

A Modified Encryption Technique using Playfair Cipher 10 by 9 Matrix with Six Iteration Steps

Subhajit Bhattacharyya¹, Nisarga Chand², Subham Chakraborty³
Mallabhum Institute of Technology, WB, INDIA

Abstract— One of the well-known digraph substitution cipher is the Playfair Cipher. It secures information mathematically by mangling message with key. The privacy of intended sender and receiver information is protected from eavesdropper. However the original 5 x 5 Playfair Cipher can support only 25 uppercase alphabets. Here we have implemented a new technique which includes a rectangular matrix having 10 columns and 9 rows and six iteration steps for encryption as well as decryption purpose. This 10 x 9 rectangular matrix includes all alphanumeric characters and some special characters. Cryptanalysis is done to show that the modified cipher is a strong one. Finally we have implemented this concept with the help of MATLAB.

Index Terms—Playfair cipher, Substitution cipher, Special characters, Cryptanalysis, Symmetric encryption.

I. INTRODUCTION

Cryptography [4] [5] is the science of using mathematics to encrypt and decrypt data. It enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. There are various encryption techniques in today's world. Symmetric key cryptography [9] technique is very useful for encryption process. In symmetric key cryptography, sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message. Symmetric key cryptography is also called the private key cryptography. Playfair cipher [3] is one of the popular symmetric encryption methods.

The first recorded description of the Playfair cipher [8] was in a document signed by Wheatstone on 26 March 1854. However Lord Playfair promoted the use of this cipher and hence it is called Playfair Cipher. It was used by the British in the Second Boer War and in World War I. It was also used by the Australians and Germans during World War II. Playfair is reasonably easy to use and was used to handle important but non-critical secrets. By the time the enemy cryptanalysts could break the message, the information would be useless to them. Between February 1941 and September 1945 the Government of New Zealand used it for communication between New Zealand, the Chatham Islands and the Pacific Islands.

The organization of the paper can be summarized as: The existing playfair algorithm using 5 x 5 matrix explained in Section-II. Limitations of existing playfair cipher discussed

in Section-III, Extended 10 by 9 playfair cipher algorithm explained in Section-IV. Experimental results are shown in Section-V. Future works are discussed in Section-VI. Conclusions are explained in Section-VII.

II. EXISTING PLAYFAIR ALGORITHM USING 5 X 5 MATRIX

The traditional Playfair cipher uses 25 uppercase alphabets. A secret keyword is chosen and the 5 x 5 matrix is built up by placing the keyword without any duplication of letters from left to right and from top to bottom. The other letters of the alphabet are then placed in the matrix. For example if we choose "PLAYFAIREXAMPLE" as the secret keyword the matrix is given in Table 1.

P	L	A	Y	F
I	R	E	X	M
B	C	D	G	H
K	N	O	Q	S
T	U	V	W	Z

Table 1

In this algorithm, the letters I & J are counted as one character. It is seen that the rules of encryption applies a pair of plaintext characters. So, it needs always even number of characters in plaintext message. In case, the message counts odd number of characters a spare letter X is added at the end of the plaintext message. Further repeating plaintext letters in the same pair are separated with a filler letter, such as X, so that the words COMMUNICATE would be treated as CO MX MU NI CA TE.

Rules:

- Plain text letters that fall in the same row of the matrix are replaced by the letter to the right, with the first element of the row circularly following the last. For example RE is encrypted as EX.
- Plain text letters that fall in the same column are replaced by the letter beneath, with the top element of the row circularly following in the last. For example, RC is encrypted as CN.
- Otherwise, each plaintext letter is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, OH becomes SD, and FD becomes AH.

III. LIMITATIONS OF EXISTING PLAYFAIR CIPHER

The main drawback of the traditional Playfair cipher is that the plain text can consist of 25 uppercase letters only. One letter has to be omitted and cannot be reconstructed after decryption. Also lowercase letters, white space, numbers and other printable characters cannot be handled by the traditional cipher. This means that complete sentences cannot be handled by this cipher. Space between two words in the plaintext is not considered as one character. A spare letter X is added when the plaintext word consists of odd number of character. In the decryption process this X is ignored. X is a valid character and creates confusion because it could be a part of plaintext, so we cannot simply remove X in decryption process.

X is used a filler letter while repeating letter falls in the same pair are separated.

