International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-2 Number-4 Issue-6 December-2012

Bit Level Generalized Modified Vernam Cipher Method with Feedback

Prabal Banerjee[1], Asoke Nath[2]
Department of Computer Science St. Xavier's College (Autonomous) Kolkata, India

Abstract

The present paper discusses how the vernam cipher method can be extended to a new symmetric key cryptographic method called Bit Level generalized modified vernam cipher method with feedback. Nath et al already developed a method called BLES version-I where they have used extensive bit level permutation encryption method. Nath et. al has already developed bit manipulation method called NJJSAA where the authors mainly used bit level right shift, bit level XOR operation. In the present paper the authors have used bit level modified vernam cipher method. using random key generator. 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 standard encryption algorithm will fail to encrypt a file where all characters are ASCII '0' or all characters with ASCII '255' but 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. The spectral analysis in the result sections shows that the present method is free from known plain text attack, differential attack or any type of brute force attack.

Keywords

BLES, NJJSAA, bit exchange, random key, differential attack

1. Introduction

With the tremendous 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 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. 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. Sending question papers or sending bank statement is now a common practice over the mail. But this method is not fully secured as anybody can intercept the data from internet and misuse it. Nowadays it is not at all difficult for a hacker to intercept an e-mail and retrieve the confidential data especially if it is not encrypted. There are many sectors such as Banking, E-business, E-commerce, Railway or Air Reservation system where 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 if a sales manager of a company is sending some crucial data related to the sales of the company to his Managing Director over the e-mail and some intruder intercepts that data from the internet and passes it on to some other rival company. This type of disaster may happen when the data is moving 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 any kind of hacking problems nowadays 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 encryptions 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. In public key cryptography there are two keys used one key is called public key which is used only for encryption purpose and the other is called private key which is used only for decryption purpose. The public key is not secret and it can be shared by anybody but the decryption key should be kept by the receiver only and by 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. So the security problem in sensor node is a real problem. In the present work we are proposing a symmetric key method where we have used modified generalized vernam cipher method using feedback method which can be applied in corporate sectors, academic institutions, Defence network etc.

The present method uses bit manipulation methods with bit-level vernam cipher method. 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 randomised matrix is unique, hence the anagrams generated are free of any repetitions. As the applied key generated from actual key is considerably randomised, the data finally gets shuffled to such an extent that without knowing the process and key, it would be impossible to decrypt. We have implemented the bit-wise exchange method as follows:

Firstly, we 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.
The multiple key generations from a set of random characters make our 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 we generate a randomization matrix using the method developed by Nath et al(1). We are giving below algorithm of BLES:
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,mat)
Step 9: Reverse the contents of B into which function Encrypt_byte has written
Step 10: Call Encrypt_bit(B,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: Exit

Function Encrypt_byte (File B, 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 B
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

Function Encrypt_bit(File B, 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: n=0
Step 6: if n>=8, goto step 11
Step 7: Read a byte ch from P
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

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

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

Decryption Algorithm

As the encryption and decryption process of the discussed algorithm is self complimentary, the application of the encryption algorithm on the ciphertext will produce the original plaintext, provided the given key matches for both of them.

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()

3. Results and Discussion

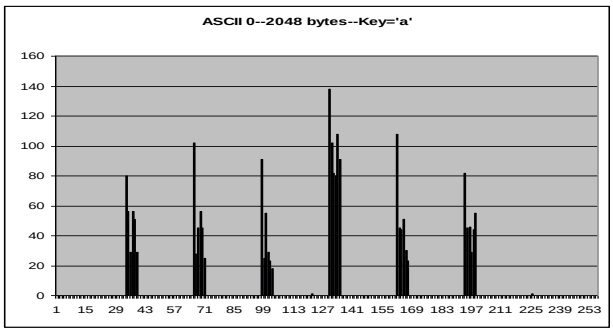

Fig-1: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '0' and Key='a'.

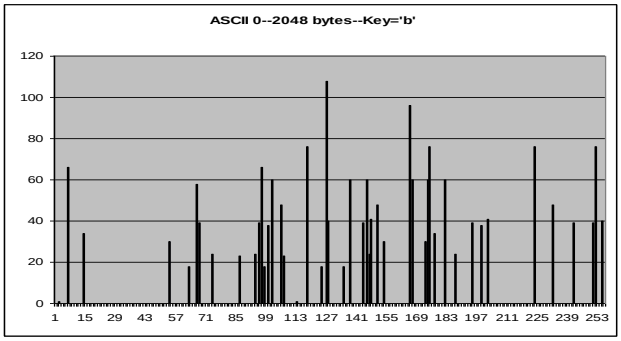

Fig-2: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '0' and Key='b'.

