ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Bit Level Symmetric Key Encryption Algorithm (BLSKEA-1) Version-1

Asoke Nath, Madhumita Santra, Supriya Maji, Kanij Fatema Aleya

Associate Professor, Dept. of Computer Science, St. Xavier's College(Autonomous), Kolkata, India

M.Sc. Student, Dept. of Computer Science, St. Xavier's College (Autonomous), Kolkata, India

M.Sc. Student, Dept. of Computer Science, St. Xavier's College (Autonomous), Kolkata, India

M.Sc. Student, Dept. of Computer Science, St. Xavier's College (Autonomous), Kolkata, India

ABSTRACT: The present method deals with bit level encryption and decryption method. Nath et al(2014) already introduced bit level encryption method using feedback. But in the present paper the authors have used some simple but very effective bit level encryption method. The plain text is initially converted to bits and after that bit-wise complement is done on some random prime positions. The entire bit stream is reversed and again applied bit-complement operation in some random prime position. The bit complement is followed by bit-wise XOR operation and then the modified bit streams placed in a 2-dimesional array and perform some bit operations such as leftshift, upshift, diagonal shift, cycling, rightshift number of times to make the bit patterns random. The bit operations are performed number of times and finally bits were converted to bytes and transferred to some output file. The results show that the present method is very much effective to encrypt password, sms of any other confidential message.

KEYWORDS: Bit level encryption; differential attack; brute force attack; leftshift; rightshift

I. INTRODUCTION

The last two decades there was a tremendous explorations in internet technologies both in hardware level and as well as in software level now it is a great challenge to send any confidential data in raw from one computer to another computer or from one user to another user. Data security is now a very important issue in data communication and network. In the old days most of the data were kept in manual registers or files and that is why data were fully secured because the data were not available in internet. But now-a-days almost all data are available in internet and hence those who are hackers they can intercept those data and can do any kind of disasters also. Moreover the hackers also try to extract important personal data from any social networks and can misutilize those data. It is now almost essential to send any kind of confidential data in encrypted form. The last 2 decades the subject called cryptography is now a very important and relevant. Cryptography is the subject which deals with how one can modify a readable text or document or any object to unreadable text or documents or some other objects. There are some standard cryptography algorithms such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), Rivest Shamir Addlemen(RSA), Elliptic Curve Cryptography(ECC) etc. Many researchers also modified or developed some revised or new methods also. Nath et al already developed several cryptographic methods such as MSA, DJSA, TTJSA, Bit Level Encryption Standard, Modern Encryption Standard, Ultra Encryption standard, Multi way feedback encryption standard, 2 dimensional multi way feedback encryption standard, 3-dimensional multi way feedback encryption standard. Recently Nath et al developed some bit level encryption algorithms which are not only more complex than standard cryptography algorithms but also the methods are very safe because the hackers will not be able to decrypt the cipher text without knowing the key and also the exact method. In the present method the authors have introduced a pure bit level encryption method. In the present method the authors have implemented some bit wise operations. To encrypt any text/file the user has to input some secret key. From the given secret key the program will calculate some important parameters such as randomization number, encryption number etc. The given text will be converted to bits and then start to complement the bits available in some random prime position. After taking complement of bits the entire bits will be reversed and then again complements of bits are taken in random prime positions. After completion of complement operation the Bit-wise XOR operation is done. After performing Bit-wise XOR then entire modified bits were taken in a 2-dimensional array and then perform some simple bit operations such as leftshift, upshift, diagonal shift, cycling, rightshift in number of ways to make the entire bit patterns almost random. The whole operation performed number of times to make the final encrypted text totally random. The authors applied this method on some trivial patterns such as a text which contains all ASCII ‘1’ or ASCII ‘2’ or ASCII ‘255’ etc. Generally any standard method will generate cipher text where same pattern may be repeated. However, the present method applied on the above patterns but the outputs are totally unpredictable. Since the proposed method based on bit levels so therefore, some standard attacks such as brute force attack, known plain text attack, differential attack are not applicable here. The proposed method may be applied in mobile data encryption such as encryption of One Time Pass word(OTP), in android mobile this method may be used to encrypt any kind of data.

II. Randomization Number Generation

Here an example is shown how to generate the randomization number (round) from the key which is entered by the user. Round must be between 11and 31. For randomization number generation one has to follow the following steps:

Step1: User will input some key-text such as “AB”.
Step2: From given key calculate length of the key.
Step3: Retrieve the character from the key until the length of the key is 0.
Step 4: Multiply the ASCII value of the character of key with its next ASCII character and store it say r.
Step 5: ran=r%31
Step 6: if ran < 10 then ran= 11
For example:-

Let the key is”AB”.

