International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-3 Issue-11 September-2013

Modern Encryption Standard version IV: (MES-IV)

Asoke Nath[1], Payel Pal[2]

Abstract

In the present paper the authors have introduced a new cryptographic method named as Modern Encryption System Standard version IV which is basically a symmetric key cryptographic method. Here the authors have used three different type of cryptographic methods .Those method are columnar transposition method, bit level generalized vernam cipher method with feedback and bit wise XOR operation. This system is the extension of MES-III and partly Bit level Encryption Standard (BLES) –II and III. BLES-I, II, III and MES-I, II, III developed by Nath et al where MES-I, II, III mostly based on mainly byte level encryption method. BLES-I, II and III are based on mostly bit level encryption methods. In the present MES-IV method authors have tried to combine different bit level encryption to make the entire encryption system more secured. The introduction of feedback in bit level generalized vernam cipher method prevents attacks such as differential attack or plain text attack. The random key generator has been used to construct the keypad for vernam cipher method. The present encryption method is free from common attacks and it is almost impossible to break the present method without knowing the exact key and the methods. This encryption method will be used to encrypt password, short messages, financial data etc.

Keywords

Plain text, Cipher text, Randomization, Columnar Transposition, Feedback

1. Introduction

Keeping secrets is not easy. In fact human tendency is such that when told that something is a secret and asked to keep it secret, people are actually quite eager to share that secret with everyone else.

Asoke Nath, Department of Computer Science, St. Xavier's College(Autonomous), Kolkata, India.
Payel Pal, Indira Gandhi National Open University,St. Xavier's College(Autonomous),Kolkata, India.

In the early days of serious computing there was not a great deal of emphasis on security, because the systems in those days were proprietary or closed. The chances of someone getting an access to the information being exchanged were not very high.

As the minicomputers and microcomputers evolved in the 1970s and 1980s, the issue of information security started to gain more prominence. However it was the internet, which changed the whole computing paradigm and brought a tremendous change in the way computers communicated with each other. The world of computers had suddenly become very open. Therefore, it is very important to know how we can make information exchange secure. The confidentiality and genuineness of data has now become a very important issue. To send any important information from one user to another user normally the people are using e-mail as their transmission media. But the message of the e-mail can be trapped by the hacker between sender and receiver provided it is in raw form. To get rid of 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 key cryptography.

The cryptography 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. In symmetric key we have to maintain only one key and hence the key management is simple. In public key cryptography we maintain two keys one is public key which is known to everybody and that can be used for encryption purpose and there is another key called private key which is kept secret key and that is used for decryption purpose only.
Nath et al [1,2,3,4,5,6] developed different symmetric key cryptosystem. The advantage of symmetric key method is that key management is

very simple. Recently Nath et al developed cryptography method called Modern Encryption Standard version-I and Modern Encryption Standard version-II. and Modern Encryption Standard version-III. The present version is called Modern Encryption Standard version-IV. Here the authors have used three independent cryptography methods namely columnar transposition method, bit-wise generalized vernam cipher method and bit wise XOR method. In the present version the different method can be used for large size of file encryption. The output shows that the encryption is very strong as the encrypted text is totally different if there is only one character different in two patterns. The standard encryption algorithm like RSA or DES if we apply on a pattern where all characters are same then after encryption the encrypted text will also show same repeated pattern. But the present method applied on repeated pattern but the output contains totally different pattern. In the present work the authors are proposing a symmetric key cryptography where firstly the keygen() function is called to generate the encryption number and the randomization number. Then the bit level columnar transposition encryption is applied block wise. After that bit level generalized vernam cipher method with feedback is applied. Hare the encryption key used is taken from the randomized two dimensional array. Finally the bit wise xor encryption is applied. This system of encryption can suitable to encrypt different financial secret data, corporate data, password, defense network etc.