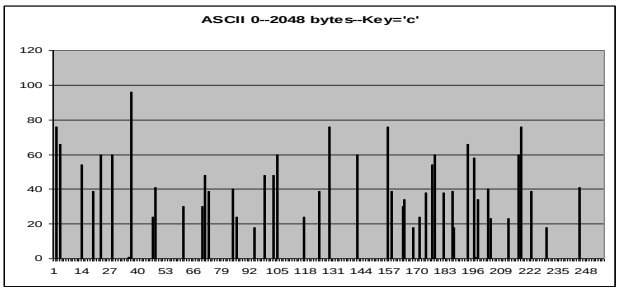

Fig-3: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '0' and Key='c'.

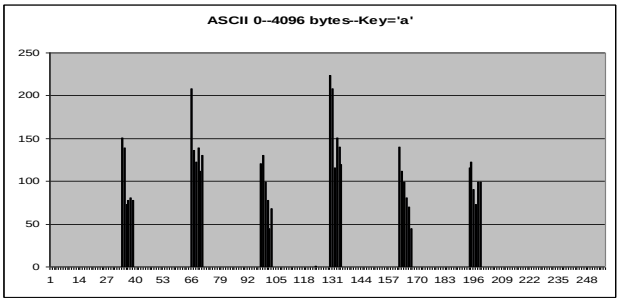

Fig-4: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '0' and Key='a'.

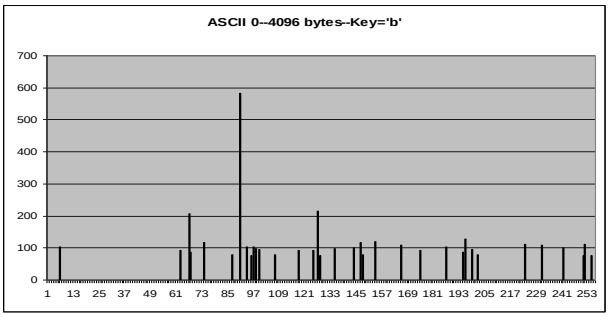

Fig-5: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '0' and Key='b'.

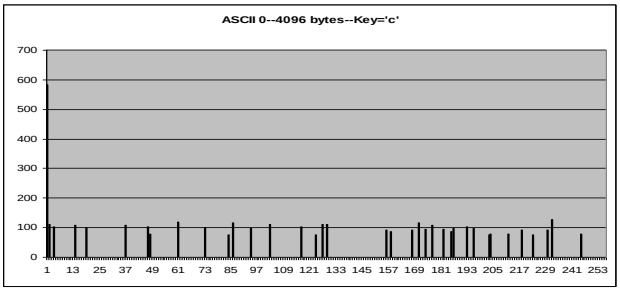

Fig-6: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '0' and Key='c'.

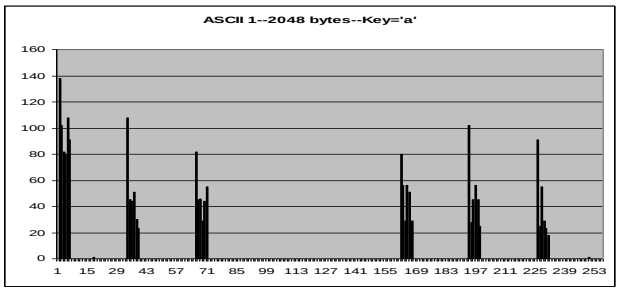

Fig-7: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '1' and Key='a'.

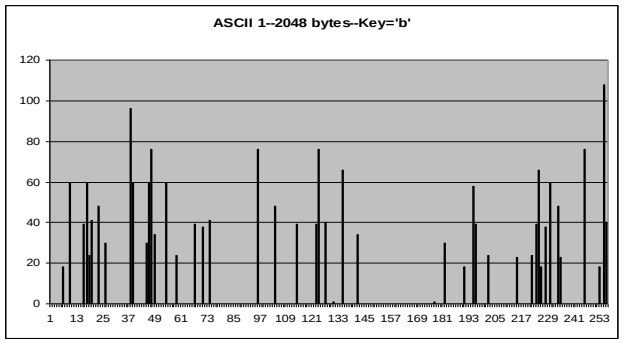

Fig-8: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '1' and Key='b'.

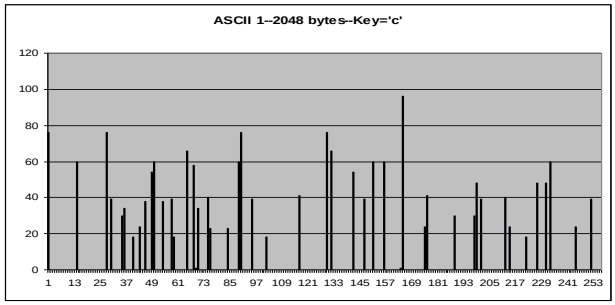

Fig-9: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '1' and Key='c'.