In a mono alphabetic cipher the attacker has to search in 26 letters only. Playfair cipher being a polyalphabetic cipher the attacker has to search in $26 \times 26 = 676$ diagrams. Although the frequency analysis is much more difficult than in mono alphabetic cipher still using modern computational techniques the attacker can decipher the cipher text.

To overcome the drawbacks we implement a modified cipher which uses a 10 x 9 matrix which will contain almost all the printable characters.

IV. EXTENDED 10 X 9 PLAYFAIR CIPHER ALGORITHM

This extended play fair algorithm is based on the use of a 10 by 9 matrix of letters constructed using a keyword. The 10 x 9 matrix contains almost all the printable characters. This includes lowercase and uppercase alphabets, punctuation marks, numbers and special characters. The matrix is constructed by filling in the letters, numbers or special characters of the keyword from left to right and from top to bottom, and the filling in the remainder of the matrix with the remaining letters in alphabetic order and digits in ascending order from 0 to 9 and special characters. The upper case alphabets are placed first then the lower case alphabets following the digits 0 to 9 can be placed next cells of the lower case alphabet z in an ascending order. And finally the special characters which are arranged in an order which is shown in Table1-6. In this we have not counted I/J as one letter instead we are placing both I and J in two different cells in order to avoid the ambiguity to the user at the time of decipherment. This algorithm can allow the plain text containing of alpha numeric values; hence the user can easily encrypt alpha numeric values efficiently. The plain text containing contact numbers, date of birth, house numbers and other numerical values can be easily and efficiently encrypted using this algorithm.

A. Assumption

Here we have used six reserved keywords: Monarchy, Duplicate29, Nisarga1987, Subho27, Eagle*& and Shiva@#. Then we construct six 10 by 9 matrices with the help of these six keywords. The six 10 by 9 matrices are shown in figure 1-6.

Keyword: Monarchy

M	o	n	a	r	c	h	y	b	d
e	f	g	i	j	k	l	m	p	q
s	t	u	v	w	x	z	A	B	C
D	E	F	G	H	I	J	K	L	N
O	P	Q	R	S	T	U	V	W	X
Y	Z	0	1	2	3	4	5	6	7
8	9	~	,	.	/	;	“	\	
<	>	?	:	{	}	-	=	!	@
#	\$	%	^	&	*	()	_	+

Fig 1

Keyword: Duplicate29

D	u	p	l	i	c	a	t	e	2
9	b	d	f	g	h	j	k	m	n
o	q	r	s	v	w	x	y	z	A
B	C	E	F	G	H	I	J	K	L
M	N	O	P	Q	R	S	T	U	V
W	X	Y	Z	0	1	3	4	5	6
7	8	~	,	.	/	;	“	\	
<	>	?	:	{	}	-	=	!	@
#	\$	%	^	&	*	()	_	+

Fig 2

Keyword: Nisarga1987

N	i	s	a	r	g	1	9	8	7
b	c	d	e	f	h	j	k	l	m
n	o	p	q	t	u	v	w	x	y
z	A	B	C	D	E	F	G	H	I
J	K	L	M	O	P	Q	R	S	T
U	V	W	X	Y	Z	0	2	3	4
5	6	~	,	.	/	;	“	\	
<	>	?	:	{	}	-	=	!	@
#	\$	%	^	&	*	()	_	+

Fig 3

Keyword: Subho27

S	u	b	h	o	2	7	a	c	d
e	f	g	i	j	k	l	m	n	p
q	r	s	t	v	w	x	y	z	A
B	C	D	E	F	G	H	I	J	K
L	M	N	O	P	Q	R	T	U	V
W	X	Y	Z	0	1	3	4	5	6
8	9	~	,	.	/	;	“	\	
<	>	?	:	{	}	-	=	!	@
#	\$	%	^	&	*	()	_	+

Fig 4

Keyword: Eagle*&

E	a	g	l	e	*	&	b	c	d
f	h	i	j	k	m	n	o	p	q
r	s	t	u	v	w	x	y	z	A
B	C	D	F	G	H	I	J	K	L
M	N	O	P	Q	R	S	T	U	V
W	X	Y	Z	0	1	2	3	4	5
6	7	8	9	~	,	.	/	;	“
\		<	>	?	:	{	}	-	=
!	@	#	\$	%	^	()	_	+