2. Algorithm

Bitwise Columner Transposition Encryption Function Algorithm : col_trans_enc(file f1,file f2)

This function takes two files f1 and f2 as argument

step 1 : Open f1 in read mode.
step 2 : Open a file f2 in write mode.
step 3 : set i=0 and n=0
step 4 : set j=0
step 5 : set mat[i][j]=n
step 6 : set n=n+1
step 7 : Set j=j + 1 and if j<16 go to step 5
step 8 : set i=i+1 and if i<16 go to step 4
step 9 : Call randomization()
step 10 : set i=0 and set j=0
step 11 : Go to the start of f1
step 12 : read next 256 character from f1 and assign number of available character to n and assign the read characters to a[256]
step 13 : if n<=0 go to step 56
step 14 : set l=0
step 15 : if l>=n go to step 22
step 16 : Call function char_to_bit(a[l], binary) //this assign each bit of ch to elements of array binary[8]
step 17 : Set k=0
step 18 : if k>=8 go to step 21
step 19 : Set table[l][k]=binary[k] + 48
step 20 : Set k=k + 1 and go to step 18
step 21 : Set l=l+1 and go to step 15
step 22 : assign all element of filearr[8] to 0
step 23 : set i1=0 and j1=0
step 24 : set k=0
step 25 : if k>=8 go to step 44
step 26 : Set temp=mat[i][j]%8 + 1 // a%b returns the remainder after dividing a by b
step 27 : set i=i+1
step 28 : If i=16 set i=0 and j=j+1
step 29 : If j=16 set j=0.
step 30 : set m=0
step 31 : if m>=k go to step 34
step 32 : if temp=filearr[m] go to step 34
step 33 : increase m by 1 and go to step 31
step 34 : if m<k go to step 26
step 35 : set filearr[k]=temp
step 36 : Set k=k + 1
step 37 : set l=0
step 38 : if l>=n go to step 43
step 39 : set table1[i1][j1]=table[l][temp-1]
step 40 : set j1=j1+1
step 41 : if j1=8 set j1=0 and i1=i1+1
step 42 : set l=l+1 and go to step 38
step 43 : Go to step 25
step 44 : set l=0
step 45 : if l>=n go t step 53
step 46 : set add=0
step 47 : set k=0
step 48 : if k>=8 go to step 51
step 49 : set add=add+ (table1[l][k]-48)*power(7-j)
step 50 : set k=k+1 and go to step 48
step 51 : Write add to file f2
step 52 : set l=l+1 and go to step 45
step 53 : if n<256 go to step 56
step 54 : read next 256 character from f1 and assign number of available character

to n and assign the read characters to a[256]

step 55 : Go to step 13
step 56 : Close all the files.
step 57 : stop

bitwise vernam encryption with feedback:vernambitenc(file input,file output)

