A Novel Paradigm for Symmetric Cryptosystem

Shadi R. Masadeh
Faculty of Information Technology
Isra University
Amman,
Jordan

Hamza A. Al_Sewadi
King Hussein Faculty of Computing
Prince Sumaya University for
Technolog
Amman, Jordan

Mohammad A. Wadi
Independent researcher
Amman,
Jordan

***Abstract*—Playfair cipher is the first known digraph polyalphabetic method. It relies on 5x5 uppercase alphabets matrix with simple substitution processes to be used for encryption and decryption. This paper proposes an enhanced variant of Playfair cipher algorithm that incorporates an algorithm for elaborate key generation starting with a seed accompanying the ciphertext and will be referred to as a Novel Paradigm for Symmetric Cryptosystem (NPSC).**

The key generation, encryption and decryption processes implement modular calculations instead of the simple substitution used in the traditional Playfair cipher. It supports both alphabetic characters and numerals. This variant considerably enhances the security strength without increasing the matrix size as demonstrated by the experimentation results. Comparative studies of various critical factors with other reported versions of Playfair cipher and results were also included.

Keywords—cryptography; security; symmetric systems; polyalphbetic cipher; key generation

I. INTRODUCTION

The first practical digraph substitution cryptosystem was Playfair cipher which is invented by Charles Wheatstone in 1854. It encrypts pairs of letters rather than single letters as in simple substitution cipher, hence reducing the hazard of frequency analysis attack significantly. Playfair cipher is reasonably fast and easy to be applied by low skilled persons, hence it was tactically used during World Wars I & II by all warring parties. Due to its powerful then and simple calculations, it proved suitable for the protection of non-critical information during combat. By the time the enemy cryptanalysts could break the message the information was useless to them [1].

The traditional Playfair cipher structure is a 5x5 matrix that contains the uppercase English alphabet with letters I and J are treated as equals. Uncounted books, papers and surveys included the structure, operation and characteristics of this ciphering technique were reported [2-8]. Due to its many peculiarities and shortcomings outlined by Srivastava and Gupta [9], many extended or modified versions of Playfair cipher schemes that would include alphanumeric and special characters were suggested in recent years as listed in section 2. However, this paper presents a new variant that incorporates modular arithmetic calculations rather than simple substitution and involves alphabetic and numerals text messages. It suggests a new key creation method out of a key seed for constructing the matrices for the encryption/decryption processes.

After the definition o Playfair Cipher in section 1, a brief survey of related work is given in section 2, followed by a detailed explanation of the proposed cryptosystem scheme in section 3. Section 4 includes the implementation of the algorithm and section 5 lists the experimental results and comparison with the original Playfair cipher and some of its variances. Finally section 6 concludes the paper.

II. PLAYFAIR CIPHER VARIANTS

The most important variants of Playfair cipher will be chronically considered here.

Murali and Senthilkumar [10] reported a rapid increase in security of the transmission over an unsecured channel by mapping random numbers to keyword of Playfair cipher. The corresponding numbers are then transmitted to the recipient instead of alphabetical letter.

Sastry et. al. [11] considered the 7 bits ASCII characters representation for the plaintext message characters denoted by codes from 0 to 127. Shanon's concept of confusion and diffusion was achieved by suitable variation in the traditional Playfair rules together with modification in the substitution tables.

Babu et. al. [12] implemented 6x6 matrix instead of 5x5 matrix in order to include number in their cipher, however, lowercase letters, white space and other printable characters cannot be handled. In order to include other uppercase letters and other characters, Srivastava and Gupta [9] modified their work by forming an 8 x 8 matrix. Beside they extended the technique by converting obtained cipher to the corresponding ASCII code values in decimal and further to corresponding 7 bits binary values, then applying Linear Feedback Shift Register to obtain the final ciphertext.

Agrawal et. al. [13] also used 5x5 matrix with consideration of letters I and J, but the frequency of digraph in the message is calculated first and the two letters with the lowest frequency were combined in the matrix immediately after the letters of the keyword.

