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 V: (MES-V)

Asoke Nath[1], Bidhusundar Samanta[2]

Abstract

In this paper the authors have introduced a new symmetric encryption method named as Modern Encryption System Standard Version V. The system is basically an extension of MES I,II,III & IV and Bit level Encryption Standard(BLES)-II & III. MES-I,II,III mostly based on byte level encryption method. BLES-I,II,III are based on mostly bit level encryption methods. Here mainly three different module of encryption have used.Those methods are Modified Generalized Vernam Cipher Method with feedback, Bit level Generalized Modified vernam cipher method with feedback, and Bit wise XOR operation. The Modified Generalized Vernam Cipher Method is the Byte level method and this is a block cipher method. Here 'Feedback' of each character is used for the encryption of the next character. In the Bit level Genaralized Modified Vernam Cipher Method with Feedback key used is the same length as the input file. The key is essentially a stream of bits. This method is used multiple times in both ways from left to right and then from right to left. In the Bit wise XOR operation ,bit wise XOR operation performed with bit-1 with bit-n(last bit) and substituted in the position n and bit-2 with bit-n-2 and substituted in position n-2. The present method applies multiple encryption and multiple decryption. From the entered key string the randomization number and encryption number are calculated using a method proposed by Nath et al. This present method will be used for encrypting short message, password,bank data, and other confidential data. This method is free from brute force attack,plain text attack or differential attack.

Keywords

Plain text, Cipher text, Randomization, Bit level encryption, Feedback

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

1. Introduction

In this age of universal electronic connectivity, of viruses and hackers, of electronic eavesdropping and electronic fraud it is a big challenge for a sender to send confidential data from one place to another through network. The confidential data cannot be sent from one computer to another computer as the intruder and hacker can intercept the data. The Hackers have created various crack software. Using that software anyone can break any password and can log into any confidential site. All these are happening because of free network access. Network access is now free to anyone. So when a user is working in a network environment then the user must be very careful about his/her confidential data. . Any kind of private data should not be sent in raw form from one computer to another. The private/confidential data must be encrypted first and then it should be sent over the internet. Otherwise anytime the disaster may come. To overcome this problem one has to send the encrypted text or cipher text form client to server or to another client instead of sending in unencrypted form.

Cryptography and cryptanalysis is now a very important research area in modern digital communication network. Nowadays network security and cryptography is an emerging research area where the programmers are constantly trying to develop some strong encryption algorithm so that the confidential data when encrypted remain secret from the attacks of hackers and intruders.

The cryptography methods can be divided into two categories: (i) symmetric key cryptography where one key is used for both encryptions an 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. The main advantage of symmetric key cryptography is that the key management is very simple as one key is used for both encryption as well as for decryption purpose. In this method the key is called secret key and it should be known to sender and receiver both.

The present method is a symmetric key cryptographic method which is introduced as The Modern Encryption Standard Version V. This is an upgraded version of earlier version developed by Nath et al. 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 The Modern Encryption Standard Version V uses three different encryption method such as modified generalized byte level Vernam cipher method with feedback, bit level generalized modified vernam cipher method with feedback, and bitwise xor encryption method. In this version both bit level and byte level encryption method are applied to develop more secure encryption. In both byte and bit level of vernam cipher method feedback from previous encryption is used for next encryption which results in more potent encryption and in both cases the key used is taken from the randomized array. The keygen() function is called at the start of the encryption which generate the encryption number and the randomization number. The output shows that the encryption is very strong as the encrypted text is totally different .The present method applied on repeated pattern but the output contains totally different pattern. This method is useful for encryption of different text, password, defense data, bank data etc.

