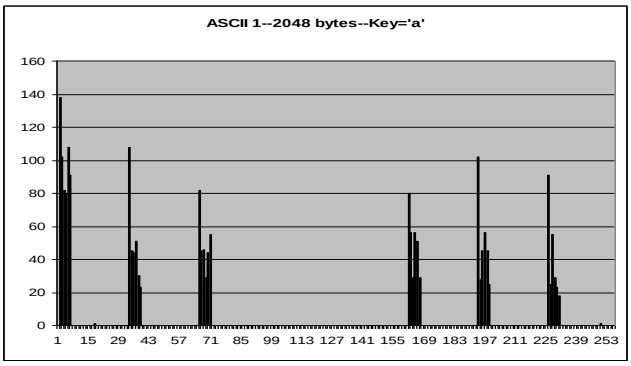

Fig-5: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '1' and Key='a' after bitwise encryption only.

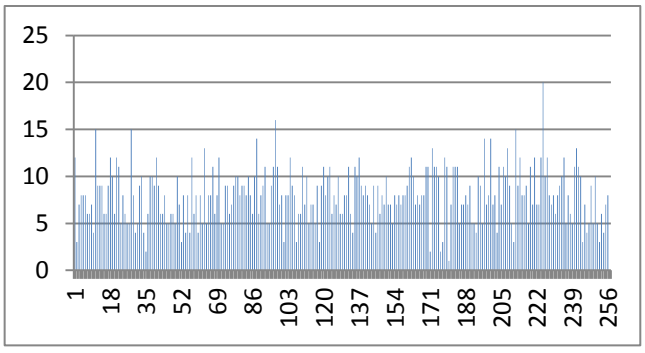

Fig-6: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '1' and Key='a' after bit and bytewise encryption.

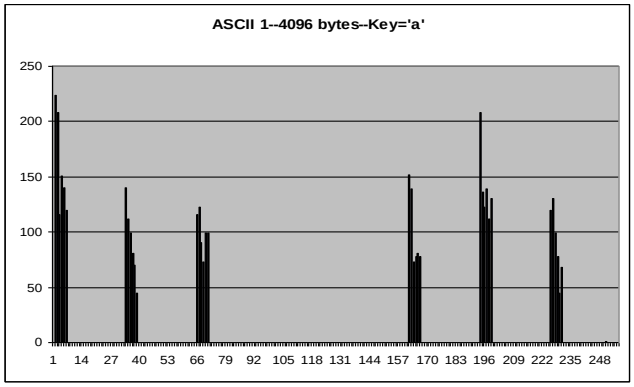

Fig-7: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '1' and Key='a' after bitwise encryption only.

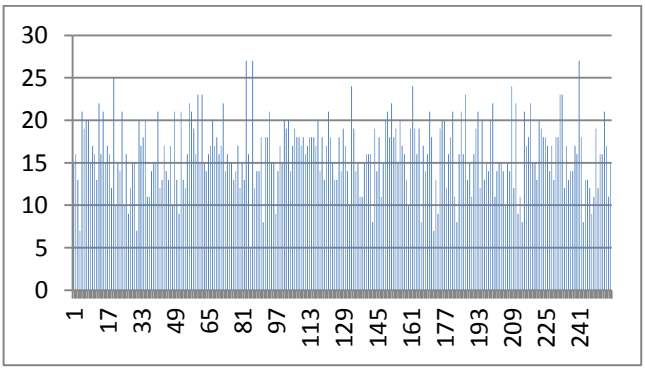

Fig-8: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '1' and Key='a' after bit and bytewise encryption.

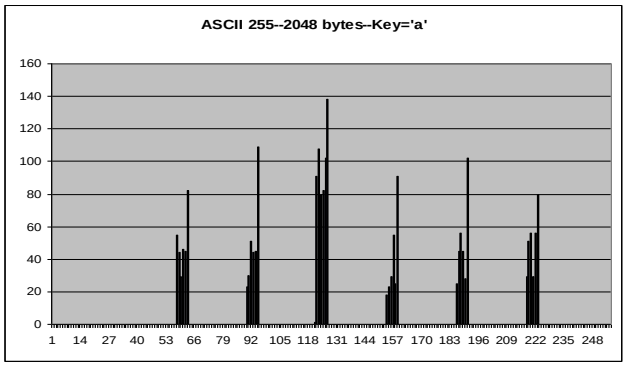

Fig-9: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '255' and Key='a' after bitwise encryption only.