Fig 5

Keyword: Shiva@#

S	h	i	v	a	@	#	b	c	d
e	f	g	j	k	l	m	n	o	p
q	r	s	t	u	w	x	y	z	A
B	C	D	E	F	G	H	I	J	K
L	M	N	O	P	Q	R	T	U	V
W	X	Y	Z	0	1	2	3	4	5
6	7	8	9	~	,	.	/	;	“
\		<	>	?	:	{	}	-	=
!	\$	%	^	&	*	()	_	+

Fig 6

B. Algorithm

- First we take input message which is user defined.
- If any space or punctuations occurs, then it should be automatically removed from the input message.
- After that we check any double occurrence, and then add “X” automatically in between these two characters.
- After removing the unwanted space we get a modified message that is called the digraph message.
- Next we encrypt this digraph message with the Keyword “Monarchy”.
- After that corresponding five iteration steps introduced with five different keywords: “Duplicate29”, “Nisarga1987”, “Subho27”, “Eagle*&” and “Shiva@#”.
- During encryption process if any two character occurs same row or same column and any one of the character occurs at the last column(for same row character) or at the last row(for same column character) then in the encrypted message they becomes first column character(for same row character) or first row character(for same column character).
- Next we decrypt the last encrypted message with keyword “Shiva@#” and repeat the same decryption process five times with five different keywords: “Eagle*&”, “Subho27”, “Nisarga1987”, “Duplicate29” and “Monarchy”.
- During decryption process if any two character occurs

occurs same row or same column and any one of the character occurs at the first column(for same row character) or at the first row(for same column character) then in the encrypted message they becomes last column character(for same row character) or last row character(for same column character).

- From the last stage of the decrypted message we get a message which is same as digraph message.
- This message devoid of space but may include several capital “X”. Some of which may be unnecessary because they are inserted between two same occurrence character or may be inserted at the end of the message to make the message alphabet count even. Some of which may be with the original message. So we have to take only the necessary capital “X” and to discard the unnecessary capital “X”.
- To make the above condition happen we scan the last decrypted message from left to right. If any capital “X” occurs we check the right most and left most character of this “X” if this two character found same we discard the corresponding “X”. If “X” occurs at the last of the string we also have to discard this “X” to recover the original message. For any other condition we have to include the “X” with the original message.
- After removing the unnecessary capital “X” we get our original message.

C. Cryptanalysis

The various types of cryptanalytic attacks are as follows.

1. Brute force attack
2. Cipher text only attack
3. Chosen plaintext/cipher text attack

1. Brute force attack

The size of the key domain is 90! (Factorial 90). Thus brute force attack will be very difficult for the modified Playfair cipher.

2. Cipher text only attack

The frequencies of digrams are preserved in the cipher text (to some extent). The cryptanalyst can launch a cipher-text only attack. However the number of digrams to be searched would be $90 \times 90 = 8100$.

3. Chosen plaintext/cipher text attack

Obtaining the key is relatively straightforward if both plaintext and cipher text are known.