2. Algorithm bytewise vernam cipher with feedback encryption function:vernamenc(file f1 file f2)

step 1 : set ch1=0
step 2 : set n2=0
step 3 : set i=0
step 4 : if i>=16 go to step 11
step 5 : set j=0
step 6 : if j>=16 go to step 10
step 7 : set mat[i][j]=n2
step 8 : set n2=n2+1
step 9 : set j=j+1 and go to step 6
step 10 : set i=i+1 and go to step 4
step 11 : call randomization()
step 12 : n2=0
step 13 : set i=0
step 14 : if i>=16 go to step 21
step 15 : set j=0
step 16 : if j>=16 go to step 20
step 17 : set key[n2]=mat[i][j]
step 18 : set n2=n2+1
step 19 : set j=j+1 and go to step 16
step 20 : set i=i+1 and go to step 14
step 21 : open file f1 in read mode
step 22 : open file f2 in write mode
step 23 : set times3=1
step 24 : set pass=1
step 25 : read first 256 character from file f1 and assign it to array a[256] and assign the number of character read to n
step 26 : if n!=256 go to step 51
step 27 : set i=0
step 28 : if i>=n go to step 31
step 29 : set str[i]=a[i] // str[256] is an array
step 30 : set i=i+1 and goto step 28
step 31 : call encryption(str,n)
step 32 : read first 256 character from file f1 and assign it to array a[256] and assign the number of character read to n
step 33 : if pass=1 set times=(times + times3*11)%64 and increase pass by 1
step 34 : if pass=2 set times=(times + times3*3)%64 and increase pass by 1
step 35 : if pass=3 set times=(times + times3*7)%64 and increase pass by 1
step 36 : if pass=4 set times=(times + times3*13)%64 and increase pass by 1
step 37 : if pass=5 set times=(times + times3*times3)%64 and increase pass by 1
step 38 : if pass=6 set times=(times + times3*times3*times3)%64 and set pass=1
step 39 : increase times3 by 1
step 40 : call randomization()
step 41 : set n2=0
step 42 : set i=0
step 43 : if i>=16 go to step 50
step 44 : set j=0
step 45 : if j>=16 go to step 49
step 46 : set key[n2]=mat[i][j]
step 47 : increase n2 by 1
step 48 : set j=j+1 and go to step 45
step 49 : set i=i+1 and go to step 43
step 50 : go to step 26

step 51 : set i=0
step 52 : if i>=n go to step 55
step 53 : str[i]=a[i]
step 54 : set i=i+1 and go to step 52
step 55 : call encryption(str,n)
step 56 : close all files

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

Bit wise XOR encryption function:bitxorenc(file f1,file f2)

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

bytewise vernam cipher with feedback decryption function:vernamdec(file f1 file f2)

This algorithm is reverse of vernamenc algorithm

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

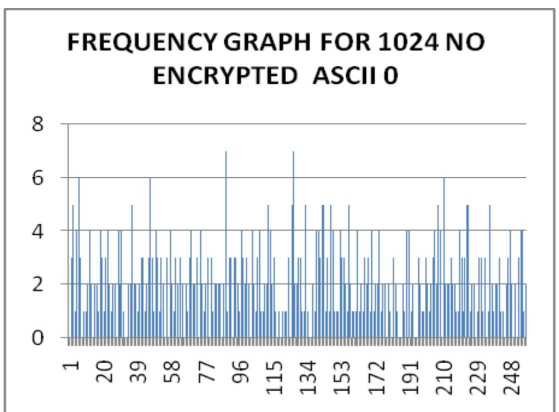
This algorithm is reverse of vernambitenc algorithm

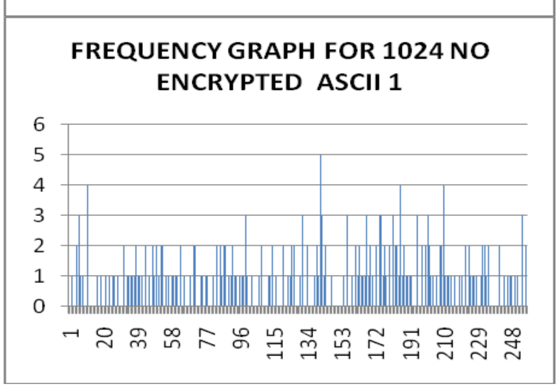
bitwise xor decryption function:bitxordec(file f1,file f2)

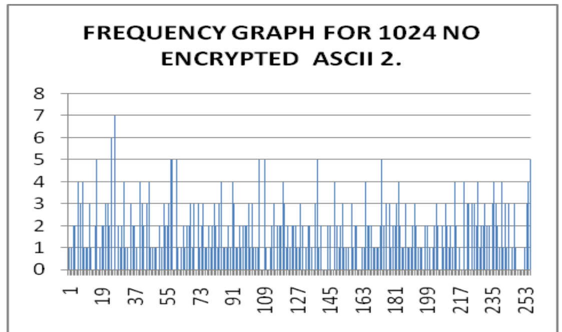
This algorithm is reverse of bitxorenc algorithm

3. Results and Discussion

The MES-V(Modern Encryption Standard version V) applied on different type of text files. For example this method when applied to a text containing 1024 numbers of ASCII 0 gives encrypted characters of different types which is shown in the graph below. Also shown the graphs of other encrypted ASCII characters below.







