Modern Encryption Standard(MES) version-I: An Advanced Cryptographic Method

Somdip Dey
Department of Computer Science
St. Xavier's College [Autonomous]
Kolkata, India.
e-mail: somdipdey@ieee.org; somdipdey@acm.org

Asoke Nath
Department of Computer Science
St. Xavier's College [Autonomous]
Kolkata, India.
e-mail: asokejoy1@gmail.com

***Abstract*— In the present world we need a high security for transmitting any digital information from client to another client or one machine to another machine. In the present work the authors focus on how one can achieve high order data security while transmitting from one place to another place. The authors here propose a new encryption standard (algorithm), which is the amalgamation of two different encryption algorithms developed by Nath et al. namely TTJSA and DJSA in randomized fashion. The title of the proposed method is Modern Encryption Standard version-I (MES ver-I) and, the method is achieved by splitting the file, which is to be encrypted, and encrypting the split sections of the file in various ways using TTJSA and DJSA cipher methods. The method has been tested on different files and the results were very satisfactory. The primary idea behind the implementation of MES ver-I is to build a strong encryption method, which should be unbreakable by any kind of brute force method or differential attack. In the result section the authors have shown the spectral analysis which clearly shows that the present method is free from any kind of cryptography attack namely brute force method, known plain text attack or differential attack.**

Keywords-TTJSA; DJSA; randomization; bit manipulation; cryptography;

I. INTRODUCTION

Today, the digital world is growing rapidly and security of the data in this digital world is of highest priority. For an example, a person is communicating with his relatives over the internet and sending money-order to that person. Now, if there are no security measures in the digital world, then there is a possibility that a hacker may intercept the communicating data and revert back the money order procedure to credit the account of the hacker. This is really unacceptable and undesirable too. For this reason, cryptographers and scientist are trying hard every day to invent new cryptographic methods to secure the communication and information in the digital world.

Now, Cryptography can be basically segregated into two types:

(i) Symmetric Key Cryptography
(ii) Public Key Cryptography

Symmetric Key Cryptography is the type of cryptographic method, where only one key is used for encryption purpose and the same key is used for decryption also.

Whereas in Public Key Cryptography, one key is used for encryption technique and another key, which is basically called a 'Public Key', is used for the decryption method.

The cryptographic method proposed in this paper is a type of symmetric key cryptographic method, which is in fact the combination of two different symmetric key cryptographic methods, TTJSA [1] and DJSA [5], which were also developed by Nath et al.[1][2][3][4][5][8][9][14][15].

MES ver-I (Modern Encryption Standard version-I) is essentially the combination of existing encryption methods developed by Nath et al. In MES Version-I we will be considering two independent methods (i) DJSA [5] which is the extended version of MSA [2] and (ii) TTJSA [1] which is again a combination of 3 independent methods namely (1) NJJSAA [3] method, (2) MSA method and (3) generalized modified Vernam Cipher method [1]. The idea of Modern encryption standard is to make a symmetric key cryptography method, which should be unbreakable.

II. MES VERSION-I METHOD

A. *Algorithm of MES Version-I*

In this section we discuss the algorithm, which is followed by MES Version-I Cipher method. The Algorithm is as follows:

1) Encryption Algortihm

- Step-1: Split plain text file testdata.txt into 2 files t1.txt and t2.txt
- Step-2: Encrypt t1.txt using TTJSA method.

  [t1.txt -> Encrypt -> t11.txt]
- Step-3: Encrypt t2.txt using TTJSA method.

  [t2.txt -> Encrypt -> t21.txt]
- Step-4: Encrypt t21.txt using TTJSA method.

978-1-4673-4805-8/12/$31.00 ©2012 IEEE

[t21.txt -> Encrypt -> t2e.txt]

- Step-5: Combine t2e.txt and t11.txt.

  [t2e.txt + t1q1.txt -> Combine -> t12e.txt]

- Step-6: Encrypt t12e.txt using DJSA method.

  [t12e.txt ->Encrypt -> t12e1.txt]

- Step-7: Split t12e1.txt into 2 files.

  [t12e1.txt -> Split -> t12e11.txt + t12e12.txt]

- Step-8: Encrypt t12e11.txt using TTJSA.

