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. 4, Issue 5, May 2016

3-Dimensional Bit Level Encryption Algorithm Ver-1 (3DBLEA -1)

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

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

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

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

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

ABSTRACT:The last one decade many symmetric as well as public key encryption algorithms have been developed by the researchers across the globe. The researchers developed several bit level encryption algorithms which are almost impossible to break without knowing the actual key and the actual procedure. These methods are quite complex to decrypt by using any kind of standard attacks such as Brute force attack, known plain text attack, statistical attack, differential attack etc. In the present paper the authors have introduced 3-Dimensional bit level encryption algorithm version-1(3DBLEA-1). The present method (3DBLEA-1) is based on several bit level operations and 3 dimensional columnar transposition method. The authors have introduced bit level transposition methods along 3 directions i.e. x, y and z-directions of bits. After applying bit level operations the authors used DNA sequence on it. The algorithm works on bit level so it is not possible for intruder to decrypt it without knowing the key as well as the exact method. By using this algorithm one can encrypt any file such as .txt file, .doc file, .jpg file, .exe file, .wav file, .pdf file or any other file. After encryption or decryption the original size of file will remain unaltered. A thorough investigation made on change in single bit in any position of the cipher text and it was found that decryption will not work. It is not possible to get back original plain text file if there is one change in bits in encrypted text. A thorough testing is done on various types of files and it was found that the 3DBLEA-1 is unbreakable.

KEYWORDS: Bit level encryption; differential attack; brute force attack; leftshift; rightshift, DNA sequence.

I. Introduction

In the present paper the authors have introduced a new symmetric key cryptographic method which is based on bit level columnar transposition method applied along three dimensions (axis) on plain text. Along with this the authors have implemented some bit wise operations and convert the bits into DNA sequence and randomize it. The process starts by converting the plain text characters into bits. The authors applied complement operation on the prime position plain text bits. After complement the entire bits will be reversed and 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, diagonalshift, cycling, rightshift in number of ways to make the entire bit patterns almost random. After that the authors have implemented the 3 dimensional transposition methods. The process starts by dividing the bits into a number of square matrices where the size of the matrix and the numbers of layers are balanced. For example, if the plain text has 3 characters we convert it into bits i.e. 24 bits then the size of the matrix will be 3X3 and the number of layers will be 6 and 2 additional bits will be left out as residual bits. The number of rows and columns and the number of layers are calculated by some simple mathematical calculations. The present encryption method can be applied multiple times to make the system fully secure. After finishing bit-level operations every 2 bits were converted to DNA sequence such as A(='00'), C(='01'), G(='10'), T(='11'). The DNA sequence was shuffled using some predefined shuffling functions. Then the DNA characters were converted into bits and then each 8-bit converted to byte and store in some output file. This output file corresponds to Cipher text file. To encrypt any text/file the user has to input some initial text_key. This initial text_key can contain any number of characters. This will be treated as the secret key. From this secret key the proposed algorithm will calculate some important parameters such as randomization

DOI: 10.15680/IJIRCCE.2016. 0405077

number, encryption number etc. And from this secret key the program select the keys for 3- dimensional transposition methods also. The authors applied this method on some trivial patterns such as a text which contains all ASCII '0' or 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.

II. LITERATURE SURVEY

Some bit level encryption algorithms that have been already developed are DNA based cryptography and BLSKEA version-1 that was already developed by Nath et al. Now the authors discussed about the previous versions below.