step 1 : set k=0
step 2 : set i=0
step 3 : if i>=16 go to step 10
step 4 : set j=0
step 5 : if j>=16 go to step 9
step 6 : set mat[i][j]=k
step 7 : increase k by 1
step 8 : increase j by 1 and go to step 5
step 9 : set i=i+1 and go to step 3
step 10 : call randomization()
step 11 : set i=j=0
step 12 : set cr1=0
step 13 : open input file as fpn
step 14 : read next character from fpn and assign to ch
step 15 : if eof is found go to step 36
step 16 : call char_to_bit(ch,bitpattern[8])
step 17 : call char_to_bit(mat[i][j],key_bit[8])
step 18 : set i=i+1
step 19 : if i=16 set i=0 and set j=j+1
step 20 : if j=16 set j=0
step 21 : set cr=(bitpattern[0]+key_bit[0]+cr1)%2
step 22 : set cb[0]=cr1=cr
step 23 : set k=1
step 24 : if k>=8 go to step 29
step 25 : set cr=(bitpattern[k]+key_bit[k]+cr1)%2
step 26 : set cb[k]=cr
step 27 : set cr1=cr
step 28 : set k=k+1 and go to step 24
step 29 : set add=0
step 30 : set k=0
step 31 : if k>=8 go to step 34
step 32 : set add=add+cb[k]*power(7-k)
step 33 : increase k by 1 and go to step 31
step 34 : write add to file f1
step 35 : go to step 14
step 36 : set k=0
step 37 : set i=0
step 38 : if i>=16 go to step 45
step 39 : set j=0
step 40 : if j>=16 go to step 44
step 41 : set mat[i][j]=k
step 42 : increase k by 1
step 43 : increase j by 1 and go to step 40
step 44 : set i=i+1 and go to step 38
step 45 : call randomization()
step 46 : set i=j=0
step 47 : set cr1=0
step 48 : read next character from f1 and assign to ch
step 49 : if eof is found go to step 70
step 50 : call char_to_bit(ch,bitpattern[8])
step 51 : call char_to_bit(mat[i][j],key_bit[8])
step 52 : set i=i+1
step 53 : if i=16 set i=0 and set j=j+1
step 54 : if j=16 set j=0
step 55 : set cr=(bitpattern[0]+key_bit[0]+cr1)%2
step 56 : set cb[0]=cr1=cr
step 57 : set k=1
step 58 : if k>=8 go to step 63
step 59 : set cr=(bitpattern[k]+key_bit[k]+cr1)%2
step 60 : set cb[k]=cr
step 61 : set cr1=cr
step 62 : set k=k+1 and go to step 58
step 63 : set add=0
step 64 : set k=0
step 65 : if k>=8 go to step 68
step 66 : set add=add+cb[k]*power(7-k)
step 67 : increase k by 1 and go to step 65
step 68 : write add to file f2
step 69 : go to step 48
step 70 : set k=0
step 71 : set i=0
step 72 : if i>=16 go to step 79
step 73 : set j=0
step 74 : if j>=16 go to step 78
step 75 : set mat[i][j]=k
step 76 : increase k by 1
step 77 : increase j by 1 and go to step 5
step 78 : set i=i+1 and go to step 72
step 79 : call randomization()
step 80 : set i=j=0
step 81 : set cr1=0
step 82 : read next character from f2 and assign to ch
step 83 : if eof is found go to step 104
step 84 : call char_to_bit(ch,bitpattern[8])
step 85 : call char_to_bit(mat[i][j],key_bit[8])
step 86 : set i=i+1
step 87 : if i=16 set i=0 and set j=j+1
step 88 : if j=16 set j=0
step 89 : set cr=(bitpattern[0]+key_bit[0]+cr1)%2

```
step 90  : set cb[0]=cr1=cr
step 91  : set k=1
step 92  : if k>=8 go to step 97
step 93  : set
           cr=(bitpattern[k]+key_bit[k]+cr1)%2
step 94  : set cb[k]=cr
step 95  : set cr1=cr
step 96  : set k=k+1 and go to step 92
step 97  : set add=0
step 98  : set k=0
step 99  : if k>=8 go to step 102
step 100 : set add=add+cb[k]*power(7-k)
step 101 : increase k by 1 and go to step 99
step 102 : write add to file f3
step 103 : go to step 82
step 104 : call file_rev(f3,f4)
step 105 : set k=0
step 106 : set i=0
step 107 : if i>=16 go to step 114
step 108 : set j=0
step 109 : if j>=16 go to step 113
step 110 : set mat[i][j]=k
step 111 : increase k by 1
step 112 : increase j by 1 and go to step 109
step 113 : set i=i+1 and go to step 107
step 114 : call randomization()
step 115 : set i=j=0
step 116 : set cr1=0
step 117 : read next character from f4 and
           assign to ch
step 118 : if eof is found go to step 139
step 119 : call char_to_bit(ch,bitpattern[8])
step 120 : call char_to_bit(mat[i][j],key_bit[8])
step 121 : set i=i+1
step 122 : if i=16 set i=0 and set j=j+1
step 123 : if j=16 set j=0
step 124 : set
           cr=(bitpattern[0]+key_bit[0]+cr1)%2
step 125 : set cb[0]=cr1=cr
step 126 : set k=1
step 127 : if k>=8 go to step 132
step 128 : set
           cr=(bitpattern[k]+key_bit[k]+cr1)%2
step 129 : set cb[k]=cr
step 130 : set cr1=cr
step 131 : set k=k+1 and go to step 127
step 132 : set add=0
step 133 : set k=0
step 134 : if k>=8 go to step 137
step 135 : set add=add+cb[k]*power(7-k)
step 136 : increase k by 1 and go to step 134
step 137 : write add to file output file
step 138 : go to step 117
step 139 : close all files
step 140 : stop
```