Now we calculate length of the key=”AB”.
Here Length of the key (n) =2
r=65*66+66*67 =8712
So number of round (ran) =r%31=8712%31=1
Which is less than 10 so ran=11.

III. Proposed Algorithm

A. *Algorithm For Function Encryption():*

Step-1: Start
Step- 2: Input the key.
Step- 3: Call the function bit_transformation(file) to convert the bytes of input file into bits
Step- 4: calculate the size of the bit pattern.
Step- 5: Calculate p=perfect square (file)
Step- 6: Create a 2d array with size=(p+1)*(p+1)
Step- 7: Calculate p1=size1-(p*p); p2=p*p;
Step-8: Create two 1d array sizes=p1+1 and a 1 d array with size p2+1 // arr1[p1+1]; arr2[p1+1]; arr3[p2+1];
Step-9: Call random=randomization_no(key)
Step-10: Create a 1d array with size=size1+1 // arr4[size1+1]
Step-11: Call the function complement(file3,arr4,size1) to complement the Prime position bits of the entire bit stream.
Step-12: Call the function reverse(file3,arr4,size1) to reverse the entire bit patterns.
Step-13: Call function complement(file3,arr4,size1) to complement the Prime position bits of the entire bit stream.
Step-14: Call the function XOR(file3,arr4,size1) to perform bit-wise XOR operation bit-1 with bit-n and substitute in n-th bit and so on.
Step-15: Call the function XOR1(file3,arr4,size1) to perform bit-wise XOR operation bit-1 with bit-n and substitute in 1-st bit and so on.
Step-16: Repeat step-14 and step-15 till you exhaust all bits.

Step-17: Call the function array_representation(file3,p,arr,p1,arr1) to represent the bits in 2d array

Step-18Perform bit-wise leftshift(arr,p); // To shift all bits in each row by 1 unit on LHS

Step-19: Perform bit-wise upshift(arr,p); //To shift all bits in each column by 1 unit in upward direction

Step-20: Perform bit-wise diagonalshift(arr,p); // To exchange bits along two diagonals

Step-21: Perform bit-wise cycling(arr,p); // To perform circular shift of bits anti clock wise and then clock-wise in alternate periphery of the square

Step-22: Perform bit-wise rightshift(arr,p); // To shift all bits in each row by 1 unit on RHS

Step-23: Perform bit-wise downshift(arr,p); // To shift all bits in each column by 1 unit in downward direction

Step-24:Call the function twod_to_1d (arr,arr1,arr3,arr2,p,p1,p2,file3)

Step-25: Repeat step-11 to step-24 say ‘n’ number of times.

Step-26: Call the function bit_to_byte(file) to Convert bits to bytes and store it into a file.

Step 27: End.

B. *Algorithm For Function Decryption():*

Step-1: Start

Step- 2: Input the key.

Step- 3: Call the function bit_transformation(file) to convert the bytes of input file into bits

Step- 4: calculate the size of the bit pattern.

Step- 5: Calculate p=perfect square(file)

Step- 6: Create a 2d array with size=(p+1*(p+1) //arr[p+1][p+1]

Step- 7: Calculate p1=size1-(p*p); p2=p*p;

Step-8: Create two 1d array size=p1+1 and a 1 d array with size p2+1 // arr1[p1+1]; arr2[p1+1]; arr3[p2+1]

Step-9: Call random=randomization_no(key)

Step-10: Create a 1d array with size=size1+1 //arr4[size1+1];

Step-11: Call the function array_representation(file3,p,arr,p1,arr1) to represent the bits in 2d array

Step-12: Perform bit-wise upshift(arr,p);

Step-13: Perform bit-wise leftshift(arr,p);

Step-14: Perform bit-wise cycling(arr,p);

Step-15: Perform bit-wise diagonalshift(arr,p);

Step-16: Perform bit-wise downshift(arr,p);

Step-17: Perform bit-wise rightshift(arr,p);

Step-18: Call the function twod_to_1d (arr, arr1, arr3, arr2, p, p1, p2, file3).

Step-19: Call the function XOR1(file3,arr4,size1) to perform bit-wise XOR operation bit-1 with bit-n and substitute in 1-st bit and so on.

Step-20: Call the function XOR(file3,arr4,size1) to perform bit-wise XOR operation bit-1 with bit-n and substitute in n-th bit and so on.