A. *DNA Based Cryptography:*

DNA cryptography is a relatively new paradigm that has attracted great interest in the field of information security. DNA coding technology is used to convert binary data to DNA strings. Since scientists found that binary computers have many physical limitations, especially in data storage and computation, they have concentrated on DNA computers and tried to implement this new science in the information security field. DNA cryptography is a new concept that needs many improvements. Although there are still problems with DNA cryptography, many scientists are trying to solve them because they believe that, with the characteristics of DNA computers they have more advantages than conventional cryptography. DNA coding technology is another concept in cryptography that is intended to encode binary data to a DNA strand and vice versa. Binary data can be encoded in DNA by using sequence of alphabet. It is known that DNA sequences contain four basic letters A, C, G,T : '00' →A, '01'→C, '10'→G, '11'→T
For example, a binary string like'01001011' is converted to 'CAGT'.
The cryptosystem is based on the vigenere cipher, which is a poly-alphabetic cipher. Poly-alphabetic ciphers are multi-substitution ciphers, which mean that each letter in the plain text is substituted in different forms. The main achievement of this study is identifying a DNA cryptosystem, which is new science in information security. But the vigenere ciphers have some problem. The first problem is that it uses the English alphabet so it is obvious that with frequency analysis we can guess the correct letter of the cipher text. But in this project all encryption process in bit level. In this project first the authors have done 3-dimensinal encryption process for n times. Then covert the bit level cipher text into DNA sequences that are the form of A, C, G and T form. Then perform some randomization operation to randomize the DNA sequence. Then the authors converted the DNA sequence into bits and then the bits are converted into byte form.

B. *Bit Level Symmetric Key Encryption Algorithm(BLSKEA-I) version-I:*

This 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.

This project is the extension of BLSKEA-1 version-1. In this project the authors also have done some randomization operation on plain text after converting the plain text from byte to bit. Then the authors have done some additional operation that is bit wise 3-dimensional that in row wise. Column wise and depth wise bit-level columnar transposition method which will make the system much more secured. The authors also add DNA cryptography in this project which makes the project more secured.

III. Encryption Number and Randomization Number Generation

A. *Proposed method for Calculation of encryption number:*

To describe the method let us choose some secret key=”ABCD”
Convert ASCII Codes of secret key to bits=01000001010000100100001101000100
Length of bits=32.
Calculate all the prime numbers from 1 to 32 as base.
The prime numbers are-2,3,5,7,11,13,17,19,23,29,31.
Calculates the position of 1’s as power
The positions of 1’s are 2,8,10,15,18,23,24,26,30.
Calculate the sum(s).
$s=2^2+3^8+5^{10}+7^{15}+11^{18}+13^{23}+17^{24}+19^{26}+23^{30}+29^2+31^8=71094350559944271526480840722677417842744$.
Sum all prime position number the sum is 43
Encryption number=43%89=43. [Note: maximum encryption number taken here=89]

B. *Proposed method for Calculation of randomization number:*

To generate randomization number calculates the sum of all digits.
Sum of all digits in s
=7+1+0+9+4+3+5+0+5+5+9+9+4+4+2+7+1+5+2+6+4+8+0+8+4+0+7+2+2+6+7+7+4+1+7+8+4+2+7+4+4 =184.
Randomization number=184%29=10 [Note: maximum randomization number taken here=29]

IV. Calculation of Column, Row and Depth Number for Transposition Method

Let, plain text=“AAAA”.
Convert the plain text into bits.
Size of plain text=32.
Let n= 32/2=16
Calculate all perfect square from 1 to 15.
Here the perfect squares are 4 and 9.
Then store the perfect squares into an array and take the middle position of array.
Here we take 4.
So the column=2 and row=2.
Depth =plain_text size/ (column*row)=32/4=8.
So, the depth or layer is 8.

V. Column, Row and Depth Key Selection For Transposition methods

In encryption number generation we have calculated the sum using base and power.
From this sum get the key for columnar, row wise and depth wise transposition method.
Let the sum is=71094350559944271526480840722677417842744.

(i) First from this sum take the number from 1 to column/row number for columnar and row wise key randomization or 1 to depth number for depth key randomization. For example, if the column and row number is 5 then the key for column and row wise transposition method is: 1, 4, 3, 5, 2.We get this key from the sum.

(ii) Then add the number 1 to column/row/depth which is not present in the sum. For example, if the number of depth is 12 then we choose the number from sum- 7, 1, 9, 4, 3, 5, 2, 6, 8, but in the sum there is no 10, 11 and 12 so we add these three numbers with the key. Then the key will be-7, 1, 9, 4, 3, 5, 2, 6, 8, 10, 11
Resuffle the key using some shifting operation: Leftshift, Upshift, Diagonalshift, Cycling, Rightshift, Downshift

(iii) After that we get the key for 3D columnar transposition method i.e. column wise, row wise and depth wise.