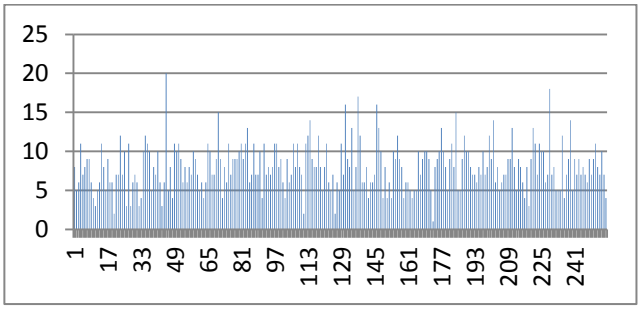

Fig-10: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '255' and Key='a' after bit and bytewise encryption.

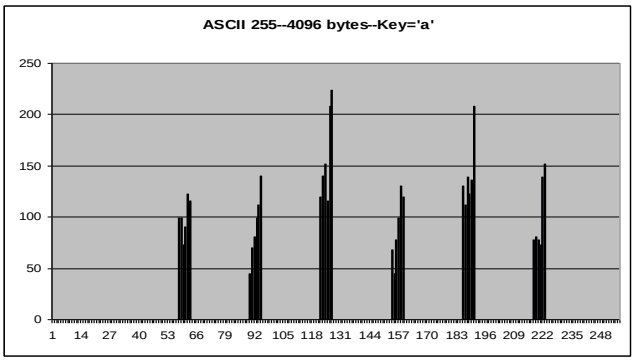

Fig-11: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '255' and Key='a' after bitwise encryption only.

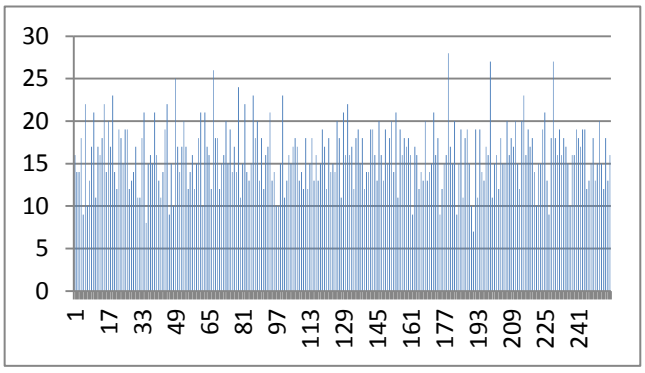

Fig-12: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '255' and Key='a' after bit and bytewise encryption.

Table 1: A Plain Text file containing a paragraph and its encrypted file after bit and bytewise encryption.

With one hundred and fifty one years of service to the nation, St. Xavier's College, Kolkata, has grown today into a leading educational institution in India. Both St. Xavier's School and College are proud of their contribution to the cause of education and culture in Bengal for 151 years. These twin institutions have produced many great educationists and students who earned regard and respect for their Alma Mater.	}"#uoMíð¢9DeO°°%Ë¢a×ïmØŽ ¶ DIbàŠƒ \..3ãd□¿^€\nõ\£ãšÌÒO„ ® ×ß->äüŒ_ÏØCeÕ§ã÷<..sílå7]³- ±□u?b€□·□N1Ï25Q-h□0!íªj Â□cÿå□€ÃH6ì9:□¿q□□€dìS..4 ˆ µ ‡ I□ÁÕšS^□¹hÙf+□□1öúÄPáÑè þ□°□□q□ž T§·□<§Tp¶Û=□à□$zµ□ŽÎvËÐŠ —A□»±?66ÏDø□OÇ□ŽRJ ejã»±Ž ßœv)Ñålû, rŠ - ïøäÛÎ ™æ" —DHÖdTäô>|gRæ9×¿ TiŠ™, " úY·£"/0hDßð□·é©U4ûM é□OÿùÄ□ö«Tr4□Ýò§fœêw□þê¯ † ¥† - • ‡ [óM¥—£¿f£ˆ ..Ñ` • ®s† (Ü÷ â□Ó,žYÎ□ £Â– ®Û³qÃ ' Š w• ôþ8CB¯D2˜ |□ýÿj□€c□Føê*□Ž~¹è

6. CONCLUSION AND FUTURE SCOPE

The encrypted text cannot be decrypted without knowing the exact initial random matrix. The size of random matrix taken is 16x16. The numbers in 16x16 may be arranged in 256! Ways. To complete the whole process the authors have chosen any of the random matrix to perform bit exchange method and there is no similarity between any two matrices and even if there is then it is very hard to find out the similar ones. The order of encryption applied in this paper is bit-level first and then byte level. Multilevel applications of these techniques in different order will yield considerably different results. The spectral analysis shows that the present method is free from standard cryptography attacks namely brute force attack, known plain text attack and differential attack. The present method will be most effective to encrypt short message such as SMS in mobile phone, password encryption and any type of confidential message. If the file size is large then the present method will take more time to encrypt. So therefore our proposed method may be used in defence systems, Banking systems, Sensor networks, Mobile computing etc. The present method may be further upgraded by introducing bit level bit exchange method which is used in BLES Version-I.