Tunga and Mukherjee [14] extended the matrix to 16x16 in a multiple array structure in order to facilitate for information regarding spaces and the inclusion of "X" in the alphabet matrix. Besides it incorporated a shifting mechanism for rows and columns of the matrix to ensure that the encrypted text contains any ASCII code ranging between 0 – 255. Another 16x16 Matrix Playfair variant is also suggested by Dhenakaran and Ilayaraja [15].

Basu and Ray [16] implemented a rectangular 10x9 matrix in order to support almost all the printable characters including white space, and therefore increasing the size of the key domain to 90! and hardening the cryptanalysis tasks.

Hans et al. [17] implemented pattern swapping to produce a multiple key changes up to a maximum of fifty times. He adopted Random pattern of eight digits containing only numbers from 1-4.

Chand and Bhattacharyya [18] added to the 6x6 matrix of Babu et. al. [12] four iteration steps. They included letters and numbers from 0 to 9 without combining I and J in the same cell, hence user can write messages with all alphanumeric characters.

III. The Novel Paradigm for Symmetric Cryptosystem (NPSC)

An enhanced cryptographic scheme (NPSC) is proposed in this work which is inspired by Playfair cipher that can encrypt alphanumeric messages. However it follows completely different and elaborate procedure. It adopts an elaborate key creation method and consists of two encryption/decryption algorithms and relies on modular arithmetic calculation for key generation and cryptographic processes. Two 5x5 matrices were employed as the backbone for this scheme, one for the alphabetic characters and the other for numerals. The key creation algorithm starts with a key seed that is chosen by the sender and attached to the encrypted message in order to produce these matrices, and then they will be used for message encryption at the sender side and decryption of the received cipher message at the receiver side. Details of all algorithms will be listed below.

A. Key generation Algorithm:

The key generation of the key starts with a seed that must consist of reasonable number both letters and numbers decided by the sender and embedded as a header for the ciphertext message. Its length is defined by certain agreement between the communicating parties. This seed is used independently by both sending and receiving parties to create the key according to properly designed process, and then produce two 5x5 matrices; one for the text and the other for the numerals. The first step for the key creation is to split the seed in two strings; text string and numerals string. Then text string is treated with algorithm-A of Fig 1 to create the first matrix and the numerals string is treated with algorithm-B of Fig 1 to create the second matrix.

Algorithm-A: for the text key string.

1. *Open a file for the text string*
2. *Remove duplicate characters from the text file*
3. *Determine needed characters in the text file until length equals **4, 9** or **16***
4. *Invert text file*
5. *Remove duplicate letters from the text file.*
6. *Repeat step 3until key length equals **4, 9** or **16***
7. *Place the text file in the lower right corner of 5x5 matrix using lower case letters*
8. *Subtract the text's characters from the alphabet, place the result into set **X***
9. *Place contents of **X** in the matrix using lower case letters in clock wise manner.*

Number each matrix position sequentially and save it for later encryption and decryption

Algorithm-B: for the numeral key string.

1. *Open file for numerical string*
2. *Convert each number into a letter using the alphabet array*
3. *Save the new text file*
4. *Remove duplicate letters from the text file*
5. *Add missing letters in the text file until length equals **4, 9** or **16***
6. *Invert text file*
7. *Remove duplicate letters*
8. *Add unused letters to the text file from the alphabet until length equals either 4, 9 or 16*
9. *Place the text file in the upper left corner of 5x5 matrix using upper case letters*
10. *Subtract the text's letters from the alphabet, place the result into **X***
11. *Place **X** in the matrix using upper case letters in clockwise manner*

Number each matrix position sequentially and save it for later encryption and decryption

Fig. 1. The key creation algorithms

The created key length must be either equals 4, 9, or 16 characters. However, if the length of the key seed is not equal to any of these required number of characters, then some characters shall be added. For the algorithm-A, the added letters are determined by eq. 1, so, if the current key contains m characters, then next added character is determined by calculating its location value k first.