V. EXPERIMENTAL RESULTS

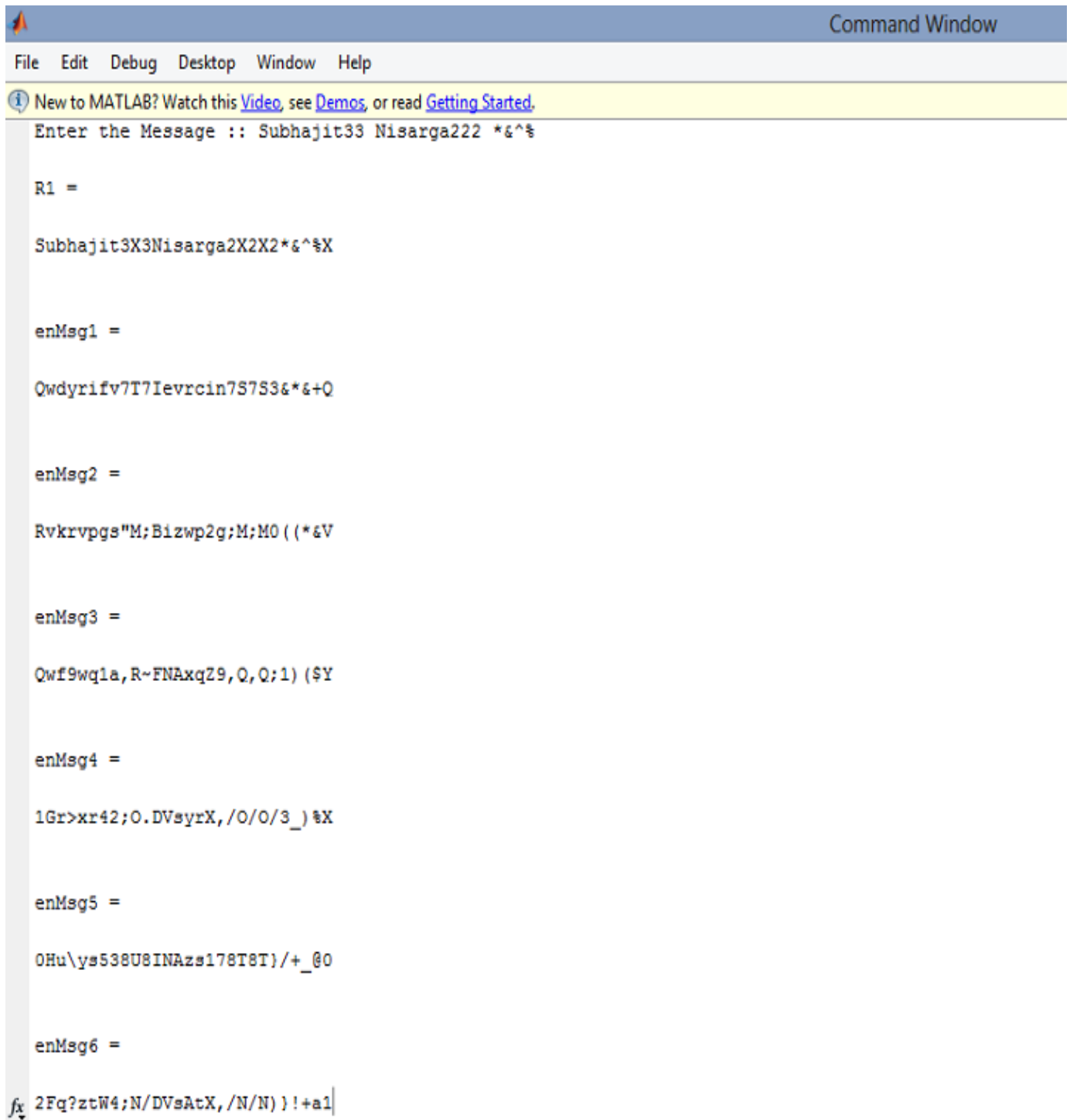
In this thesis for implementation of techniques MATLAB 7.0.2 version is used. MATLAB® is a high-performance language for technical computing.

In our experiment we have used six different keywords and with the help of this six keywords we have encrypt and decrypt the text messages successfully.

Here we include two figures. The original text message with its encrypted six versions is shown clearly in the figure 7

while in the figure 8 the six corresponding decrypted messages with the last original recovered text message is shown below.