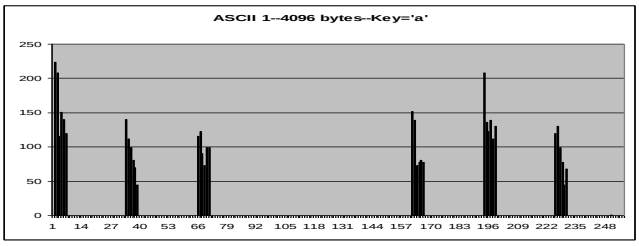

Fig-10: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '1' and Key='a'.

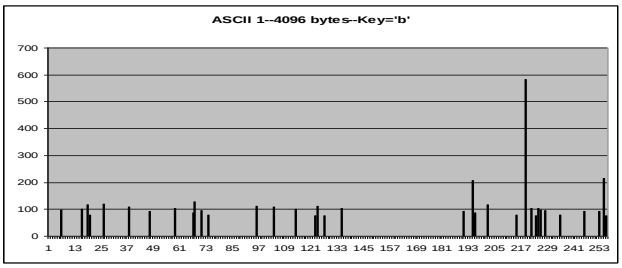

Fig-11: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '1' and Key='b'.

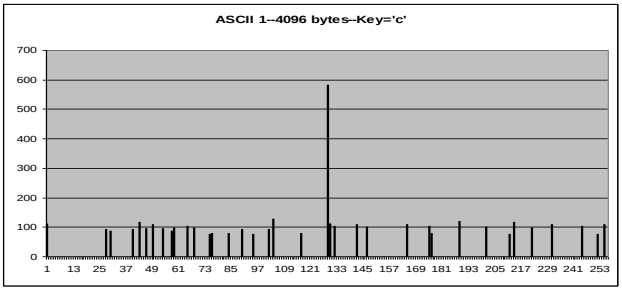

Fig-12: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '1' and Key='c'.

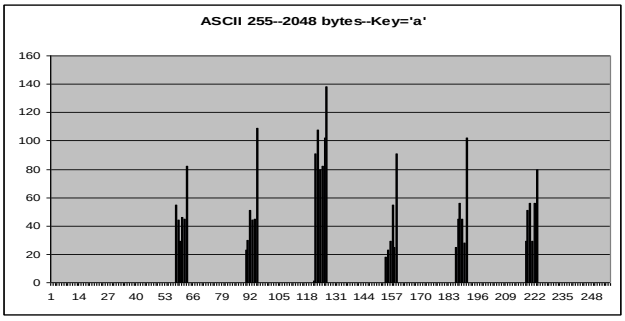

Fig-13: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '255' and Key='a'.

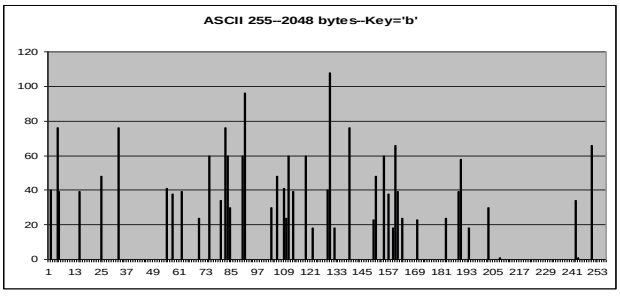

Fig-14: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '255' and Key='b'.

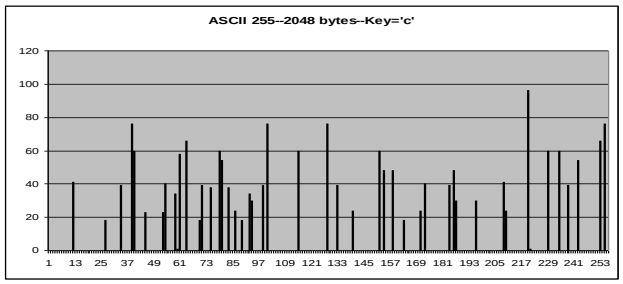

Fig-15: Frequency Spectral analyses of Plain Text file containing 2048 ASCII '255' and Key='c'.