  [t12e11.txt -> Encrypt -> t12n1.txt]

- Step-9: Encrypt t12e12.txt using TTJSA.

  [t12e12.txt -> Encrypt -> t12n2.txt]

- Step-10: Combine t12n2.txt, t12n1.txt files.

  [t12n2.txt + t12n1.txt -> Combine -> t12ne.txt]

- Step-11: Encrypt using DJSA method.

  [t12ne.txt -> Encrypt -> t12nee.txt]

- Step-12: Encrypt t12nee.txt using TTJSA.

  [t12nee.txt -> Encrypt -> t12nef.txt]

- So the final encrypted file will be t12nef.txt

2) Decryption Algorithm

- Step-1: Decrypt t12nef.txt using TTJSA.

  [T12nef.txt -> Decrypt -> t12.txt]

- Step-2: Decrypt using DJSA method.

  [t12.txt -> Decrypt -> t12d.txt]

- Step-3: Split t12d.txt into two files.

  [t12d.txt -> Split -> t122.txt + t121.txt]

- Step-4: Decrypt t121.txt using TTJSA.

  [t121.txt -> Decrypt -> t11d.txt]

- Step-5: Decrypt t122.txt using TTJSA.

  [t122.txt -> Decrypt -> t12d.txt]

- Step-6: Combine t11d.txt and t12d.txt.

  [t11d.txt + t12d.txt -> Combine -> t1_2.txt]

- Step-7: Decrypt t1_2.txt using DJSA method.

  [t1_2.txt -> Decrypt -> t1_2d.txt]

- Step-8: Split t1_2d.txt into 2 files.

  [t1_2d.txt -> Split -> t2d.txt + t1d.txt]

- Step-9: Decrypt t2d.txt using TTJSA method.

  [t2d.txt -> Decrypt -> t2d1.txt]

- Step-10: Decrypt t2d1.txt using TTJSA method.

  [t2d1.txt -> Decrypt -> t2df.txt]

- Step-11: Decrypt t1d.txt using TTJSA method.

  [t1d.txt -> Decrypt -> t1df.txt]

- Step-12: Combine t1df.txt and t2df.txt.

  [t1df.txt+t2df.txt -> Combine -> t12f.txt]

- t12f.txt is the final decrypted file. t12f.txt will be same as testdata.txt.

B. TTJSA Method

TTJSA [1] method is a combination of 3 distinct cryptographic methods, namely, (i) Generalized Modified Vernam Cipher Method [1], (ii) MSA [2] method and (iii) NJJSA [3] method. To begin the method a user has to enter a text-key, which may be at most 16 characters in length. From the text-key, the randomization number and the encryption number is calculated using a method proposed by Nath et al. A minor change in the text-key will change the randomization number and the encryption number quite a lot. The method have also been tested on various types of known text files and have been found that, even if there is repetition in the input file, the encrypted file contains no repetition of patterns.

1) Generalized Modified Vernam Cipher

This method is similar to the general Vernam cipher, where the character of the file is XOR-ed with the pad. But, the difference between the method used in TTJSA and the

normal one is that the generalized modified Vernam Cipher has feedback mechanism in it. For that reason, even if there is slight change in the original file, the content of the encrypted file will be totally different. This method is achieved, first by splitting the whole file into blocks (as in case of block cipher) containing 256 bytes or less than that. Each block is then XOR-ed with the pad of same size (<=256 bytes) and if there are any excess bits in the last character of that block after XOR-ing, then that is carried over to the first character of the next block. And this is known as the feedback mechanism.

2) NJJSA Method

The encryption number (=secure) and randomization number (=times) is calculated according to the method mentioned in MSA algorithm [2].

Step 1: Read 32 bytes at a time from the input file.

Step 2: Convert 32 bytes into 256 bits and store in some 1-dimensional array.

Step 3: Choose the first bit from the bit stream and also the corresponding number(n) from the key matrix. Interchange the 1st bit and the n-th bit of the bit stream.

Step 4: Repeat step-3 for 2nd bit, 3rd bit...256-th bit of the bit stream

Step 5: Perform right shift by one bit.

Step 6: Perform bit(1) XOR bit(2), bit(3) XOR bit(4),...,bit(255) XOR bit(256)

Step 7: Repeat Step 5 with 2 bit right, 3 bit right,...,n bit right shift followed by Step 6 after each completion of right bit shift.