$$\mathbf{k} = (\sum_{\mathbf{i}=1}^{\mathbf{m}} \mathbf{p_i} + \mathbf{r}) \bmod 26 \quad \ldots\ldots (1)$$

where p_i is the location value of the i^{th} characters of the current key so far and r = m+1. The character matching the value of k is then added to the key string. This process is repeated until the required key length is achieved. For example if current key length is 7, then 2 more characters need are to be added. When the required key length of characters string is achieved, it is re-written in reverse order and checked for duplicate characters, if found, they are removed and replaced by unused letters from the alphabets in order. Moreover, in the case of obtaining the key with the required number of characters immediately from the message, there will be no need for any more calculations. It should be stated here that letters I and J are considered the same. Obviously, increasing the key length would enhance the security; hence it is recommended to add more characters to the key seed till the next required length is achieved.

A certain pointer is agreed upon by the communicating parties in order to distinguish the key seed from the ciphered

text. This can be either an integer number stating the length of the seed or an agreed upon separator between.

B. Encryption Process

To encrypt a message M which composes of both letters and numerals in the proposed NPSC cryptographic scheme, two possible algorithms are followed depending on the message contents whether it is text or numerical. The characters are encrypted sequentially and taken as ciphertext in the following procedure.

Each character is first identified; if it is a letter then algorithm 1 is used for encryption and if it is an integer number, algorithm 2 is used for encryption. The two algorithms are shown in Fig 2.

The resulting ciphertext string (C) content will be a mixture of lower case and uppercase letters corresponding to letters or numerals, respectively.

Algorithm 1: For letters encryption

1. *Read letter*
2. *Read matrix A.*
3. *Read letter's sequential position in alphabet, call it P*
4. *Read letter's position in matrix A, call it X*
5. *Calculate T = (P+X) mod 26*
6. *Read the letter having the value T in the matrix A and save it as lowercase ciphertext letter in C.*

Go to next character in the message M

Algorithm 2: For numerals encryption

1. *Read numerical*
2. *Read matrix file B*
3. *Read number's sequential position, call it P*
4. *Read number's position in matrix B, place it into X*
5. *Calculate T = (P+X) mod 26*
6. *Read the letter which has the value T in matrix B, and save it as lowercase ciphertext letter in C.*

Go to next character in the message M

Fig. 2. The encryption algorithms

To clarify how this process in executed, a detailed example is shown in the implementation section.

C. Decryption Algorithm

At the receiving end, the obtained message contains the key seed attached to the ciphered message. For the receiver to decrypt a ciphertext message first the key seed is identified and the removed from the received message in order to be used for creating the two matrices A and B. then the decryption process starts in order to recover the original plain text. Lowercase characters are treated by algorithm A and uppercase letters are treated by algorithm B. and since decryption is the reverse of the encryption process, hence subtraction is used instead of addition. It should be noted here that numbers might be negative; in such case, an addition of 25 is performed in order to acquire the original number.

IV. Implementation

To clarify the proposed NPSC variant, some examples will be included here for the use of the key seed to create the encryption/decryption matrices and to execute encryption and decryption processes. Two matrices are generated one for the letters and one for the numerals.

Example 1: matrices creation:

Suppose the key seed is the word "security5167", then the seed for matrix A is "security" and that for matrix B is "5167". The following will be followed for creating matrix A:

The alphabet is numbered first from 0 to 25, then seed letters are put in a table as in table I-a after removed duplicates, with the target to have 4 or 9 or 16 letters. In this table one extra character is required to get 9 letters. It is calculated in eq. 2.

TABLE I. Determining the Key

s	e	c	U	R	I	t	y	_
1	2	3	4	5	6	7	8	9

(a)

r	y	t	I	u	c	e	s	_
1	2	3	4	5	6	7	8	9

(b)

$$18 + 4 + 2 + 20 + 17 + 8 + 19 + 24 + 9 \bmod 26 = 17 \quad . \; . \; . \quad (2)$$

The value of 17 represents [r] in the alphabet, therefore the obtained key will be "securityr"

Now reversing the letters of this key and removing duplicates produces table I-b which again requires another letter.