7. ACKNOWLEDGEMENT

The authors are very much grateful to the Department of Computer Science for giving the opportunity to work on symmetric key Cryptography. One of the authors (AN) sincerely expresses his gratitude to Fr. Dr. Felix Raj, Principal of St. Xavier's College (Autonomous) for giving constant encouragement in doing research in cryptography.

8. REFERENCES

[1] Symmetric Key Cryptography using Random Key generator: Asoke Nath, Saima Ghosh, Meheboob Alam Mallik: "Proceedings of International conference on security and management(SAM '10)" held at Las Vegas, USA July 12-15, 2010), Vol-2, Page: 239-244(2010).

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

[3] A new Symmetric key Cryptography Algorithm using extended MSA method: DJSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Suvadeep Dasgupta and Asoke Nath : Proceedings of IEEE International Conference on Communication Systems and Network Technologies, held at SMVDU(Jammu) 03-06 June,2011, Page-89-94(2011).

[4] New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm: Neeraj Khanna, Joel James,Joyshree Nath, Sayantan Chakraborty, Amlan Chakrabarti and Asoke Nath : Proceedings of IEEE CSNT-2011 held at SMVDU(Jammu) 03-06 June 2011, Page 125-130(2011).

[5] Symmetric key Cryptography using modified DJSSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, Proceedings of International conference Worldcomp 2011 held at Las Vegas 18-21 July 2011, Page-306-311, Vol-1(2011).

[6] An Integrated symmetric key cryptography algorithm using generalized vernam cipher method and DJSA method: DJMNA symmetric key algorithm: Debanjan Das, Joyshree Nath, Megholova Mukherjee, Neha Chaudhury and Asoke Nath: Proceedings of IEEE International conference: World Congress WICT-2011 held at Mumbai University 11-14 Dec, 2011, Page No.1203-1208(2011).

[7] Symmetric key cryptosystem using combined cryptographic algorithms- generalized modified vernam cipher method, MSA method and NJJSAA method: TTJSA algorithm – Trisha Chatterjee, Tamodeep Das, Joyshree Nath, Shayan Dey and Asoke Nath, Proceedings of IEEE International conference: World Congress WICT-2011 t held at Mumbai University 11-14 Dec, 2011, Page No. 1179-1184(2011).

[8] Symmetric key Cryptography using two-way updated Generalized Vernam Cipher method: TTSJA algorithm, International Journal of Computer Applications (IJCA, USA), Vol 42, No.1, March, Pg: 34 -39(2012).

[9] Ultra Encryption Standard(UES) Version-I: Symmetric Key Cryptosystem using generalized modified Vernam Cipher method, Permutation method and Columnar Transposition method, Satyaki Roy, Navajit Maitra, Joyshree Nath,Shalabh Agarwal and Asoke Nath, Proceedings of IEEE sponsored National Conference on Recent Advances in Communication, Control and Computing Technology -RACCCT 2012, 29-30 March held at Surat, Page 81-88(2012).

[10] An Integrated Symmetric Key Cryptographic Method – Amalgamation of TTJSA Algorithm, Adbvanced Caeser Cipher Algorithm, Bit Rotation and reversal Method: SJA Algorithm., International Journal of Modern Education and Computer Science, Somdip Dey, Joyshree Nath, Asoke Nath,(IJMECS), ISSN: 2075-0161 (Print), ISSN: 2075-017X (Online), Vol-4, No-5, Page 1-9,2012.

[11] An Advanced Combined Symmetric Key Cryptographic Method using Bit manipulation, Bit Reversal, Modified Caeser Cipher(SD-REE), DJSA method, TTJSA method: SJA-I Algorithm, Somdip dey, Joyshree Nath, Asoke Nath, International Journal of Computer Applications(IJCA 0975-8887, USA), Vol. 46, No.20, Page- 46-53,May, 2012.

[12] Ultra Encryption Standard(UES) Version-IV: New Symmetric Key Cryptosystem with bit-level columnar Transposition and Reshuffling of Bits, Satyaki Roy, Navajit Maitra, Joyshree Nath, Shalabh Agarwal and Asoke Nath, International Journal of Computer Applications(IJCA)(0975-8887) USA Volume 51-No.1.,Aug, Page. 28-35(2012)

[13] Bit Level Encryption Standard(BLES) : Version-I, Neeraj Khanna, Dripto Chatterjee, Joyshree Nath and Asoke Nath, International Journal of Computer Applications(IJCA)(0975-8887) USA Volume 52-No.2.,Aug, Page.41-46(2012).

[14] Bit Level Generalized Modified Vernam Cipher Method with Feedback, Prabal Banerjee, Asoke Nath, Proceedings of International Conference on Emerging Trends and Technologies held at Indore, Dec 15-16,2012.

[15] Cryptography and Network Security, William Stallings, Prentice Hall of India.