3) MSA Method

Nath et al. [2] proposed a symmetric key method where they have used a random key generator for generating the initial key and that key is used for encrypting the given source file. MSA method is basically a substitution method where we take 2 characters from any input file and then search the corresponding characters from the random key matrix and store the encrypted data in another file. MSA method provides us multiple encryptions and multiple decryptions. The key matrix (16x16) is formed from all characters (ASCII code 0 to 255) in a random order.

The randomization of key matrix is done using the following function calls:

Step-1: call Function cycling()

Step-2: call Function upshift()

Step-3: call Function downshift()

Step-4: call Function leftshift()

Step-5: call Function rightshift()

How the above functions will work have been discussed in detail by Nath et al [2]. The idea of these functions is to make elements in a square matrix in a random order so that no one can predict what will be the nearest neighbor of a particular element in that matrix. This method is basically modified Playfair method. In Playfair method one can only encrypt Alphabets but in MSA one can encrypt any character whose ASCII code from 0-255 and one can apply multiple encryption here which is not possible in normal Playfair method.

C. DJSA Method

This method is also proposed by Nath et al [5]. In this method we first generate a randomized key of size 256X256X2. To create Random key of size(256x256x2) we have to chose any text-key which is a secret key. The size of text-key must be less than or equal to 16 characters long. These 16 characters can be any of the 256 characters(ASCII code 0 to 255). The relative position and the character itself is very important in this method to calculate the randomization number and the encryption number. Then two numbers are generated from the key itself and they are named "randomization number" and "encryption number".

Now, the randomization number is used to randomize the key and the encryption number is used to encrypt the file 'encryption number' of times. Then, the randomized key is used to encrypt the original file 'encryption number' of times like the usual Playfair Cipher method.

III. Block Diagram of The Proposed Method

In this section, we provide the block diagram of MES Version-I Cipher Method.

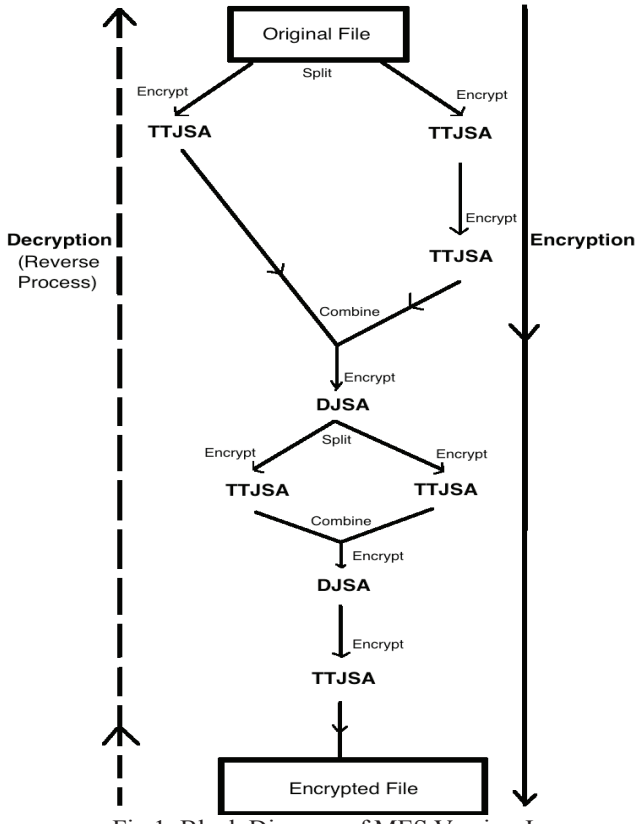

Fig 1: Block Diagram of MES Version-I

IV. Results and Discussions

The method discussed in this paper has been tested on various file types and the results were very satisfactory. In Table I we provide two such test cases.

Table I: Test Cases

Serial No.	Test Cases
1	File containing 4096 bytes of ASCII Value (0) i.e. NULL
2	File containing an article of size 3645 bytes

A. *Test Case 1*

Fig 2.1 shows the frequency analysis of the original file of test case 1, and Fig 2.2 shows the frequency analysis of the encrypted file of test case 1.