Now repeating calculation similar to eq. 2, results into letter [v], hence the key becomes "rytiucesv". It is used to filled the bottom left corner of a 5x5 matrix A, then this matrix is completed with remaining letters of the alphabet without duplication as described in section 3.1 using lowercase letters, see Fig. 3.

1 p	2 q	3 w	4 x	5 z
6 o	7 f	8 g	9 h	10 k
11 n	12 d	13 r	14 y	15 t
16 m	17 b	18 i/j	19 u	20 c
21 l	22 a	23 e	24 s	25 v

Fig. 3. Matrix A

The key seed for the numeral segment in this example is **"5167"**, which will be used for the creation of matrix B. After removing duplicate numbers the will look like table II-a, then replacing the corresponding letters from the alphabet produces table II-b.

TABLE II. Determining the Key

5	1	6	7
1	2	3	4

(a)

F	B	G	H
1	2	3	4

(b)

Then after reversing the letters sequence of table II-b, the key will be "HGBF". These letters will be placed in the upper right corner of a 5x5 matrix B, and will be completed with remaining letters of the alphabet without duplication as described in section 3.1 using uppercase letter, see Fig. 4.

1 H	2 G	3 A	4 K	5 R
6 B	7 F	8 C	9 L	10 S
11 I/J	12 E	13 D	14 M	15 T
16 Q	17 P	18 O	19 N	20 U
21 Z	22 Y	23 X	24 W	25 V

Fig. 4. Matrix A

Example 2: Encrypt the message M = "security5167"

Solution:

s: 24 + 1 = 25 which is [v]
e: 23 + 2 = 25 which is [v]
c: 20 + 3 = 23 which is [e]
u: 19 + 4 = 23 which is [e]
r: 13 + 5 = 18 which is [i]
i: 18 + 6 = 24 which is [s]
t: 15 + 7 = 22 which is [a]
y: 14 + 8 = 18 which is [a]
5 which equal F: 7 + 9 = 16 which is [Q]
1 which equal B: 6 + 10 = 16 which is [Q]
6 which equal G: 2 + 11 = 13 which is [D]
7 which equal H: 1 + 12 = 13 which is[D]

Therefore the ciphertext is "vveeisaaQQDD".

V. Experimeental Results

The proposed NPSC variant has been programmed in C++ language, and executed on different platforms with the following aspects in mind:

- Execution speed measurement for different package size under various operating environment (Windows 7, Windows Server 2012 and Linux).
- Transmission speed of encrypted messages over various network setups (wired and wireless networks).
- Comparison with playfair cipher.
- All results have been tested on the same environment using Intel Laptop core I .7 processor with 8GB RAM

Therefore, the test included measurements of some important factors, such as CPU run time, power consumption, and the packet transmission time. All measurements were conducted for different package sizes; namely small package (1MB), medium package (10 MB) and large package (1 GB) for two different environments; namely wired network and wireless network. The results were listed in tables III –V and a comparison histogram is plotted for all studies cases in Figures 5-7.

Table III compares the measured factors of the proposed NPSC Cipher with the original Playfair cipher tested on a network using Intel Laptop core i7 processor with windows 7 operating system (OS), and then a histogram plot is drawn in Fig 5. It is shows that all measured factors have improved values in case of NPSC as compared with Playfair Cipher.

TABLE III. Measurement of Various Factors for a Network Using Windows 7 OS

		Wired Network			Wireless Network		
Algorithm Name	Packet Size	CPU Time (ms)	Power Consumption (mw)	Transmission Time (ms)	CPU Time (ms)	Power Consumption (mw)	Transmission Time (ms)
Playfair	Small (1MB)	2.9	5.1	6.7	6.2	11.3	12.8
	Medium (10MB)	14.3	25.2	29.1	21.3	35.6	39.2
	Large (1GB)	20.4	35.4	40.3	31.6	42.1	50.6
NPSC	Small (1MB)	2.1	5.3	5.2	6.5	10.3	11.1
	Medium (10MB)	11.2	22.9	27	21	30	34.2
	Large (1GB)	18.7	33.2	39.5	30.2	41.6	49.1