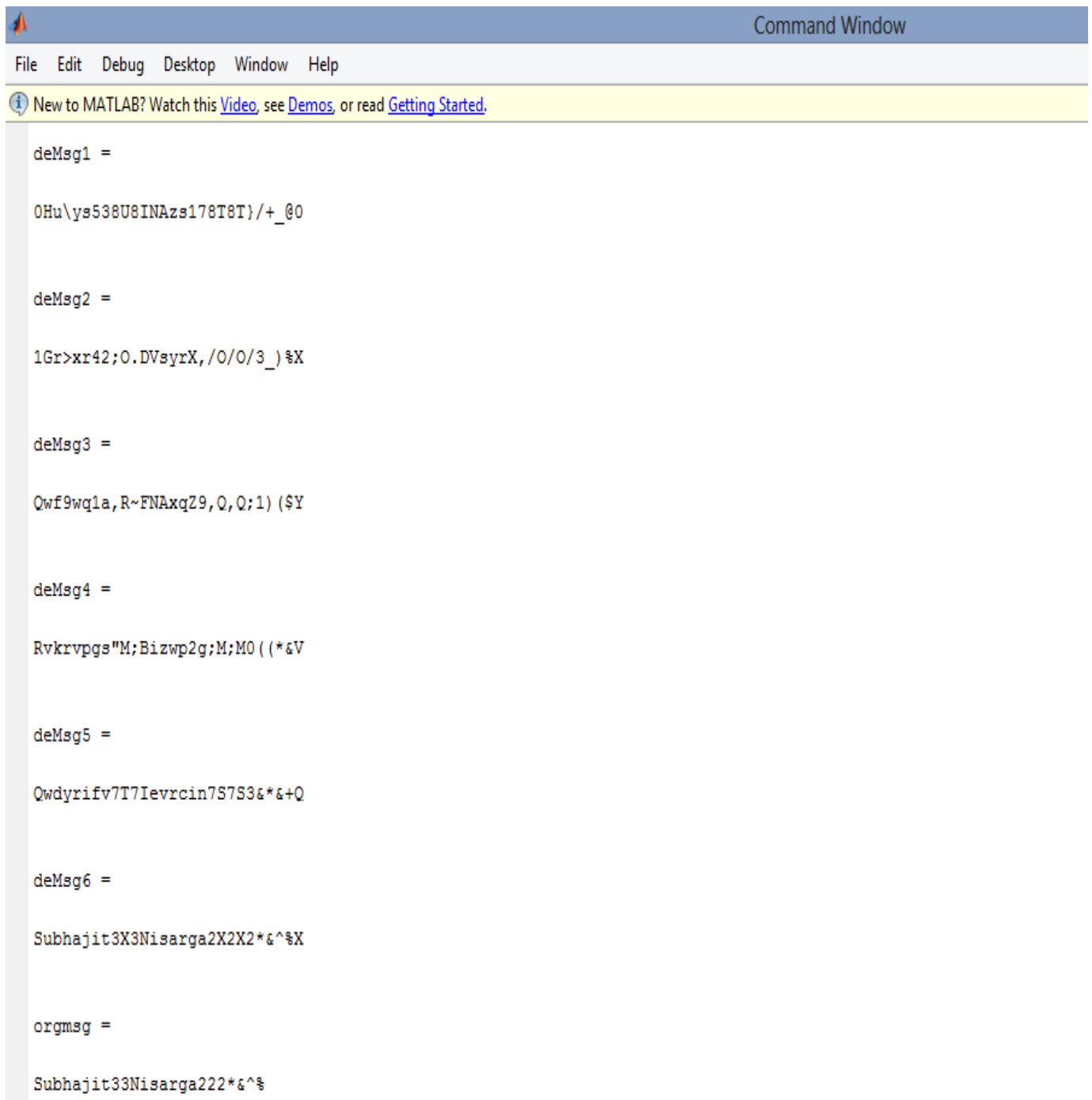
Original text message with its encrypted six versions



```
Command Window
File Edit Debug Desktop Window Help
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Enter the Message :: Subhajit33 Nisarga222 *&^%
R1 =
Subhajit3X3Nisarga2X2X2*&^%X
enMsg1 =
Qwdyrifv7T7Ievrcin7S7S3&*%+Q
enMsg2 =
Rvkrvpgs"M;Bizwp2g;M;M0 (*&V
enMsg3 =
Qwf9wq1a,R~FNAXqZ9,Q,Q;1) ($Y
enMsg4 =
1Gr>xr42;O.DVsyxX,/O/O/3_) %X
enMsg5 =
0Hu\ys538U8INAZs178T8T)/+_@0
enMsg6 =
fx 2Fq?ztW4;N/DVsAtX,/N/N) }!+a1
```

Fig 7

Six decrypted text messages with original received message



```
Command Window
File Edit Debug Desktop Window Help
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

deMsg1 =
0Hu\ys538U8INAs178T8T}/+ _@0

deMsg2 =
1Gr>xr42;0.DVsyxX,/0/0/3_) %X

deMsg3 =
Qwf9wq1a,R~FNaxqZ9,Q,Q;1) ($Y

deMsg4 =
Rvkrvpgs"M;Bizwp2g;M;M0 ( (*&V

deMsg5 =
Qwdyrifv7T7Ievrcin7S7S3*&+Q

deMsg6 =
Subhajit3X3Nisarga2X2X2*^&%X

orgmsg =
Subhajit33Nisarga222*^&%
```

Fig 8

VI. FUTURE WORKS

This extended play fair algorithm is based on the use of six 10 x 9 matrices which include all uppercase and lowercase letters, numeric digits from 0 to 9 and 28 selected special characters. We can extend this concept by using 18 x 12 matrix which includes 154 special characters along with all alpha numeric characters. Finally, we can make this 18 x 12 matrix concept stronger by using several iteration steps.