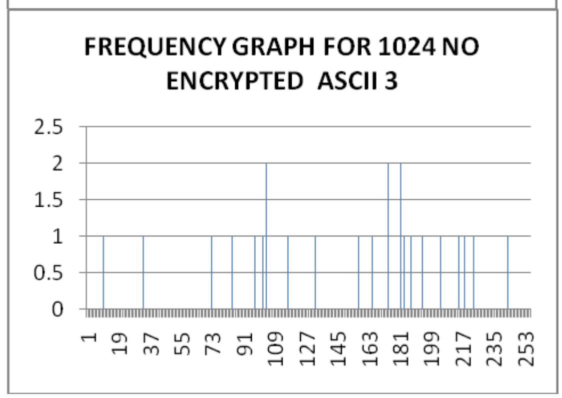


Fig 1: Frequency Graph of Different ASCII codes

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 original text and Encrypted Text

Sl. No.	Original Text	Encrypted Text
1	AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA AAAA (64-As)	岸□뛠䝃□≮셏ᘐ뱍 ᅩ문سر譙 I粼ㅡ□닺뵐ⵜ 뷻□劧顫شمم Ė晷笐킚ᵹ嚽
2	BAAAAAAAAAAAAAA AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA AAAAA (B + 64 As)	[åÑ³ • ½ é¨`ViÐºq>$\{4÷[-î¹ üÐ:Ù□ sÇ □ F□ ”«z'¦ÒI•Þp üÅ ‰‘Pœ, × • kÿ¨"æ
3	AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA AAAAB (64 As + B)	Q‹ÙÆD1ï+÷¥,oÂYy€ ¾!O¥Á) æß ¦¯ÛÐgéwƒx™ÄTâ˜·1ÂcëLäx lÈ.üB¨7Í³ÒÔíœƒ 8GÁ
4	00000000	óB2ýÿ—. •

5	11111111	½‡` I•
6	01010101	}·iÉþùÉ •
7	1111111100000000	< _ëWŒÀðG□ Š- F æ
8	HE IS GOOD	믈茚惫Œ鲞
9	abcabcabcabc	öÄtbÃ ËœY Žˆ
10	HE IS GOON	D]»ÖÆs ÿ\
11	CE IS GOON	òÂV ˆ%NØƒ›