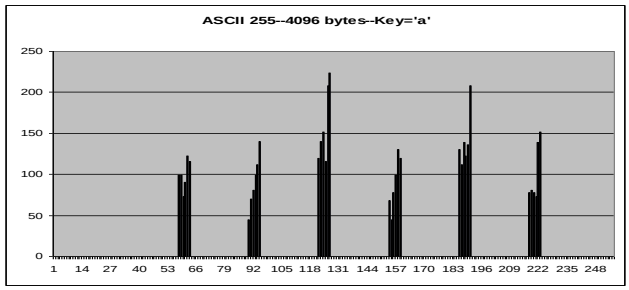

Fig-16: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '255' and Key='a'.

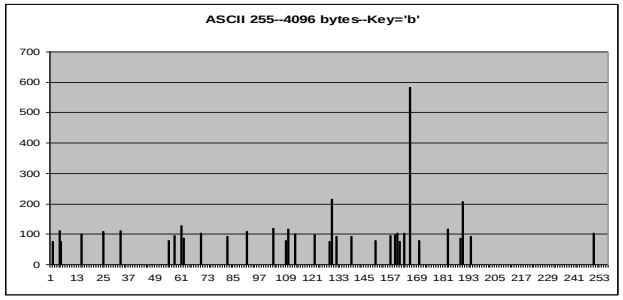

Fig-17: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '255' and Key='b'.

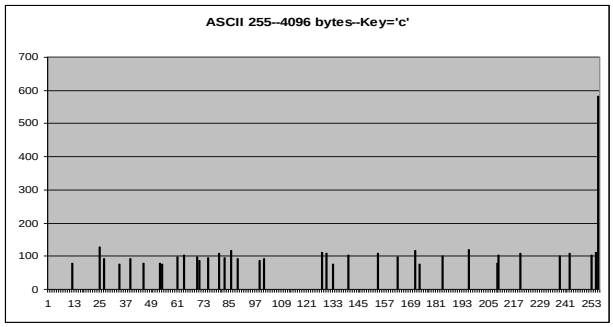

Fig-18: Frequency Spectral analyses of Plain Text file containing 4096 ASCII '255' and Key='c'.

Table 1: A Plain Text file contains a paragraph and its encrypted file

St. Xavier's is 152 years old. Proud of our long standing heritage and invested in our timeless tradition, we attest our motto of upholding the illustrious legacy of this esteemed institution, our commitment to creativity and exacting standards; our affiliation to the onerous task of providing liberal education and quality opportunities to students to develop to full potential. We are conscious of our rich history of pioneering movements and producing "men and women for others".	4&{0ntî"±¾-ªÄðoˆ©DÃ'ì† Ôñ .;¤@o£f%¾N cLuà^Ã"µ¹.?0¥àe´?pŽ:770õL+ ¿z- CÔ¦-0•®_zõìˆð°M>¤ôz0½ià-Ø àèlªØ»ÆÌêH;– F°òz›KÂàIØ³ÈÅÞ#°»4n£‰P[B"#áBæôQ□ Ê¾åÂ)Ùà#Pð›©J ÈÖÛ©àž>ƒ©"ÈX©…,n®;ñeÐ ‹¹mK0æ-$õ¥ÅW/lÁè"çäu-z¿l?s · ß´-P•èŸÔ]ÿ- b'ıÍz4>ú½)aøòŒ?€ïò,,`nzú4ô • J,  "ø±Ž„'°Lôî43/ævôcÑzón±V¾- VÚ9…h,#¹XÀ'R • Ý±#:#æfA± ø,;ÁRÒ7¡Ñbªh ÝÂr¹‰t‡z×…d Ÿˆòb7Ïˆà"" _È?,ìù-Põµ=/z • ¬z Rª˜-ô1Z0 ˆÝzH • ÐèlØ0;åÌYJz³±lozã+`¶ø 'jÕ'îèÛäÞf& `e;îˆ±ca"öBûˆå¦°±D7 • ãL± 5Æ[B{A'#3BùÊfcÐýóA ÐmÀ zÙhÈ

4. Conclusion and Future Scope

The encrypted text cannot be decrypted without knowing the exact initial random matrix. The size of random matrix taken is 16 x 16. The numbers in 16x16 may be arranged in 256! Ways. To complete the whole process we choose 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 spectral analysis shows that our 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.

Acknowledgment

We are very much grateful to the Department of Computer Science to give us this opportunity to work on symmetric key Cryptography. A.N. 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.

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 Jull 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 LasVegas 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 to be 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] Cryptography and Network Security, William Stallings, Prectice Hall of India.

Asoke Nath is the Associate Professor in Department of Computer Science. Apart from his teaching assignment he is involved with various research work in Cryptography, Steganography, Green Computing, E-learning. He has presented papers and invited tutorials in different International and National conferences in India and in abroad.

Prabal Banerjee Pursuing Bachelor of Science (Computer Science Honours) at St. Xavier's College(Autonomous), Kolkata. Presently involved in research work in Bit level encryption methods .