<u>Bit Wise XOR Encryption Function: bitxorenc(File F1,File F2)</u>

This function takes two files f1 and f2 as argument

```
step 1  : Open the file f1 in read mode
step 2  : Open the file f2 in write mode
step 3  : set l=size of file f1
step 4  : set n1=l/32
step 5  : set n1=l%32 //  a%b returns the
          remainder after dividing a by b
step 6  : Go to the start of file f1
step 7  : set i=0  and n=0
step 8  : set j=0
step 9  : set mat[i][j]=n
step 10 : set n=n+1
step 11 : Set j=j + 1 and if j<16 go to step 9
step 12 : set i=i+1 and if i<16 go to step 8
step 13 : set i=1
step 14 : if i>secure then go to step 17
step 15 : call randomization()
step 16 : set i=i+1 and  go to step 14
step 17 : set i=1
step 18 : if i>n1 then go to step 23
step 19 : Read next 32 character from file f1
          and assign it to array data1[32]
step 20 : Call bit_stream(data1[32])
step 21 : Call encrypt_bit()
step 22 : set i=i+1 go to step 18
step 23 : if n2=0 go to step 30
step 24 : set i=0
step 25 : if i>=n2 go to step 30
step 26 : Read next character fron file f1 and
          assign to data2[i] of array data2[32]
step 27 : set
          data2[i]=rshift_residual(data2[i],5)
step 28 : write data2[i] to file f2
step 29 : set i=i+1  go to step 25
step 30 : close all files
```

<u>Bitwise Columnar transposition decryption function algorithm:col_trans_dec(file f1,file f2)</u>

This algorithm is the reverse process of col_trans_enc algorithm

bitwise vernam decryption with feedback:vernambitdec(file input,file output)

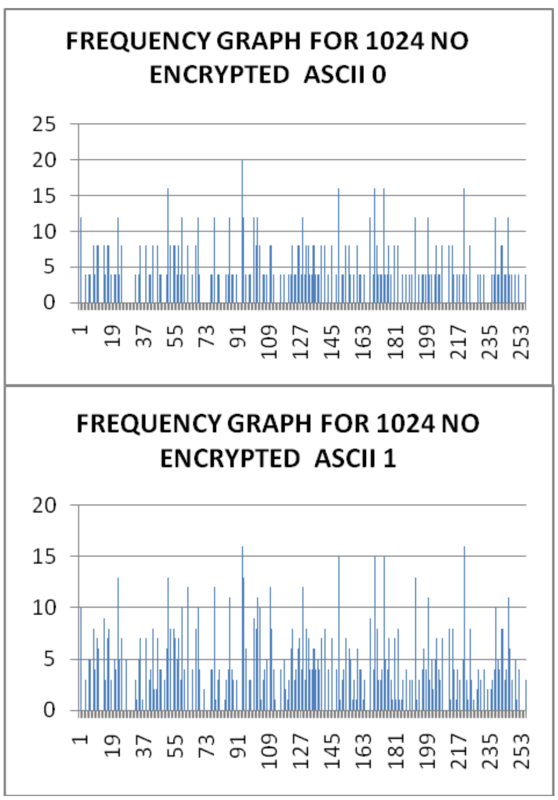
This algorithm is the reverse of vernambitenc algorithm.