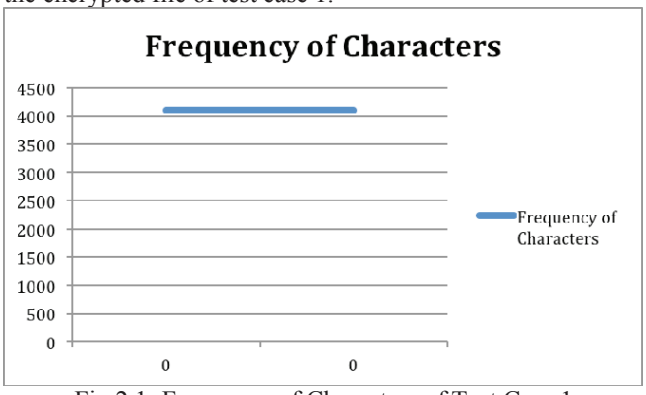

Fig 2.1: Frequency of Characters of Test Case 1

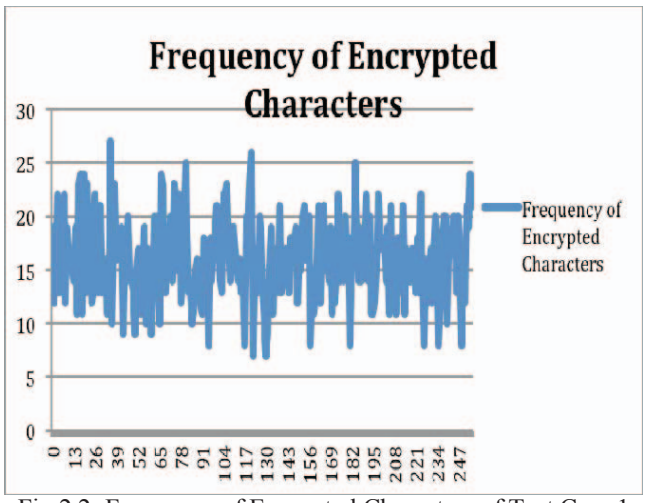

Fig 2.2: Frequency of Encrypted Characters of Test Case 1

B. *Test Case 2*

Fig 3.1 shows the frequency analysis of the original file of test case 2, and Fig 3.2 shows the frequency analysis of the encrypted file of test case 2.

