Matrix and Mutation Based Cryptosystem (MMC)

Rajdeep Chowdhury [1], Arijit Saha [2], Pratip Kumar Biswas [3] and Arijit Dutta [4]
[1]Assistant Professor, Department of Computer Application, JIS College of Engineering, Kalyani,
Nadia-741235, West Bengal, India
[2,3,4] Students, Department of Computer Application, JIS College of Engineering, Kalyani,
Nadia-741235, West Bengal, India

Summary

In this paper, both encryption and decryption methodologies have been proposed and coined as "Matrix and Mutation Based Cryptosystem (MMC)." It commences with the character in the (2, 2) position of the matrix and then transformed into next 9 characters followed by conversion into hexadecimal equivalent. After that, transposing of the matrix is followed by row shift and column shift. Then, the number at position (1, 1) is taken and transformed into binary and genetic function MUTATION is used at bit position 2, 4, and 6. Next, it is converted into decimal. This decimal numbers are used to mix / mingle the colors in the standard color palette in RGB format.

During decryption methodology, the color is selected and is opened using the standard color palette. Value of R is taken and converted into binary, genetic function MUTATION is applied at position 2, 4, 6 and converted into hexadecimal equivalent. Again we create 3x3 matrices and put the hexadecimal number at position (1, 1). Next we use the key (-6 from (1, 1), -3 from (1, 1)) for column and (-2 from (1, 1), -1 from (1, 1)) for row. Next, we shift the columns and rows and transpose the matrix to get / decipher the original text.

A comparison of the proposed technique with existing algorithm Triple-DES have been done in encryption & decryption time and non-homogeneity of source and encrypted files.

Keywords:
Standard Color palette, Matrix, Mutation, Cryptosystem, Encryption, Decryption.

1. Introduction

Information security has become a critical aspect of 21st century's computing systems. In this era, with the help of networking, each and every computer is connected to one another virtually [5]. So, at this point/juncture of time, maintenance of secrecy and security of information has become a necessity [9]. Due to this, modern day researchers are working on different kind of encryption and decryption methodologies for transferring data safely from one point to the other [7]. The algorithm combines and conjures up the features of matrix transposition and shifting of rows and columns, along with hexadecimal number system. The features of the genetic function MUTATION is used along with the standard color palette.

2. The Scheme

Encryption:

Step 1: Creating the matrix, converting into hexadecimal equivalent, transposing and shifting rows and columns.
Example: Here we take the plain text to be "IJCSNS" and establish via case study.

Matrix "I"

0	0	0
0	I	0
0	0	0

J	K	L
M	N	O
P	Q	R

4A	4B	4C
4D	4E	4F
50	51	52

4A	4D	50
4B	4E	51
4C	4F	52

Manuscript received March 5, 2011
Manuscript revised March 20, 2011

4C	4F	52
4A	4D	50
4B	4E	51

⇩

52	4C	4F
50	4A	4D
51	4B	4E

Matrix “J”

0	0	0
0	J	0
0	0	0

⇩

K	L	M
N	O	P
Q	R	S

⇩

4B	4C	4D
4E	4F	50
51	52	53

⇩

4B	4E	51
4C	4F	52
4D	50	53

⇩

4D	50	53
4B	4E	51
4C	4F	52

⇩

53	4D	50
51	4B	4E
52	4C	4F

Matrix “C”

0	0	0
0	C	0
0	0	0

⇩

D	E	F
G	H	I
J	K	L

⇩

44	45	46
47	48	49
4A	4B	4C

⇩

44	47	4A
45	48	4B
46	49	4C

⇩

46	49	4C
44	47	4A
45	48	4B

⇩

4C	46	49
4A	44	47
4B	45	48

Matrix "S"

0	0	0
0	S	0
0	0	0

↓

T	U	V
W	X	Y
Z	[	\

↓

54	55	56
57	58	59
5A	5B	5C

↓

54	57	5A
55	58	5B
56	59	5C

↓

56	59	5C
54	57	5A
55	58	5B

↓

5C	56	59
5A	54	57
5B	55	58

Matrix "N"