<u>Bitwise Xor Decryption Function: Bitxordec(File F1,File F2)</u>

This algorithm is the reverse of bitxorenc algorithm.

3. Results and Discussion

The Modern Encryption Standard version IV (MES-IV) is applied on different type of text. This is applied on various repeated patterns such as 1024 numbers of ASCII 0 to ASCII 16. The corresponding frequency graphs are shown below. The graph shows the diverse nature of encrypted text even the input text is of repeated same character.



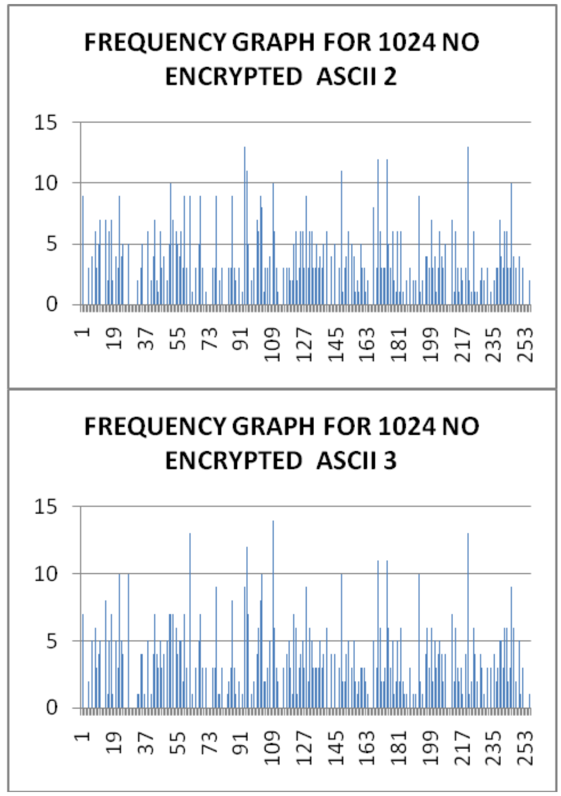


Fig-1: Frequency Graph of ASCII code '0','1','2','3'

This encryption method applied to different type of text/patterns and shown below are the pairs of such different type of patterns and the corresponding cipher text.

Table-1: Some known plain texts vs. encrypted text

Sl. No.	<u>ORIGINAL TEXT</u>	<u>ENCRYPTED TEXT</u>
1	AAAAAAAAAAAAAA AAAAAAAAAAAAAA AAAAAAAAAAAAAA AAAAAAAAAAAAAA AAAAAAAA (64-As)	莊갔血 ⅹ쌀ㄒ晞龍黯晰浤 됨嗜.;稠⌧□酘에脆黝 稲 펄♀ど竟頞懇眅
2	BAAAAAAAAAAAAA AAAAAAAAAAAAAA AAAAAAAAAAAAAA AAAAAAAAAAAAAA AAAAAAAAA (B + 64 As)	ÇŸiKw à-T?ô~ hed]»k Ôï ¿ žƒö r í©VôYÈÿ6®Ô°¦ÖïM£ÛuÎâ á?AÔì J£?u ðà™/

3	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAB (64 As + B)	Ç›-›o mTŸð□èAd]«b ÔïD¯ ¶ ô r i©VôYÈÿ6®Ô°¦ÖïM£ÛuÎâ á□AÔì J£□u ðà™/
4	00000000	J i!´ý
5	11111111	“ J i!´ý
6	01010101	– J i!´ý
7	1111111100000000	[0A¢RZ£ WÜ¢R ‡p
8	HE IS GOOD	¥ÒWÎê°Œ£5w
9	abcabcabcabc	„ □Dâ¢xº 5
10	HE IS GOON	¥Ðuìê°Œ£5w
11	CE IS GOON	´Àq¨®°Œ£5w

In the table below the plain text and the corresponding encrypted text are shown. The text of Sl. No. 12 & 13 are exactly same except the fourth character. However the encrypted texts are quite different under the same encryption key. The Sl. No. 14 & 15 shows the same thing.