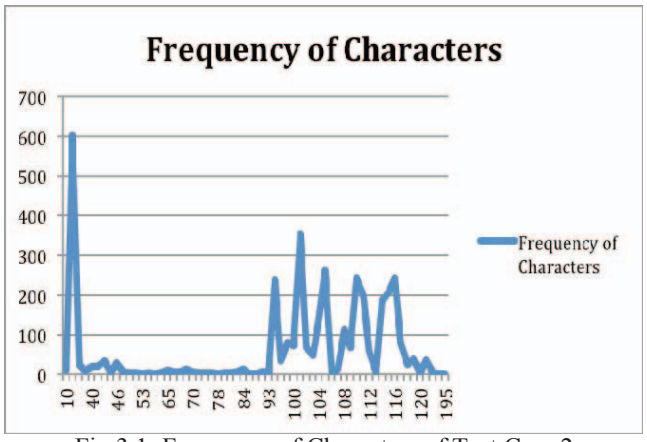

Fig 3.1: Frequency of Characters of Test Case 2

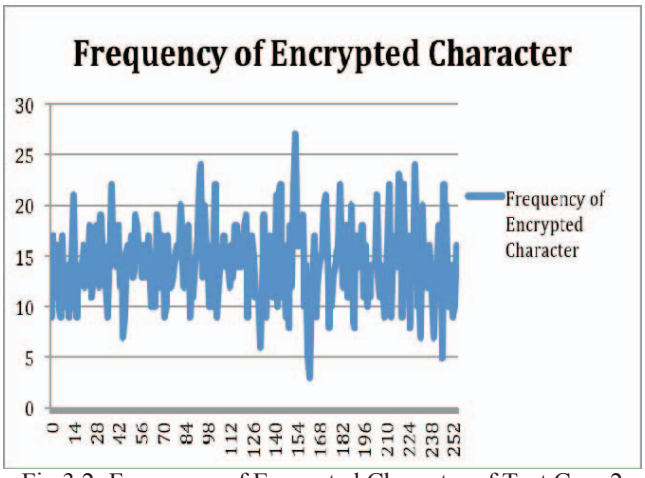

Fig 3.2: Frequency of Encrypted Character of Test Case 2

V. Cryptanalysis

One of the classical cryptanalysis method used is by detecting the frequency of characters in the encrypted text (message). So to test the effectiveness of MES Version-I method, spectral analysis of the frequency of characters are closely observed. Using this method, MES Version-I, we ran many analysis and tested different files as input and used various methods of cryptanalysis. To show the usefulness and integrity of this cryptographic module, we used spectral analysis of the frequency of characters.

As a test case, we have used a file containing 4096 bytes of ASCII Value 255. In Fig 4.1 we show the spectral analysis of the frequency of characters of the original file, and in Fig 4.2 we show the spectral analysis of the frequency of characters of the encrypted file.

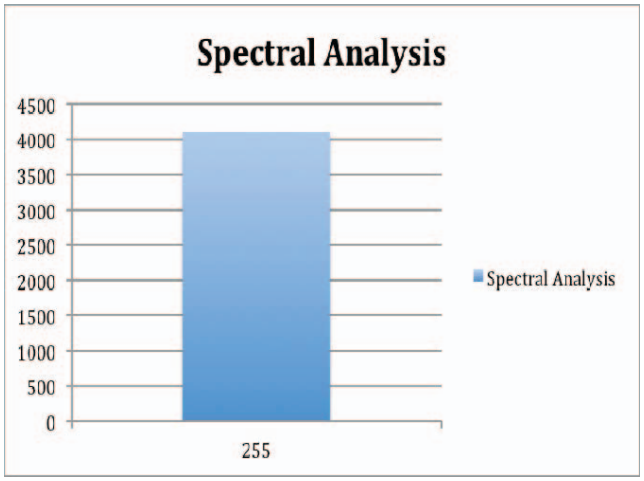

Fig 4.1: Spectral Analysis of Frequency of Characters of 4096 bytes of ASCII Value 255

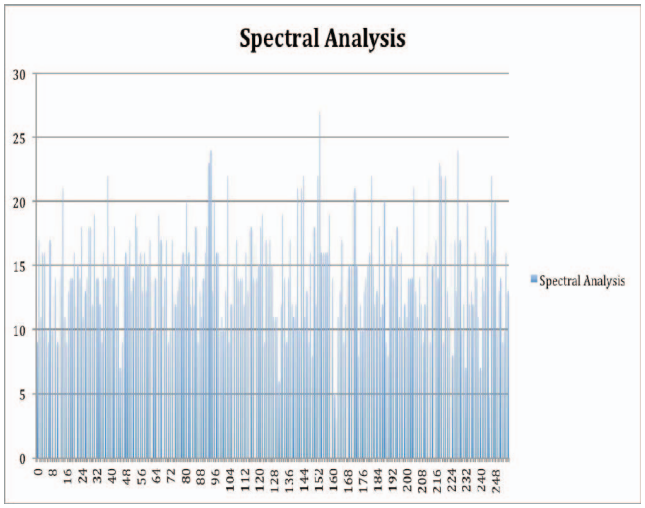

Fig 4.2: Spectral Analysis of Frequency of Encrypted Characters of 4096 bytes of ASCII Value 255

Thus, from the spectral analysis and also from the frequency analysis of the result section, it is evident that there is no pattern or repetition in the encrypted file. For this reason the method is very effective and strong.

VI. Comparisson With Other Methods

In this section, we compare the MES version-I method with normal TTJSA and DJSA cryptographic methods.

First, in Table II we compare the methods in case of execution time.

Table II: Comparison In Case of Execution Time (secs)

File Size	TTJSA	DJSA	MES Ver-I
1024 bytes	1	2	4
2048 bytes	2	2	6
4096 bytes	2	3	6

Although, the effectiveness of the MES version-I can not be judged by reviewing the comparison of execution time.

In MES version-I method we execute the TTJSA and DJSA methods several times by splitting the same file. While in TTJSA method or in DJSA method we execute the whole method only single time. Since, in MES method we are executing TTJSA and DJSA methods in so many different ways, that is why the total execution time is more than its predecessors.