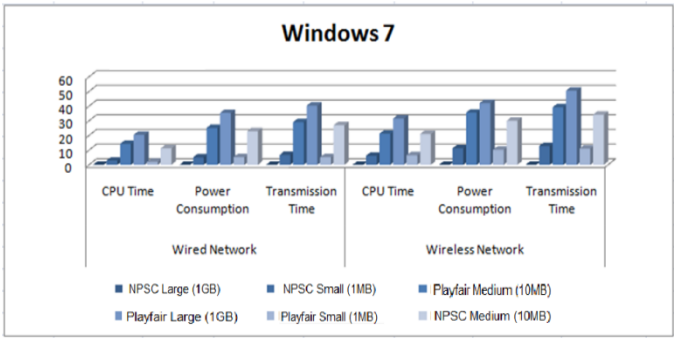

Fig. 5. Comparison of various factors for a network using windows 7 OS

Table IV compares the measured factors of the proposed NPSC Cipher with the original Playfair cipher tested on a network using windows Server 2012 operating system, and a histogram plot is drawn in fig 6. It is also shows that all measured factors have improved values in case of NPSC as compared with Playfair Cipher.

TABLE IV. Measurement of Various Factors for a Network Using Windows Server 2012 OS

		Wired Network			Wireless Network		
Algorithm Name	Packet Size	CPU Time (ms)	Power Consumption (mw)	Transmission Time (ms)	CPU Time (ms)	Power Consumption (mw)	Transmission Time (Mbps)
Playfair	Small 1MB	3.7	10.6	9.5	7.4	17.2	12.5
	Medium 10MB	16.9	26.1	35.9	24.2	38.4	40.2
	Large 1GB	23	45.6	50.6	37.6	52.1	59.2
NPSC	Small 1MB	3.5	10.0	7.4	6.5	16.1	12
	Medium 10MB	15.5	27.6	30.2	20.3	34.6	34.7
	Large 1GB	22.7	43.3	45.3	33.1	50.2	51.1

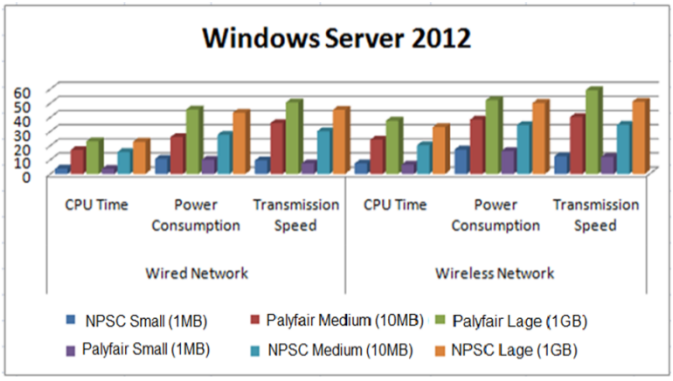

Fig. 6. Comparison of various factors for a network using windows Server 2012 OS

Table V compares the measured factors of the proposed NPSC Cipher with the original Playfair cipher tested on a network using Linux operating system, and a histogram plot is drawn in fig 7. Again it is shows that all measured factors have improved values in case of NPSC as compared with Playfair Cipher.