	ORIGINAL TEXT	ENCRYPTED TEXT
12	The root of all modern programming languages is ALGOL,introduced in the early 1960s.ALGOL was the first computer language to use a block structure. Although it never became popular in USA, it was widely used in Europe. ALGOL gave the concept of structured programming to the computer science community.	Ç8 iØãÕ□y=g’ - Ç|e=ƒ@Eãl!2BÑWÈ¢Ö§□¼è ¨ i> Q›³ùÚ Çcß Û□'VNTÁ6Cçæ» Á [Â.RŽ3Ï´¬Œ'µþ ðfK- ,pE¦EŽbÒŸ‘ž□GÉ¥Fã s7‘aN€@ ìÉq_kú□ º□ . H~Ð4 ¦□áç¼»¶˜!”Ws®ðXäÅž Ú¿;Vˆœ□ÇY8ñ³t6%sÎ³□ ‰èF¤4cz ²e]mfE…ó êƒ‡µÙµ ‘»:: Ñaæøs›Í÷å□å¤Ó¿ÄŠ×\{Òð 1oWƒy□- #□(ŠÉÙ·Ü(©×«Èšm>s ¬RU![ƒpR õcfiÑDŠ«ZKE@Äà}z U>Ú «b □½ÈÀ€!Çìù4,Æêð%»¸a”[e D9Ü
13	The boot of all modern programming languages is ALGOL,introduced in the early 1960s.ALGOL was the first computer language to use a block structure. Although it never became popular in USA, it was widely used in Europe. ALGOL gave the concept of structured programming to the computer science community.	iæ(›kÄ ÔÁy<æÓ ø□ý%å□AÕ£ ¡7BÅwR,tÇ®¬~ ´ïh~ P òË□MF# Ú g&ÎQÁ"c}Æ rà ÍÀ2¾□sÏµ-Í à¸‰°³¼I,¼05&@Žvò ±<ífÙ3Dÿèrw‘`ÏÁrVªH1‡iû SÊ . häð–{‡□wå W·Ø!•Ö2œ¥ e…F Û/{& ™□Óy¢Ñ 5åÌ¯a ÉèG%uQ/V3%…ogÕÅƒ›ïƒ“ •C•¡Ñ›:* Áaæøs›Í÷õ‰¡%“wÆ‹G Rô !OM£y□- #□(ŠÉÙ·Ü(©×«Èšm>s ¬RU![ƒpR õcfiÑDŠ«ZKE@Äà}z U>Ú «b □½ÈÀ€!Çìù4,Æêð%»¸a”[e D9Ü
14	Modern batch processing is generally tied into a timesharing system. In this system the programme and the data are typed into the computer via a timesharing terminal or a personal computer.	öÖO d 9šªA y} rŽ□!x c¢ =$«6Y0‹Z+ ß=@Å«Â7 ½□H–gú Zå±—£¡ 'J)X uHgÂ a¨¡ Ob¯KÕ>ÍÏ <_æƒø¬ÌZ»ú!9¨æÈpH Ea ìé½ãîs' Ïœ^úò é Ê <ùðbz–,F ,ôð□Ò ƒ ! øåòM§Æ”Ñ¬çh ÿ-{î ùk □Å·0ƒÑ DÃ•©~ó€˜3» I3ìÜZÖœ□äUK%º
15	Moderp patch processing is generally tied into a timesharing system. In this system the programme and the data are typed into the computer via a timesharing terminal or a personal computer.	’c□›ÃkÇûëK- ú01'@8á£™|2ÌH´ §®□ul€{þ- ÖC÷.6_¿~É×U¯Hû¥j ¬1ÈSÚŽHöDöGac@¸7 SŽ® Õ?LŽ>i gÃ ®ÍÊûŠ¡<¨òèêh£ @ øé©Kîs' Kßˆ {²Úë ZTLyõbn¶¶f¬,ôð□Ò ƒ ! øåòM§Æ”Ñ¬çh ÿ-{î ùk □Å·0ƒÑ DÃ•©~ó€˜3» I3ìÜZÖœ□äUK%º