VII. Conclusion and Future Scope

The present method is very much flexible in comparison to any standard methods. We can split the original file more than two also and we can apply TTJSA and DJSA in alternate ways and the whole process we can apply multiple times. The present method may be applied to encrypt any database or confidential data like bank data. This method may be further strengthened if we apply this in bit level. Split the plain text file into 2 files and apply bit level encryption separately in two different blocks. Then club the two encrypted files and apply the bit level encryption on the combined file. We are now working on this.

Acknowledgment

Somdip Dey (SD) expresses his gratitude to all his fellow students and faculty members of the Computer Science Department of St. Xavier's College [Autonomous], Kolkata, India, for their support and enthusiasm. Asoke Nath (AN) is grateful to Dr. Fr. Felix Raj, Principal St. Xavier's College, Kolkata for giving opportunity to work in the field of data hiding and retrieval.

References

[1] Symmetric key cryptosystem using combined cryptographic algorithms - Generalized modified Vernam Cipher method, MSA method and NJJSAA method: TTJSA algorithm " Proceedings of Information and Communication Technologies (WICT), 2011 " held at Mumbai, 11th – 14th Dec, 2011, Pages:1175-1180.

[2] 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), P-Vol-2, 239-244(2010).

[3] 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).

[4] 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, Vol3, issue-2, Page 66-71,Feb(2011)

[5] 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 CSNT-2011 held at SMVDU(Jammu) 3-5 June,2011, Page-89-94(2011).

[6] Somdip Dey, "SD-REE: A Cryptographic Method To Exclude Repetition From a Message", Proceedings of The International Conference on Informatics & Applications (ICIA 2012), Malaysia, p. 182 – 189.

[7] Somdip Dey, "SD-AREE: A New Modified Caesar Cipher Cryptographic Method Along with Bit-Manipulation to Exclude Repetition from a Message to be Encrypted", Journal: Computing Research Repository - CoRR, vol. abs/1205.4279, 2012.

[8] Somdip Dey, Joyshree Nath and Asoke Nath. Article: An Advanced Combined Symmetric Key Cryptographic Method using Bit Manipulation, Bit Reversal, Modified Caesar Cipher (SD-REE), DJSA method, TTJSA method: SJA-I Algorithm. *International Journal of Computer Applications* 46(20): 46-53, May 2012. Published by Foundation of Computer Science, New York, USA.

[9] Somdip Dey,Joyshree Nath,Asoke Nath,"An Integrated Symmetric Key Cryptographic Method – Amalgamation of TTJSA Algorithm, Advanced Caesar Cipher Algorithm, Bit Rotation and Reversal Method: SJA Algorithm", IJMECS, vol.4, no.5, pp.1-9, 2012.

[10] Somdip Dey, "SD-EI: A cryptographic technique to encrypt images", 2012 IEEE International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), pp. 28-32.

[11] Somdip Dey, "SD-AEI: An advanced encryption technique for images", 2012 IEEE Second International Conference on Digital Information Processing and Communications (ICDIPC), pp. 69-74.

[12] 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).

[13] 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). .

[14] Somdip Dey, "An Image Encryption Method: SD-Advanced Image Encryption Standard: SD-AIES", International Journal of Cyber-Security and Digital Forensics (IJCSDF) 1(2), pp. 82-88.

[15] Somdip Dey, "Amalgamation of Cyclic Bit Operation in SD-EI Image Encryption Method: An Advanced Version of SD-EI Method: SD-EI Ver-2", International Journal of Cyber-Security and Digital Forensics (IJCSDF) 1(3), pp. 238-242.

[16] Symmetric key Cryptography using two-way updated – Generalized Vernam Cipher method: TTSJA algorithm, Trisha Chatterjee, Tamodeep Das, Shayan dey, Joyshree Nath, Asoke Nath , International Journal of Computer Applications(IJCA, USA), Vol 42, No.1, March, Pg: 34 -39(2012).

[17] Bit Level Encryption Standard(BLES): Version-I, Neeraj Khanna, Dripto Chatterjee, Joyshree Nath, Asoke Nath, International Journal of Computer Applications(IJCA, USA), Vol 52-No.2, Pg. 41-46(2012).