VI. PROPOSED ALGORITHM

A. *Algorithm For Function Encryption():*

```
Step-1: Start
Step-2: Input the key.
Step-3: Convert the bytes of input file into bits
Step-4: Calculate the size of the bit pattern.
Step-5: Complement the Prime position bits of the entire bit stream.
Step-6: Reverse the entire bit patterns.
Step-7: Again complement the Prime position bits of the entire bit stream.
Step-8: Perform bit-wise XOR operation on bit-1 with bit-n and substitute in n-th bit and so on.
Step-9: Perform bit-wise XOR operation bit-1 with bit-n and substitute in 1-st bit and so on.
Step-10: Repeat step-8 and step-9 till you exhaust all bits.
Step-11: Store the bits into 2 dimensional array (nXn) and residual bits into a 1-dimensional array and perform the following shifting operations on the 2 dimensional arrays.
Step-12: Perform bit-wise leftshift(); // To shift all bits in each row by 1 unit on LHS
Step-13: Perform bit-wise upshift(); //To shift all bits in each column by 1 unit in upward direction
Step-14: Perform bit-wise diagonalshift(); // To exchange bits along two diagonals
Step-15: Perform bit-wise cycling (); // To perform circular shift of bits anti clock wise and then clock-wise in alternate periphery of the square
Step-16: Perform bit-wise rightshift(); // To shift all bits in each row by 1 unit on RHS
Step-17: Perform bit-wise downshift (); // To shift all bits in each column by 1 unit in downward direction
Step-18: After that shift the residual bits into 2-D array and shift the same number of bits from 2-d array to the residual array.
Step-19: Repeat step-12 to step-18 say ‘n’ number of times (according to randomization number).
Step-20: Calculate the column row and depth for 3- dimensional transposition methods.
Step-21: Select the keys for columnar transposition, row-wise transposition and depth wise transposition method from the secret key.
Step-22:  Perform bit wise columnar transposition method on the bits.
Step-23: Perform row wise transposition method on the bits.
Step-24:Perform depth wise transposition method on the bits.
Step-25: Repeat step-5 to step-24 say ‘n’ number of times (according to round number).
Step-26: Take 2 bits at a time and convert it into DNA sequences :  00 →A, 01 →C, 10→G, 11→T
Step-27: Randomize it using following shifting operations:
Leftshift(), Upshift(), Diagonalshift(), Cycling() ,Rightshift(), Downshift()
Step-28: Convert the DNA sequence into bits.
Step-29: Convert bits to bytes and store it into a file as cipher text.
Step 30: End.
```

B. *Algorithm For Function Decryption():*

```
Step-1: Start
Step-2: Input the key.
Step-3: Convert the bytes of input file into bits.
Step-4: Calculate the size of the bit pattern.
Step-5: Take 2 bits at a time and convert it into DNA sequences: 00→A, 01→C, 10→G, 11→T
Step-6: Randomize it using following shifting operations:
        upshift(),  leftshift(), Cycling(), Diagonalshift(), downshift(), rightshift()
Step-7: Convert the DNA sequence into bits.
Step-8: Calculate the column row and depth for 3- dimensional transposition methods.
Step-9: Select the keys for columnar transposition, row-wise transposition and depth wise transposition method from the secret key.
```

Step-10: Perform depth wise transposition method on the bits.
Step-11: Perform row wise transposition method on the bits.
Step-12: Perform bit wise columnar transposition method on the bits.
Step-13: Store the bits into 2 dimensional array (nXn) and residual bits into a 1-dimensional array and perform the following shifting operations on the 2 dimensional arrays.
Step-14: Perform bit-wise upshift();
Step-15: Perform bit-wise leftshift();
Step-16: Perform bit-wise cycling();
Step-17: Perform bit-wise diagonalshift();
Step-18: Perform bit-wise downshift();
Step-19: Perform bit-wise rightshift();
Step-20: Shift the residual bits into 2-D array and shift the same number of bits from 2-d array to the residual array.
Step-21: Repeat step-14 to step-20 say ‘n’ number of times (according to randomization number).
Step-22: Perform bit-wise XOR operation bit-1 with bit-n and substitute in 1-st bit and so on.
Step-23: Perform bit-wise XOR operation on bit-1 with bit-n and substitute in n-th bit and so on.
Step-24: Repeat step-22and step-23 till you exhaust all bits.
Step-25: Complement the Prime position bits of the entire bit stream.
Step-26: Reverse the entire bit patterns.
Step-27: Again complement the Prime position bits of the entire bit stream.
Step-28: Repeat step-8 to step-27 say ‘n’ number of times (according to round number).
Step 29: Convert bits to bytes and store it into a file as cipher text.
Step-30: End.