4. **Conclusion**

The results above show how strong the encryption method is. Three different methods of encryption techniques have been applied. The feedback mechanism gives to the strength of the encryption. The random matrix is generated on the input key. It is almost impossible to break the encryption without knowing the status of the key matrix which is randomized. The encrypted patterns shown above shows how different are the cipher text even when applied on similar text or the same text with subtle difference under same key. This shows the strength of the encryption. This method is applicable to different type of fields such as banking sectors, defence, short message encryption as well as large text encryption.

Acknowledgement

We are very much greateful to Indira Gandhi National Open University(IGNOU) and Department of Computer Science, St Xaviers College to give us such opportunity to work on symmetric key 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] 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).

[7] 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).

[8] 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).

[9] 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).

[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. No.20, Page- 46-53,May, 2012.

[12] Advanced Steganography Algorithm Using Randomized Intermediate QR Host Embedded with Any Encrypted Secret Message : ASA_QR Algorithm, Somdip Dey, Kalyan Mondal, Joyshree Nath, Asoke Nath, International Journal of Modern Education and Computer Science, No.6, Page 59-67, 2012.

[13] Ultra Encryption Standard Modified(UES) Version-I: Symmetric Key Cryptosystem with Multiple Encryption and Randomized Vernam Key using generalized Modified Vernam Cipher Method, Permutation method and Columnar Transposition Method, Satyaki Roy, Navajit Maitra, Shalabh Agarwal, Joyshree Nath, Asoke Nath, International Journal of Modern Education and Computer Science, No.7, Page 31-41(2012).

[14] Ultra Encryption Standard(UES) Version-III: Advanced Sysmmetric Key Cryptosystem with Bit-level Encryption Algorithm, Satyaki Roy, Navajitb Maitra, Shalabh Agarwal, Joyshree Nath, Asoke Nath, International Journal of Modern Education and Computer Science, No.7,Page 50-56(2012).

[15] Modified Caeser Cipher method applied on Generalized Modified Vernam Cipher method with feedback, MSA method and NJJSAA method: STJA Algorithm, Somdip Dey, Joyshree Nath and Asoke Nath, Proceedings of International Conference Worldcomp 2012 held at Las Vegas, USA, FCS-12, Page-112 – 119(2012).

[16] 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, Navajit Moitra, Joyshree Nath, Shalabh Agarwal and Asoke Nath, Proceedings of International Conference Worldcomp 2012 held at Las Vegas, USA, FCS-12, Page-97 – 104(2012).

[17] Ultra Encryption Algorithm(UEA): Bit level Symmetric key Cryptosystem with Randomized Bits and Feedback Mechanism, International Journal of Computer Applications(IJCA) USA, Vol-49, No.5, Page-34-40(2012).

[18] 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).

[19] 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).

[20] Bit Level Encryption Standard (BLES) : Version-II , Gaurav Bhadra, Tanya Bala, Samik Banik, Joyshree Nath and Asoke Nath , IEEE World Congress WICT_2012 to be held at Trivandrum, 30/10/2012-02/11/2012.

[21] Modern Encryption Standard(MES) version-I:An Advanced Cryptographic Method,Somdip Dey and Asoke Nath, IEEE World Congress WICT_2012 to be held at Trivandrum, 30/10/2012-02/11/2012.

[22] Bit Level Generalized Modified Vernam Cipher Method with Feedback, Prabal Banerjee and Asoke Nath, International Conference to be held at at Indore , Dec 15-16,2012.

Dr. Asoke Nath is Associate Professor in Department of Computer Science, St. Xavier's College (Autonomous), Kolkata-16. His major research areas are Cryptography and network security, Data hiding, Image processing, Green computing, E-learning.

Payel Pal, B.Sc (Hons). in Computer Science From University of Calcutta. Pesently pursuing MCA under IGNOU.