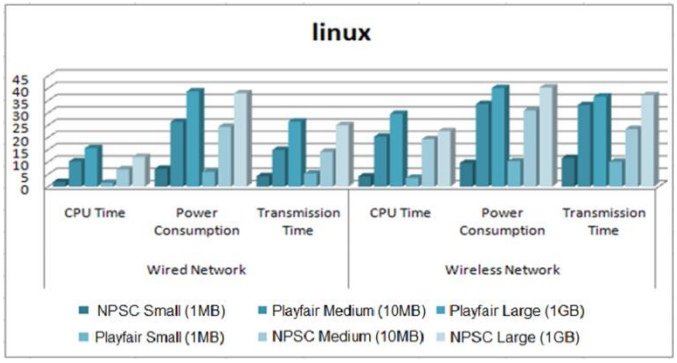

Fig. 7. Comparison of various factors for a network using Linux OS

TABLE V. MEASUREMENT OF VARIOUS FACTORS FOR A NETWORK USING LINUX OS

		Wired Network			Wireless Network		
Algorithm Name	Packet Size	CPU Time (ms)	Power Consumption (mw)	Transmi-ssion Time (Mbps)	CPU Time (ms)	Power Consumption (mw)	Transmission Time (Mbps)
Playfair	Small (1MB)	1.9	7.2	4.2	4.1	9.6	11.6
	Medium (10MB)	10.2	26.3	14.9	20.3	33.6	33.2
	Large (1GB)	15.6	38.7	26.4	29.6	40.1	36.6
NPSC	Small (1MB)	1.5	6	5.3	3.5	10.3	10.1
	Medium (10MB)	7	24.3	14.2	19.3	31	23.4
	Large (1GB)	12	38	25	22.5	40.3	37.2

Despite the elaborate computation involved in the proposed NPSC design which includes modular mathematics rather than only substitution as for the original Playfair cipher, the observed empirical results have all demonstrated improvements in algorithm execution time, power consumption and transmission speed over computer networks for different platforms.

The security of NPSC can be also compared with variant of playfair cipher, as shown in table VI.

VI. SECURITY ISSUES OF NPSC CIPHER

Any cryptosystem is designed to stand cryptanalysis attacks relying on time complexity and space complexity. The original Playfair Cipher is vulnerable to Brute force attack but has reasonable resistance to frequency analysis attack. For the NPSC cipher, these two attacks may be considered here.

A. *Brute Force Attack* [2]:

This attack systematically attempts all possible key combination; hence, larger the key space results into more secure cipher. In NPSC cipher, two independent 5x5 matrices were for encryption and decryption purpose. Therefore, the key space for building these matrices will be (26x26)x(26x26) resulting into 456976 different possible combinations for the Brute force attack as compared with 26x26 diagrams for Playfair cipher.

B. *Frequency Analysis*

Frequency analysis is the study of the frequency of occurrence for letters or groups of letters in a ciphertext [20]. It is based on the statistical fact that, each letter or group of letters for any language has certain frequency of occurrence. This frequency would be decided with more accuracy if the written text was of considerable size. The characteristic distribution of these letters is found to be almost the same for any stretch of text of reasonable length [21]. However, the proposed NPSC cipher relies mathematical calculations and not digraph substitution; a thing put the frequency attack out of the question.

C. *Comparison Analysis:*

A comparison study is conducted for the proposed NPSC cipher in contrast with the original Playfair cipher and some of its reported variants and listed in table VI [8]. It includes the key space that is available for Brute force attack, the number of possible diagrams need to be searched for ciphertext only attack, and the probability of occurrence of an element for frequency analysis attack.

TABLE VI. PLAYFAIR VARIANT COMPARISON

Playfair Cipher	Key space for Brute force attack	Number of diagrams to be searched for ciphertext only attack	Probability of occurrence of an element for frequency analysis attack
Original	25!	676	0.038
Srivastava & Gupta [9]	64!	4096	0.016
Babu et al [12]	36!	1296	0.028
Hans et al [17]	26!*24*24	Difficult	Difficult
Chad et al [17]	36!	1296	0.028
Verma et al [19]	64!	4096	0.016
Proposed NPSC	36!	456976	Difficult

VII. CONCLUSIONS

This paper introduced few variations to Playfair ciphers such as using numerals together with letters, creating the encryption/decryption key from an alphanumeric seed, and the expansion of the encryption/decryption matrices to two instead of one matrix. Also the encryption/decryption processes are performed using modular arithmetic.

These added values to the ciphering technique have given larger key domain size for brute force attack and increased number of diagrams needs to be searched for ciphertext only attack and also handicapped the frequency of occurrence analysis attack. More work is needed to be pursued to study the avalanche effect on this cipher.