0	0	0
0	N	0
0	0	0

↓

O	P	Q
R	S	T
U	V	W

↓

4F	50	51
52	53	54
55	56	57

↓

4F	52	55
50	53	56
51	54	57

↓

51	54	57
4F	52	55
50	53	56

↓

57	51	54
55	4F	52
56	50	53

Matrix “S”

0	0	0
0	S	0
0	0	0

⇩

T	U	V
W	X	Y
Z	[	\

⇩

54	55	56
57	58	59
5A	5B	5C

⇩

54	57	5A
55	58	5B
56	59	5C

⇩

56	59	5C
54	57	5A
55	58	5B

⇩

5C	56	59
5A	54	57
5B	55	58

Step 2: Next, we select (1, 1) position of our 6 matrices. Convert them into binary and apply genetic function MUTATION at 2nd, 4th and 6th position. The mutated binary number is converted into decimal. At last, we change this decimal numbers into colors, which is the encrypted pattern. Now, continuation is done via example.
Hexadecimal numbers at position (1, 1)
(52, 53, 4C, 5C, 57, 5C)

Hexadecimal	Binary	Mutated Binary	Decimal	Color Code
52	01010010	00000110	6	■
53	01010011	00000111	7	■
4C	01001100	00011000	24	■
5C	01011100	00001000	8	■
57	01010111	00000011	3	■
5C	01011100	00001000	8	■

Cipher Text or Cipher Pattern: To get the color of the particular box, the value of R=6, G=7, B=24 are taken. To get the color of the second number, the value of R=7, G=24, B=8 are taken. This process will continue and will rollover from last to first position when required.

■ ■ ■ ■ ■ ■

This is our cipher pattern.

Exception: As a color sequence has been used here, there will be contradiction on places where all the given characters will be same. Therefore, an algorithm has been devised / designed that will stop this contradiction. The algorithm will check the consecutive numbers and if four consecutive numbers are found same then after forming the first color, when the second color will be formed, then the value of RGB will be changed in such a manner so that the color does not match with the color before it. For example, if all the characters are **NNNNee** then we get sequence (3, 3, 3, 3, 58, 58). Now, according to the proposed algorithm, such a sequence will not be allowed. The value will be adjusted in such a way that the color changes completely.
It can be observed that next the sequence is same as the before. So, the value of RGB will be changed. The value which will either be added or subtracted will be decided by the user. So, here it is chosen to add 150 with R, 160 with G and 170 with B. Hence, we get the value of R=153, G=163 and B=173.

■ ■ ■ ■ ■ ■

During the decryption process, the extra value which is either added or subtracted will be changed accordingly.

Decryption:

Cipher Pattern:

■ ■ ■ ■ ■ ■

Step 1:

Color Code	Decimal	Binary	Mutated Binary	Hexadecimal
■	6	00000110	01010010	52
■	7	00000111	01010011	53
■	24	00011000	01001100	4C
■	8	00001000	01011100	5C
■	3	00000011	01010111	57
■	8	00001000	01011100	5C

The values are put, from where they were taken, that is, from (1, 1) position of the matrix.

To get the values in the blank position, hexadecimal subtraction is being done. Subtract 6 from (1, 1) to get value of (1, 2) and subtract 3 from (1, 1) to get value of (1, 3). To get the value of (2, 1), subtract 2 from (1, 1) and to get value of (3, 1), subtract 1 from (1, 1). So, the **key text is (-6 -3 for column and -2 -1 for row).** Without this key, decryption of the given characters is impossible.

To get the other remaining values, any one of the above mentioned methodology can be used

Step 2: The value in the remaining matrix is calculated and then just reverse of the process that we followed during encryption is ensured.

Here, an example is cited for decryption as well, which ensures everyone the methodology adhered to retrieve the original text.

Matrix “52”

52	**52-6=4C**	**52-3=4F**
52-2=50	**50-6=4A**	**50-3=4D**
52-1=51	**51-6=4B**	**51-3=4E**

↓

4C	**4F**	**52**
4A	**4D**	**50**
4B	**4E**	**51**

↓

4A	**4D**	**50**
4B	**4E**	**51**
4C	**4F**	**52**

↓

4A	**4B**	**4C**
4D	**4E**	**4F**
50	**51**	**52**

↓

J	**K**	**L**
M	**N**	**O**
P	**Q**	**R**

↓

0	**0**	**0**
0	**I**	**0**
0	**0**	**0**

In this methodology, the entire original text (plain text), “IJCSNS” is being retrieved.