VII. 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.

Plain text	key	ASCII Number of Cipher text
1. 8 ASCII '1' + 1 ASCII '0'+ 8 ASCII '1'	A	160,0,189,160,207,122,66,181,113,240,148,147,114,37,97,53,38
2. 8 ASCII '1' + 1 ASCII '2'+ 8 ASCII '1'	A	50,232,84,241,49,152,76,189,23,11,168,177,109,242,72,155,82
3. 8 ASCII '1' + 1 ASCII '3'+ 8 ASCII '1'	A	143,73,64,88,92,57,115,98,218,127,168,40,158,185,84,113,2
4. 8 ASCII '0' + 1 ASCII '1'+ 8 ASCII '0'	A	234,255,34,72,109,7,120,50,163,159,205,235,13,173,175,121,39
5. 8 ASCII '0' + 1 ASCII '2'+ 8 ASCII '0'	A	197,182,223,176,254,68,73,229,8,16,241,80,225,49,154,61,3,
6. 16 ASCII '0' + 1 ASCII '1'	A	91,128,173,108,92,182,78,98,203,247,141,1,149,31,56,32,99,
7. 1 ASCII '1' + 16 ASCII '0'	A	170,98,174,1,108,148,194,108,118,172,27,105,12,156,107,204,165
8. 17 ASCII '0'	A	87,94,54,225,0,166,71,237,110,235,205,114,254,230,179,147,119
9. 17 ASCII '1'	A	29,161,169,9,162,219,125,106,188,132,148,10,129,110,125,223, 118
10. 16 ASCII '1' + 1 ASCII '0'	A	17,127,50,132,254,203,116,229,25,152,212,121,234,151,246,108, 98
11. 1 ASCII '0' + 16 ASCII '1'	A	91,128,173,108,92,182,78,98,203,247,141,1,149,31,56,32,99

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

PLAIN TEXT	KEY	CIPHER TEXT(ASCII VALUE)	CIPHER TEXT(CHAR)
AAAA	AAAA	14,33,98,174	♫!b«
AAAB	AAAA	26,196,178,208	→─▓╨
BAAA	AAAA	109,228,51,4	mΣ3♦
AAA	AAAA	114,134,237	råφ
AAB	AAAA	126,247,109	~≈m
BBA	AAAA	149,182,237	ò╢φ
BAA	AAAA	149,215,237	ò╫φ
AA	AAAA	234,203	Ω╦
AB	AAAA	72,216	H╪
BA	AAAA	200,50	╚2
A	AAAA	33	!
B	AAAA	9	TAB
C	AAAA	1	☺

Table-2: Input characters vs. corresponding cipher text.

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.

VIII. 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()and3-Dimensional columnar transposition method are applied in row wise, column wise and depth wise 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 furthercomplex bit-wise operations were used. Every application has its merits and demerits. The present method has covered almost all requirements. Further requirements and improvements can easily be done since coding is mainly structured or modular in nature.This method can be extended using DNA computing.

REFERENCES

1. AsokeNath, SaimaGhosh, MeheboobAlamMallik, “Symmetric Key Cryptography using Random Key generator:”,“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. DriptoChatterjee, JoyshreeNath, SuvadeepDasgupta and AsokeNath, “A new Symmetric key Cryptography Algorithm using extended MSA method: DJSA symmetric key algorithm”, Proceedings of IEEE International Conference on Communication Systems and Network Technologies, held at SMVDU(Jammu) 03-06 June,2011, Page-89-94(2011).
3. NeerajKhanna, Joel James,JoyshreeNath, SayantanChakraborty, AmlanChakrabarti and AsokeNath , “New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm”, Proceedings of IEEE CSNT-2011 held at SMVDU(Jammu) 03-06 June 2011, Page 125-130(2011).
4. DriptoChatterjee, JoyshreeNath, Sankar Das, ShalabhAgarwal and AsokeNath, “Symmetric key Cryptography using modified DJSSA symmetric key algorithm”, 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. Satyaki Roy, NavajitMaitra, JoyshreeNath,ShalabhAgarwal and AsokeNath, “Ultra Encryption Standard(UES) Version-I: Symmetric Key Cryptosystem using generalized modified Vernam Cipher method, Permutation method and Columnar Transposition method”, Proceedings of