Step-21: Repeat step-14 and step-15 till you exhaust all bits.

Step-22: Call the function complement(file3,arr4,size1) to complement the Prime position bits of the entire bit stream

Step-23: Call the function reverse(file3,arr4,size1) to reverse the entire bit patterns.

Step-24: Again call the function complement(file3,arr4,size1) to complement the Prime position bits of the entire bit stream

Step-25: Repeat step-11 to step-24 say ‘n’ number of times.

Step-26: Call the function bit_to_byte(file) to Convert bits to bytes and store it into a file.

Step 27: End.

IV. RESULTS AND DISCUSSION

In the table given below some plain texts and the corresponding ASCII value of cipher text are shown. There are many instances where it was observed for the same key, almost similar plain texts, the cipher texts are totally different. So without knowing the secret text-key and the actual decryption process it is quite impossible for the intruder to generate the plain text from the cipher text. The present algorithm can even encrypt ASCII 0, ASCII 1, and ASCII 255 which normally impossible in standard encryption methods like DES, RSA etc.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Plain text	key	ASCII Number of Cipher text
1. 8 ASCII '1' + 1 ASCII '0'+ 8 ASCII '1'	A	112,211,6,9,74,167,56,62,235,163,36,19,90,55,252,181,129
2. 8 ASCII '1' + 1 ASCII '2'+ 8 ASCII '1'	A	117,114,44,33,250,167,60,127,203,171,44,18,90,54,253,181,137
3. 8 ASCII '1' + 1 ASCII '3'+ 8 ASCII '1'	A	60,28,184,149,136,191,58,126,46,131,172,84,73,11,244,131,16
4. 8 ASCII '0' + 1 ASCII '1'+ 8 ASCII '0'	A	147,116,53,26,224,82,82,235,42,198,34,5,45,175,221,31,230
5. 8 ASCII '0' + 1 ASCII '2'+ 8 ASCII '0'	A	223,187,139,134,34,74,80,171,239,230,170,66,62,147,213,41,119
6. 16 ASCII '0' + 1 ASCII '1'	A	210,212,169,169,144,72,32,234,91,167,134,86,55,144,206,97,54
7. 1 ASCII '1' + 16 ASCII '0'	A	214,56,64,72,255,70,239,228,201,98,214,75,172,86,140,229,33
8. 17 ASCII '0'	A	218,26,161,174,146,74,84,234,207,238,162,67,62,146,212,41,127
9. 17 ASCII '1'	A	57,189,146,189,56,191,62,63,14,139,164,85,73,10,245,131,24,
10. 16 ASCII '1' + 1 ASCII '0'	A	49,115,154,186,58,189,74,63,154,194,128,64,64,8,239,203,81
11. 1 ASCII '0' + 16 ASCII '1'	A	53,159,115,91,85,179,133,49,8,7,208,93,219,206,173,79,70

Table -1: Some Plain texts and ASCII code of Encrypted Texts

In above table the results show that the cipher texts ate totally unpredictable even though the Plain texts contain some trivial patterns. The present method shows cipher texts always different even if input plain contains all characters same. In Figures 1 to 8 the encrypted data and also plain text data are shown. The results show that the Cipher texts patterns are totally unpredictable. The hackers will not be able apply any kind of brute force method to find Plain Text without knowing secret key. The present may be used to encrypt confidential message such as password, key etc.

DOI: 10.15680/IJIRCCE.2015. 0311089

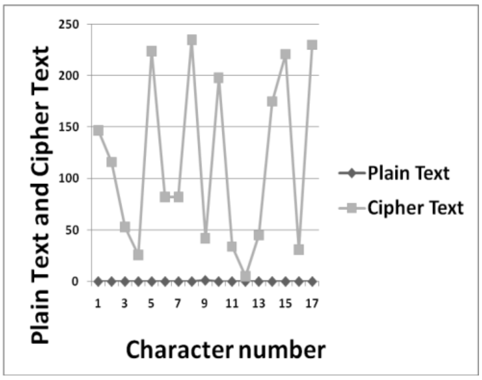

Fig-1: Encrypted 8 ASCII ‘0’+1 ASCII ‘1’+8 ASCII ‘0’