3. Results and Analysis

In this section, the proposed algorithm is being analyzed with different kinds of file.

It is being tested on Text files (.txt), Executable files (.exe) and Dynamic Link Libraries (.dll).

3.1Text file(s) analysis:

Five text files are being sampled / taken for testing. The encryption time, the decryption time and source file sizes are noted for T-DES, GFC and MMC.

Table 1 shows that for any file, the proposed MMC takes less time to encrypt and decrypt data than T-DES and takes less or same amount of time compared to GFC.

Table 1
File size v/s Encryption and Decryption time for .txt files (For T-DES, GFC & MMC algorithm)

Source file name (.txt)	Source file size (bytes)	Encryption time (second)			Decryption time (second)		
		T-DES	GFC	MMC	T-DES	GFC	MMC
Adcajavas	629	~0	~0	~0	~0	~0	~0
License	7168	2	~0	~0	2	~0	~0
Oledbjvs	10240	3	~0	~0	3	~0	~0
NeroHistory	17408	10	1	1	10	1	1
Nero	33792	12	1	1	12	1	1

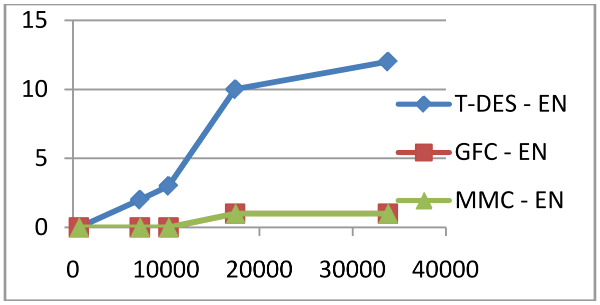


Figure 1
Encryption time for T-DES, GFC and MMC (For text files)

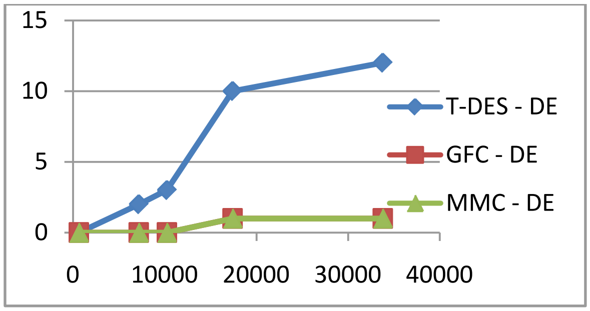


Figure 2
Decryption time for T-DES, GFC and MMC (For text files)

3.2Executable file(s) analysis:
Five executable files of different sizes are being sampled / taken for testing. Time taken for these files to be encrypted and decrypted by T-DES is more than MMC, while compared to GFC, MMC is less or same.

Table 2
File size v/s Encryption and Decryption time for .exe files (For T-DES, GFC & MMC algorithm)

Source file name (.exe)	Source file size (bytes)	Encryption time (second)			Decryption time (second)		
		T-DES	GFC	MMC	T-DES	GFC	MMC
Vlc	96256	44	1	1	45	1	1
PINBALL	355328	83	5	4	82	5	4
Dialer	613376	151	11	10	150	12	11
ImageDrive	775168	190	15	14	174	26	26
winamp	1307648	326	25	24	325	26	25

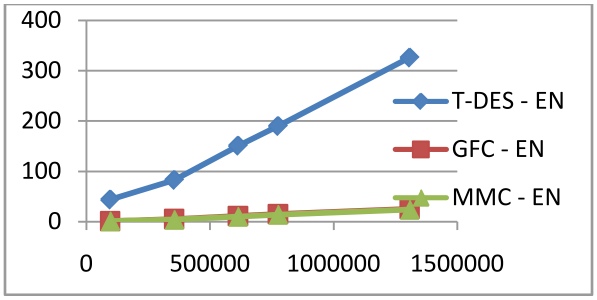


Figure 3
Encryption time for T-DES, GFC and MMC (For executable files)