VII. CONCLUSION

In this paper we have analyzed the merits and demerits of the original playfair cipher. Then we discussed the modified playfair cipher using 10 x 9 matrix. In this matrix we have used all alphanumeric characters as well as some special characters. In this modified playfair cipher six different keys and six iteration steps used to make the encrypted message stronger than the traditional playfair cipher. Finally this concept we have implemented using MATLAB.

REFERENCES

- [1] Derek Bruff, Ph.D, The Playfair Cipher Revealed Wynne MLAS 280-07 Cryptography July 13, 2009.
- [2] Dr. Bruff, Playfair Cipher. FYWS Cryptology October 27, 2010.
- [3] en.wikipedia.org/wiki/Playfair_cipher
- [4] en.wikipedia.org/wiki/Cryptography
- [5] S. Hebert, "A Brief History of Cryptography", an article available at <http://cybercrimes.net/aindex.html>
- [6] X. Du and H. H. Chen, "Security in Wireless Sensor Networks", IEEE Wireless Communications Magazine, Vol. 15, Issue 4, 60-66, 2008.
- [7] Cryptanalysis of substitution cipher chaining mode (SCC) Communications, 2009. ICC '09. IEEE International Conference El-Fotouh, M.A. Inst. for Data Process. (LDV), Tech. Univ. Munchen (TUM), Munich, Germany Diepold, K.
- [8] Behrouz A. Forouzan, Cryptography and Network Security. Special Indian Edition 2007, Tata McGrawHill Publishing Company Limited, New Delhi.
- [9] Atul Kahate, Cryptography and Network Security, 2nd Ed., Tata McGraw-Hill Publishing Company Limited, New Delhi.
- [10] William Stallings, "Cryptography and Network Security: Principles and Practice", 4th Edition, Prentice Hall, 2006.
- [11] V. Umakanta Sastry, N. Ravi Shankar, and S. Durga 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] Shiv Shakti Srivastava, Nitin Gupta "A Novel Approach to Security using Extended Playfair Cipher", International Journal of Computer Applications (0975 – 8887) Volume 20– No.6, April 2011.
- [13] Gaurav Agrawal, Saurabh Singh, Manu 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 Issue 3 [2011]10-16
- [14] Harinandan Tunga, Soumen Mukherjee "A New Modified Playfair Algorithm Based On Frequency Analysis", International Journal of Emerging Technology and Advanced Engineering, (ISSN 2250-2459, Volume 2, Issue 1, January 2012)



Subhajt Bhattacharyya received the Bachelor in Technology degree in Electronics & Communication Engineering and Master in Technology degree in VLSI & Microelectronics from West Bengal University of Technology Kolkata, India and Jadavpur University Kolkata, India from 2007 and 2011 respectively. He is currently an Assistant Professor at Mallabhum Institute of Technology Bishnupur, WB, India.

His research interest includes image processing, computer vision and network security.



Nisarga Chand obtained the Bachelor in Technology degree in Electronics & Communication Engineering and Master in Technology degree in Communication Engineering under West Bengal University of Technology, Kolkata, India from 2009 and 2011 respectively. He is a lifetime member of FOSET (Forum of Scientists, Engineers & Technologists). He is presently serving as an Assistant

Professor in Electronics and Communication Engineering Department at Mallabhum Institute of Technology; Campus: Braja-Radhanagar, P.O: Gosaipur, P.S: Bishnupur, Dist: Bankura - 722122, West Bengal, India. His research interest includes Wireless Sensor Network, Digital Communication and network security.



Subham Chakraborty received the Bachelor and Masters degree in Computer science & Engineering from West Bengal University of Technology Kolkata, India and from 2010 and 2012 respectively. He is currently an Assistant Professor at Mallabhum Institute of Technology Bishnupur, WB, India.

His research interest includes image processing, computer vision and network security.