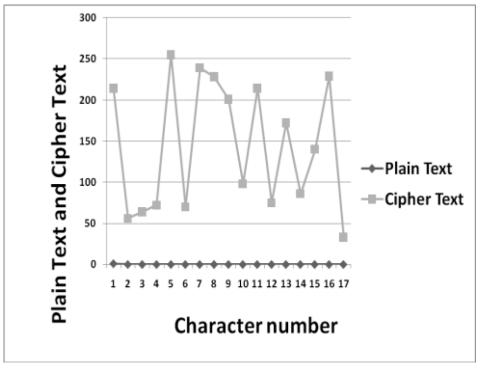

Fig-2: Encrypted 1 ASCII ‘1’+16 ASCII ‘0’

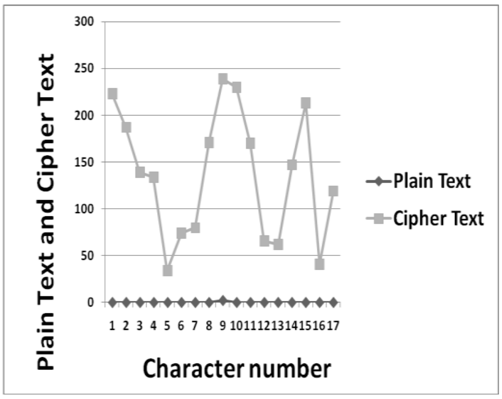

Fig-3: Encrypted 8 ASCII ‘0’+1 ASCII ‘2’+8 ASCII ‘0’

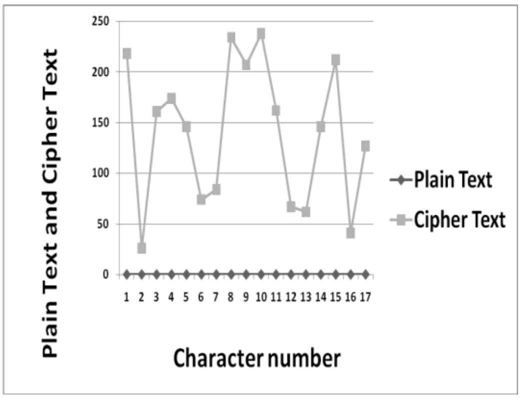

Fig-4: Encrypted 17 ASCII ‘0’

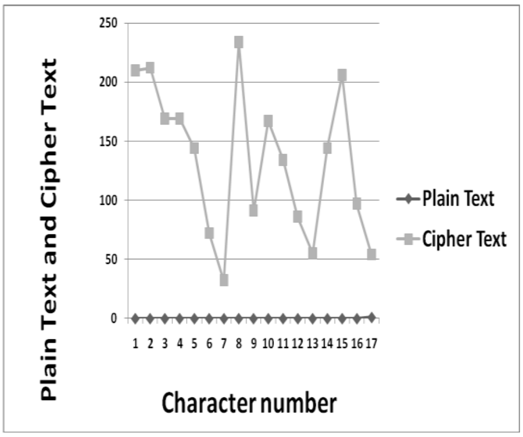

Fig-5: Encrypted 16 ASCII ‘0’+1 ASCII ‘1’

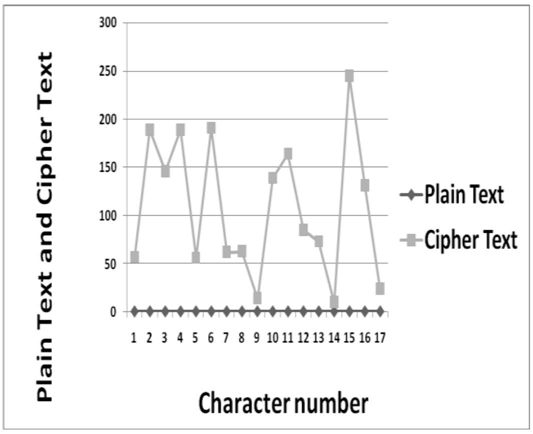

Fig-6: Encrypted 17 ASCII ‘1’

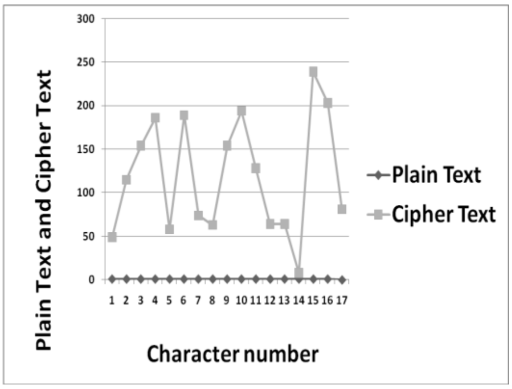

Fig-7: Encrypted 16 ASCII ‘1’+1 ASCII ‘0’