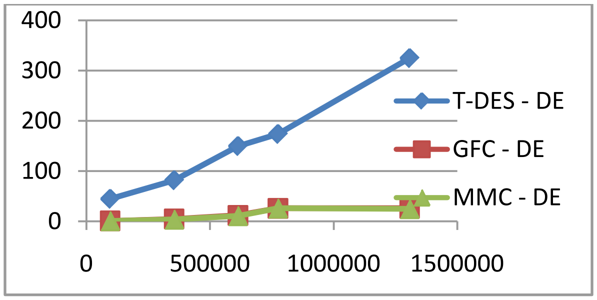


Figure 4
Decryption time for T-DES, GFC and MMC (For executable files)

3.3Analysis of DLL file(s):
Five dll files of different sizes are being sampled / taken for testing. It can be observed that T-DES takes more time compared to MMC, while compared to GFC, MMC takes less or same time to encrypt / decrypt files.

Table 3
File size v/s Encryption and Decryption time for .dll files (For T-DES, GFC & MMC algorithm)

Source file name (.dll)	Source file size (bytes)	Encryption time (second)			Decryption time (second)		
		T-DES	GFC	MMC	T-DES	GFC	MMC
Tataki	65536	17	1	1	18	1	1
Wmpband	98304	25	1	1	23	1	1
7zxa	169984	41	1	1	41	1	1
Wmpns	221184	52	2	1	54	2	1
Mpvis	368640	87	3	2	87	3	2

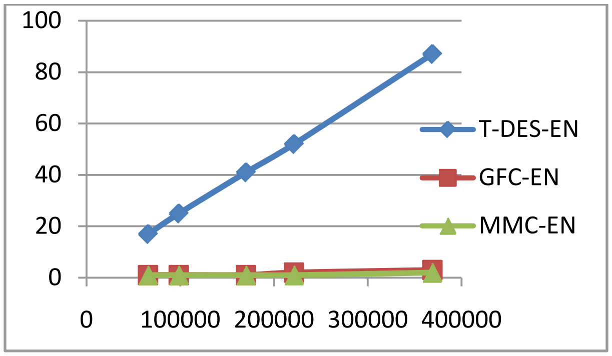


Figure 5
Encryption time for T-DES, GFC and MMC
(For dynamic link library files)

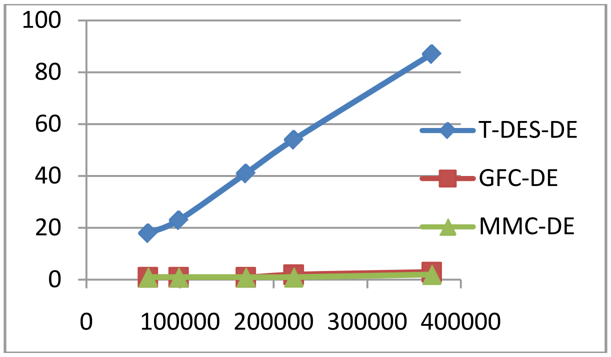


Figure 6
Decryption time for T-DES, GFC and MMC
(For dynamic link library files)

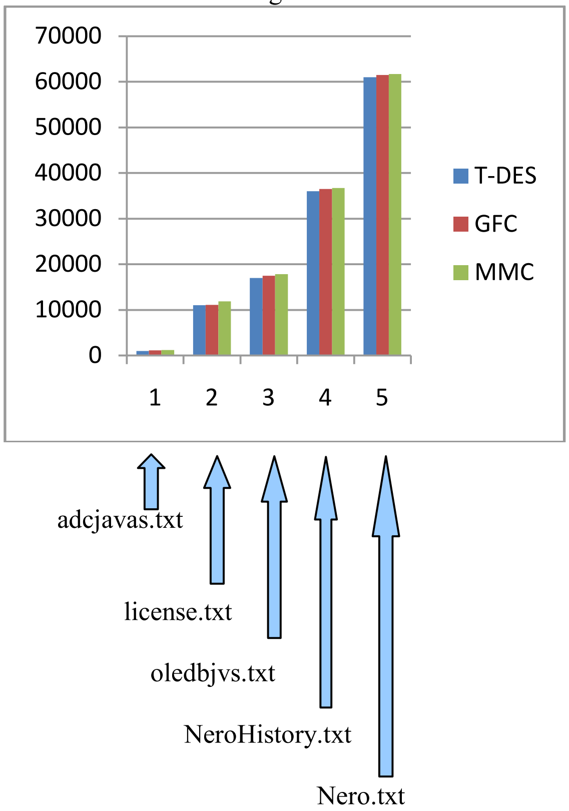
3.4Tests for homogeneity:

The well accepted parametric tests have been performed to test the non-homogeneity between the source files and the encrypted files. The Chi Square values might confirm the heterogeneity of the source files and the encrypted files. The Chi Square test has been performed using source files and encrypted files for the proposed MMC technique, the pre-proposed GFC technique and pre-established T-DES technique.