REFERENCES

[1] Wikipedia (http://en.wikipedia.org/wiki/Playfair_cipher), last visited Oct. 2015.

[2] W. Stalling, "Cryptography and network security: Principle and practice", 5th ed, Pearson Education, 2011.

[3] D. Bruff, "The Playfair cipher revealed wynne", MLAS 280-07 Cryptography July 13, 2009.

[4] E. Baldwin and D. Bruff," Playfair cipher", FYWS Cryptology October 27, 2010. <http://derekbruff.org/blogs/fywscrypto/files/2010/11/Baldwin-Essay-2.pdf>

[5] K. R. Babu, S.Udaya Kumar and A.V. Babu, "A survey on cryptography and steganography methods for information security", International Journal of Computer Applications (0975 – 8887), Vol. 12, No.2, November 2010,

[6] "The Playfair algorithm description", http://macliang.acns.carleton.edu/ falk /other/ playfair.htm.

[7] M. Kumar, R. Mishra, R. K. Pandey and P. Singh, "Comparing classical encryption withmodern techniques", proceedings of S-JPSET, Vol. 1, No. 1, 2010.

[8] P. Goyal, G. Sharma and S. S, Kushwah, "Network security: A survey paper on Playfair cipher and its variants", International Journal of Urban Design for Ubiquitous Computing Vol. 3, No.1, 2015, pp.1-6. http://dx.doi.org/10.14257/ijuduc.2015.3.1.01

[9] S. S. Srivastava and Nitin Gupta, "Novel approach to security using extended Playfair cipher", International Journal of Computer Applications (0975 – 8887), Vol 20, No.6, April 2011.

[10] P. Murali and G. Senthilkumar, "A modified version of Playfair cipher using linear feedback shift register", International Journal of Computer Science and Network Security (IJCSNS), Vol.8, No.12, December 2008.

[11] U. Sastry, N. R. Shankar and S. D. Bhavani, "A modified Playfair cipher involving interweaving and iteration", International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December, 2009, 1793-8201.

[12] K. R. Babu, U. Kumar, A. V. Babu, I.V.N.S Aditya and P. Komuraiah, "An extension to traditional Playfair cryptographic method", International Journal of Computer Applications, (0975 – 8887), Vol.17, No.5, March 2011.

[13] G. Agrawal, S. Singh and M. Agarwal, "An enhanced and secure Playfair cipher by introducing the frequency of letters in any plain text", Journal of Current Computer Science and Technology Vol. 1, No. 3, 2011, PP10-16.

[14] H. Tunga and S. Mukherjee, "A new modified Playfair algorithm based on frequency analysis", International Journal of Emerging Technology and Advanced Engineering, Vol. 2, No. 1, January, 2012.

[15] S. S. Dhenakaran and M. Ilayaraja, "Extension of Playfair cipher using 16X16 matrix", International Journal of Computer Applications (0975 – 888) vol. 48, No. 7, June, 2012.

[16] S. Basu and U. K. Ray, "Modified Playfair cipher using rectangular matrix", International Journal of Computer Applications (0975 – 8887) Vol. 46, No.9, May, 2012.

[17] H. Hans, R. Johari and V. Gautam, "An extended Playfair cipher using rotation and random swap patterns," 5th IEEE International Conference on Computer and Communication Technology, 2014.

[18] N. Chan and S. Bhattacharyya, "A Novel approach for encryption of text messages using Playfair cipher 6 by 6 matrix with four iteration steps", International Journal of Engineering Science and Innovative Technology (IJESIT), Vol. 3, No. 1, January, 2014.

[19] V. Verma, D. Kaur, R. K. Singh and A. Kaur, "3D- Playfair cipher with additional bitwise operation", Control Computing Communication & Materials (ICCCCM), 2013 International Conference on IEEE, (2013), August, 2013, PP1-6.

[20] R. P. Dhiren, "Information security theory and practice", 1st ed, Prentice-Hall of India Private Limited, 2008.

[21] Harrison K., B. Munro and T. Spiller, "Security through uncertainty", P Laboratories, February, 2007.