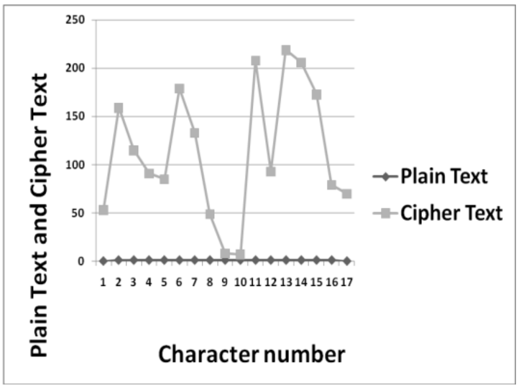

Fig-8: Encrypted 1 ASCII ‘0’+16 ASCII ’1’

V. CONCLUSION AND FUTURE SCOPE

The present method applied on different files like .txt, .png, .jpg, .ddl,.exe etc and results were quite satisfactory on any type of file. The user has to input some initial secret key for encryption and decryption. One cannot decrypt the encrypted text without knowing the initial secret key. Many standard method like leftshift(), rightshift(),downshift(), upshift(),cycling(),diagonalshift(),complement(),xor(), reverse() are applied to the plain text in the bit level so if two plain texts differ slightly, the encrypted text differ huge and so it is free from any type of brute force attack. To make this system more complex we used bit-wise operations. The authors are now working on 3-dimensional bit-wise columnar transposition method which will make the system much more secured.

REFERENCES

1. Symmetric Key Cryptography using Random Key generator: AsokeNath, SaimaGhosh, MeheboobAlamMallik: “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. A new Symmetric key Cryptography Algorithm using extended MSA method: DJSA symmetric key algorithm, DriptoChatterjee, JoyshreeNath, SuvadeepDasgupta and AsokeNath : Proceedings of IEEE International Conference on Communication Systems and Network Technologies, held at SMVDU(Jammu) 03-06 June,2011, Page-89-94(2011).
3. New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm:NeerajKhanna, Joel James,JoyshreeNath, SayantanChakraborty, AmlanChakrabarti and AsokeNath : Proceedings of IEEE CSNT-2011 held at SMVDU(Jammu) 03-06 June 2011, Page 125-130(2011).
4. Symmetric key Cryptography using modified DJSSA symmetric key algorithm, DriptoChatterjee, JoyshreeNath, Sankar Das, ShalabhAgarwal and AsokeNath, Proceedings of International conference Worldcomp 2011 held at LasVegas 18-21 July 2011, Page-306-311, Vol-1(2011).
5. 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).
6. Ultra Encryption Standard(UES) Version-I: Symmetric Key Cryptosystem using generalized modified Vernam Cipher method, Permutation method and Columnar Transposition method, Satyaki Roy, NavajitMaitra, JoyshreeNath,ShalabhAgarwal and AsokeNath, 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)
7. Advanced Digital Steganography using Encrypted Secret Message and Encrypted Embedded Cover File, JoyshreeNath, SaimaGhosh and AsokeNath, International Journal of Computer Applications(IJCA 0975-8887), Vol 46, No-14, May ,(2012).
8. Ultra Encryption Standard(UES) Version-II: Symmetric key Cryptosystem using generalized modified vernam cipher method, permutation method, colum,nar transposition method and TTJSA method, Satyaki Roy, NavajitMoitra, JoyshreeNath, ShalabhAgarwal and AsokeNath, Proceedings of International Conference Worldcomp 2012 held at Las Vegas, USA, FCS-12, Page-97 – 104(2012).
9. Ultra Encryption Standard(UES) Version-IV: New Symmetric Key Cryptosystem with bit-level columnar Transposition and Reshuffling of Bits, Satyaki Roy, NavajitMaitra, JoyshreeNath, ShalabhAgarwal and AsokeNath, International Journal of Computer Applications(IJCA)(0975-8887) USA Volume 51-No.1.,Aug, Page. 28-35(2012).
10. Bit Level Encryption Standard (BLES) : Version-I, NeerajKhanna, DriptoChatterjee, JoyshreeNath and AsokeNath, International Journal of Computer Applications(IJCA)(0975-8887) USA Volume 52-No.2.,Aug, Page.41-46(2012).
11. Bit Level Encryption Standard(BLES) : Versiob-II, GauravBhadra, Tanya Bala, SamaikBanik, JoyshreeNath and AsokeNath, Proceedings of IEEE International Conference WICT-2012 held at IIITM-K, Trivandrum Oct 30 to Nov 1, 2012, Page No. 121- 127(2012).