DOI: 10.15680/IJIRCCE.2016. 0405077

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. JoyshreeNath, SaimaGhosh and AsokeNath, "Advanced Digital Steganography using Encrypted Secret Message and Encrypted Embedded Cover File", International Journal of Computer Applications(IJCA 0975-8887), Vol 46, No-14, May ,(2012).
8. Satyaki Roy, NavajitMoitra, JoyshreeNath, ShalabhAgarwal and AsokeNath, "Ultra Encryption Standard(UES) Version-II: Symmetric key Cryptosystem using generalized modified vernam cipher method, permutation method, colum,nar transposition method and TTJSA method", Proceedings of International Conference Worldcomp 2012 held at Las Vegas, USA, FCS-12, Page-97 – 104(2012).
9. Satyaki Roy, NavajitMaitra, JoyshreeNath, ShalabhAgarwal and AsokeNath, "Ultra Encryption Standard(UES) Version-IV: New Symmetric Key Cryptosystem with bit-level columnar Transposition and Reshuffling of Bits", International Journal of Computer Applications(IJCA)(0975-8887) USA Volume 51-No.1.,Aug, Page. 28-35(2012).
10. NeerajKhanna, DriptoChatterjee, JoyshreeNath and AsokeNath, "Bit Level Encryption Standard (BLES) : Version-I", International Journal of Computer Applications(IJCA)(0975-8887) USA Volume 52-No.2.,Aug, Page.41-46(2012).
11. GauravBhadra, Tanya Bala, SamaikBanik, JoyshreeNath and AsokeNath "Bit Level Encryption Standard(BLES) : Versiob-II", Proceedings of IEEE International Conference WICT-2012 held at IIITM-K, Trivandrum Oct 30 to Nov 1, 2012, Page No. 121- 127(2012).
12. SomdipDey, AsokeNath , "Modern Encryption Standard(MES) version-I : An Advanced Cryptographic Method", 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. GauravBhadra, Tanya Bala, SamikBanik, JoyshreeNath, AsokeNath, "Bit Level Encryption Standard (BLES): Ver-III",Proceedings of International Conference Worldcomp 2013 held at Las Vegas, USA in Jul 22-25, 2013. Proceedings page 99-105(2013).
15. Prabal Banerjee, AsokeNath, "Advanced Symmetric Key Cryptosystem using Bit and Byte Level Encryption Methods with Feedback‖ advanced Symmetric Key Cryptosystem using Bit and Byte Level Encryption methods with Feedback", Proceedings of InInternational Conference Worldcomp 2013 held at Las Vegas, USA in Jul 22-25, 2013.Proceedings Page 120-126(2013).
16. AsokeNath, DebdeepBasu, Ankita Bose, SaptarshiChatterjee, SurajitBhowmik, " Multi Way Feedback Encryption Standard Ver-3(MWFES-3)",published in IEEE conference proceedings: WICT-2013 held at Hanoi in Dec 14-18(2013), page 318-325(2013).
17. AsokeNath, DebdeepBasu, Ankita Bose, SaptarshiChatterjee, SurojitBhowmik , "Multi Way Feedback Encryption Standard Ver-2(MWFES-2)",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).
23. Mohammadreza Najaftorkaman, Nazanin Sadat Kazazi, "A Method To Encrypt Information With DNA-Based Cryptography", published in International Journal of Cyber-Security and Digital Forensics (IJCSDF) 4(3): 417-426, 417 The Society of Digital Information and Wireless Communications, 2015 (ISSN: 2305-0012), Page No. 418-421.

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 **185** 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.

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. 4, Issue 5, May 2016

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.