It is a known fact that for non-homogeneity, the chi-square value should be increasing as the file size increases. Five files of different types are taken. The high chi-square value ensures the non-homogeneity between the source files and the encrypted files. In all the three cases, a good degree of non-homogeneity has been attained for MMC technique. So, it can be said that the proposed MMC technique might ensure optimal security in transmission.

Figure 7 is given below, establishing the context referred in the section above:

Figure 7



4. Conclusion

The proposed technique "Matrix and Mutation Based Cryptosystem" is very simple and easy to use encryption & decryption technique. The color code is the unique feature provided with the cryptosystem. The proposed algorithm conjures up the very nuances and facets of generalized cryptosystem and utilizes it to establish concealment of data and its authenticity. The result shows that the source files and the encrypted files are non-homogeneous, which is established by the chi-square test.

5. Future Scope

The proposed cryptosystem can be enhanced further by using more complex techniques in matrix transformation as well as using the genetic functions in a more detailed and complicated way. The essence of the proposed technique can be emphasized with concise and steady enhancement, ensuring the originality in concealment of the data.

6. References

1. Som S., Mandal J.K., and Basu S. "A Genetic Functions Based Cryptosystem (GFC)", International Journal of Computer Science and Network Security, Volume-9, Number-9, September 2009.
2. Chowdhury R., and Ghosh S. "Normalizer Based Encryption Technique (NBET) Using the Proposed Concept of Rubicryption", International Journal of Information Technology and Knowledge Management, Volume-4, Number-1, January-June 2011.
3. Chowdhury R., and Ghosh S. "Study of Cryptology Based on Proposed Concept of Cyclic Cryptography Using Cyclograph", Research Journal of Engineering and Technology, Volume-2, Number-1, March 2011.
4. Som S., Mitra D., and Halder J. "Secure-Bit Rotate and Swapped Encryption Technique (SBRSET)", National Conference on Trends in Modern Engineering System (IConTiMES 2008), February 2008.
5. Kahate, A. "Cryptography and Network Security", Tata McGraw-Hill, 2nd Edition.
6. Gottfried, B.S. "Programming with C", Tata McGraw-Hill, 2nd Edition.
7. Stallings, W. "Cryptography and Network Security", Prentice Hall, 3rd Edition.
8. Balagurusamy, E. "Programming with Java", Tata McGraw-Hill, 3rd Edition.
9. Fadia, A. "Network Security", Macmillan India Limited.

Mr. Rajdeep Chowdhury is Assistant Professor, Department of Computer Application, JIS College of Engineering affiliated to West Bengal University of Technology. He has completed Master in Computer Application [MCA] from JIS College of Engineering, Kalyani under West Bengal University of Technology. His fields of interest are Cryptography, Network Security, Strategic Management, Database Management System and Data Mining. He has finished several courses related to Computer Application, Software Engineering and Project Management, Object Oriented Design and Analysis. He has accumulated 4 years of teaching experience. He has published several Research Papers in referred National/ International journals. He has referred presentations in National / International Conferences and Seminars.

Mr. Arijit Saha is currently pursuing Bachelor of Computer Application [BCA] from JIS College of Engineering, Kalyani. His fields of interest include Cryptography, Java and Database Management System.

Mr. Pratip Biswas is currently pursuing Bachelor of Computer Application [BCA] from JIS College of Engineering, Kalyani. His fields of interest include Cryptography, Internet Security, Java and Web Designing.

Mr. Arijit Dutta is currently pursuing Bachelor of Computer Application [BCA] from JIS College of Engineering, Kalyani. His fields of interest include Cryptography, Java and Database Management System.