12. Modern Encryption Standard(MES) version-I : An Advanced Cryptographic Method, SomdipDey, AsokeNath, Proceedings of IEEE International Conference WICT- 2012 held at IIITM-K, Trivandrum Oct 30 to Nov 1, 2012, Page No. 242-247(2012).

13. Bit Level Generalized Modified Vernam Cipher Method with Feedback, International Journal of Advanced Computer Research (ISSN(print):2249- 7277 ISSN(online): 2277-7970), Volume-2, Number-4 Issue-6, Page-24-30, Dec(2012).

14. Bit Level Encryption Standard (BLES): Ver-III, GauravBhadra, Tanya Bala, SamikBanik, JoyshreeNath, AsokeNath, Proceedings of International Conference Worldcomp 2013 held at Las Vegas, USA in Jul 22-25, 2013. Proceedings page 99-105(2013).

15. Advanced Symmetric Key Cryptosystem using Bit and Byte Level Encryption Methods with Feedback‖ dvanced Symmetric Key Cryptosystem using Bit and Byte Level Encryption methods with Feedback, Prabal Banerjee, AsokeNath, Proceedings of InInternational Conference Worldcomp 2013 held at Las Vegas, USA in Jul 22-25, 2013.Proceedings Page 120-126(2013).

16. Multi Way Feedback Encryption Standard Ver-3(MWFES-3), AsokeNath, DebdeepBasu, Ankita Bose, SaptarshiChatterjee, SurajitBhowmikpublished in IEEE conference proceedings: WICT-2013 held at Hanoi in Dec 14-18(2013), page 318-325(2013).

17. Multi Way Feedback Encryption Standard Ver-2(MWFES-2), AsokeNath, DebdeepBasu, Ankita Bose, SaptarshiChatterjee, SurojitBhowmik , International Journal of Advanced Computer Research(IJACR), Vol 3, Number-1, Issue-13, Page-29-35, Dec(2013).

18. AnkitaBasu, DebdeepBasu, SaptarshiChatterjee, AsokeNath, SurajitBhowmik, Bit Level Multi Way Feedback Encryption Standard Ver-1(BLMWFES-1), published in Proceedings of IEEE conference CSNT-2014 held at Bhopal, page-601-605, April 7(2014).page-793-799, April 7(2014).

19. AsokeNath , Bit level Multi Way Feedback Encryption Standard Ver- 2(BLMWFES-2) , proceedings of International IEEE conference Advanced Communication, Control & Computing Technologies(ICACCCT) 2014 held at Syed Ammal Engineering College, Page 1702-1707(2014).

20. ArijitGhosh, PrabhakarChakraborty, AsokeNath, ShamindraParui, ―3d Multi Way Feedback Encryption Standard Version I(3dMWFES1)‖,International Journal of Advance Research in Computer Science and Management Studies, ISSN:2321-7782(Online), Vol 2, Issue 10,Oct, Page:206-218(2014).

21. ArijitGhosh, PrabhakarChakraborty, AsokeNath, "3d Multi Way Feedback Encryption Standard Version II(3dMWFES-II)", International Journal of Computer Science and Information Technologies(IJCSIT), Vol.6(3), Page 2990-2997(June 2015).

22. AsokeNath, Ranjini Mukherjee, Dona Sarkar, ChaitaliPatra, "2-Dimensional Multi Way Feedback Encryption Standard Version-1(2dMWFES-1)", International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol-3, Issue 6, Page 5024-5033(30-th June 2015).

BIOGRAPHY

Dr. Asoke Nath is Associate Professor in the Department of Computer Science, St. Xavier's College(Autonomous), Kolkata, India. His field of research areas comprises of Cryptography and Network security, Steganography, Green Computing and Green Technology, e-learning, MOOCs, Big Data Analytics, Mathematical Modelling of Social Networks etc. Dr. Nath published more than **158** Research papers in Journals and conference proceedings.

Madhumita Santra is a student of M.Sc. Computer Science, St. Xavier's College(Autonomous),Kolkata, India. Currently she is doing research work in field of Cryptography.

Supriya Maji is a student of M.Sc. Computer Science, St. Xavier's College(Autonomous),Kolkata, India. Currently she is doing research work in field of Cryptography.

Kanij Fatema Aleya is a student of M.Sc. Computer Science, St. Xavier's College(Autonomous),Kolkata, India. Currently she is doing research work in field of Cryptography.