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

	Original Text	Encrypted Text
12	Information security has become a very critical aspect of modern computing systems. With the global acceptance of the Internet,virtually every computer in the world today is connected to every other. While this has created tremendous productivity and unprecedented opportunities in the world we live in, it has also created new risks for the users of these computers.	f¾ï à ç |3 (BG‹³m */R²ÚÓ€ŠØ_na□ ".œb nž: •)äÓJ õuÁÞéLintj¤Þ±üJf ÆŠ® :•±‡´ î.ÿz' · 0}'2·Ço- îZ3ì"ÿ½¢ :"j9„,ÜÎ ®zÀûn´y%eö $®Ð}{À› wSõÜâì.É.÷ ³*Ú}Dc'□ ÆI×ÛÍ»kÆº€ƒ ¼úp Û ž7'mºÌ{0%ß¤"L— eÎ².KGe® +□ □ 9†ä& çï□ ºªM□ <íDøDÆ¸ºX£Ïº 3ÜXî®A ªà L1‹fú^ L{ä„ûòQ š}Œif> □ £evW}1m‰²¶XU□ ï, /OSj • A „§ƒÏ–ÄPæí— Ñü¹ ÷õ'ZKÕÆ|ÊÈ*…çzÇ32 õ|Ý¤¿ ûM˜ bb-ë Z K'» ùð ^±"jý¸ÀPoú °<ÄÎx*ïGŠuxGÏD3,âDr
13	Inforpation security has become a very critical aspect of modern computing systems. With the global acceptance of the Internet,virtually every computer in the world today is connected to every other. While this has created tremendous productivity and unprecedented opportunities in the world we live in, it has also created new risks for the users of these computers.	‹òÎàºÎ olšf÷ ¦KÍ åš ¯òVî' ó= }€Ç:]Áä¨5> åúD úº#[½FÈúF™ÑO)äô øVÍƒ ² 7æj» ú= ôi=k (ê' ŸqÐ_T ŽÝ„€ˆT¤}ÀÐ8 Â G n;ÓFÕûó 5`òH"øc"!«Áx•ù"n IŸmøoŒÏŸ„ ¢ • ô S¨ äQ=zõAµ¼µ§vAp □¶dÂá4> »x2à»¬"]B#ñû! /ˆŒ*Þ^□V~o‡•ÂÇ .¼ö6· = ¶oÂ□ üÞ„À1KöUò˜ [˜ØÍøÕñŠùJ ûØô ˆÃ$RÙHœ‹;é µFoõ™~A ¢ù;ê‹TeÎV¥F6$ .-û i*ÝÔÎ FÄxES -Ç[íÞ"¡•8F— ê²„(•«`‡'ÉlÒUŠ"‡®ƒ9 zÏˆGÍ€e¨±-b Œ¼s†6è ÂúùhX„ ŸÁÁ£3,âDr
14	ISOC is a professional membership society with world wide organizational and individual membership. It provides leadership in addressing issues that confront the future of the internet and is the organization home for the groups responsible for internet infrastructure standards.	Wóu;)ÿ´mˆà" m5™êÄ"¨– ™□ Ö"7á¡ □ Ì+3‡Ç aL½ÀÄÿÝô]ŒcU¯ë¡ R!(U=™zE€zžÇ'@é • • — QòA| ÐØ Ñ M$á ƒŽ Ì÷Ñt" _h à9˜E7Ö$´ÃGsoŠK Ä Îõ§0Îx YÊOÁuJ¥>ëÎ·RñyFPÁßõ Ù^ö™¬ 2H |ž]ä:a`¹äðü`Ÿ¦@Ã•¨™ à— ‰ ‡GCLÂ˜´•HK ¨–p»(ªBôgK''á‹BoE›□ Ò‹€LÚb *òÃ { `Ê\Ç\'KI□ ?‡Õþ‡• àuí <¼àUEnñpv©cÁ¾"š/Â^ □ 5 Šq£»…Š¤°{ H¢¯F¾€:

15	ISOC in a professional membership society with world wide organizational and individual membership. It provides leadership in addressing issues that confront the future of the internet and is the organization home for the groups responsible for internet infrastructure standards.	ã "S iQ»(åªò ªî™•×Õ Ÿ×fŸUÜJEÞÅ¡½ ¯n+×'ÃL$Í»¡+º , é Œ ùëÝ¥Xg¤LHíaáP•UÄ»,Á ¡ñ×Ïm ånÄlÿ†µK!ê ŸaT^f• ;9K¥B*± \ô1¤€FÚŽ ∶·iž ˜9 #Ê,_□ ™Uy)¯Â\ŠÐg H¹ E!D8ú<_(×d‰ïh ÕGì'Lâ¥ = ãóf!ôPÉÙ{¨– d÷□ ž5p 6bö¼£ Äâˆ¾ jm±mp□ € Ã“ ¨-}P9ää]›•Ý"¤ÃØ¯ µ‘ƒ3 öô„QoT=nÑã §_Þñ¤É{kœš□ «ÃÑ¢¶°¸ñš A Ò`]P¹A□Å

4. Conclusion

The MES-V is built up on both bit level and byte level encryption method. The method is absolutely strong against any type of attack such as known plain text attack or differential attack or brute force attack. Though only the component encryption modules are applied here for once, multiple level of application will yield much more stronger technique and more potential against any type of cryptographic attack. The encrypted text cannot be decrypted without knowing the random matrix. The spectral analysis shows the diversity of encrypted characters even when the input plain text characters are of same type. The same text except a different character at any position gives quite different cipher text under the same encryption key. This method is applicable for encryption of short messages, secret data, financial data, defense data, as well as applicable for large text encryption also.

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] 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.

Bidhusundar Samanta BME from Jadavpur University. Presently pursuing MCA under IGNOU.