Novel Modified Playfair Cipher using a SquareMatrix

Harinandan Tunga
Computer Science Department
RCC Institute of Information Technology
Kolkata, India

Arnab Saha
Computer Science Department
RCC Institute of Information Technology
Kolkata, India

Akash Ghosh
Computer Science Department
RCC Institute of Information Technology
Kolkata, India

Swashata Ghosh
Computer Science Department
RCC Institute of Information Technology
Kolkata, India

ABSTRACT

This paper is about encryption and decryption of text using a secret password provided by the user. The encryption machine takes the password and source message as input and generates acipher text based on Modified Playfair Algorithm using dynamic rectangular matrix. The decryption machine takes the same password and the cipher text generated by the encryption engine as input to produce the original message. It also checks if the input is valid and blocks the user when an invalid input is provided. In this paper the entire program will be simulated by using an PHP application.

Keywords

Cryptography, Encryption and Decryption, Modified Playfair Algorithm using dynamic rectangular matrix, Playfair Cipher

1. INTRODUCTION

As information is transferred from one user to another user the data or the information becomes highly vulnerable to all kind of threats caused by adversaries (third party interventions). Now the data communication between two entities can be secured if an encryption and decryption technique is used at two end points. Encryption is done on the sender's side where the message is encrypted with a help of a key (also known by the receiver) and generate a cipher text which is then sent into the network and then on the receiver's end the receiver receives the cipher text and decrypts it back with the help the same key to the original message. Such an encryption technique is Playfair Cipher.

Playfair cipher is a manual symmetric encryption technique. It was invented by the Charles Wheatstone in the year of 1854. The technique bears the name of Lord Playfair who actually promoted the use of the technique during the World War I by British and during World War II by British and Australians.

The technique uses pairs of letters (diagraphs)to encrypt instead of single pairs which are seen in substitution ciphers. Thus makes it harder to break since 600 digraphs (25 X 24 has to be unique) are possible but in case of a simple substitution cipher technique only 26 monographs are possible. So based on frequency analysis breaking Playfair is harder than any substitution cipher techniques.

2. REVIEW OF LITERATURE

A handful of works related to Playfair Algorithm has already been done since Wheatstone on 26 March 1854. However [1]. [2], [5], [6], [7], [8], [10] suggest a keen interest in developing and enhancing the traditional Playfair Algorithm. The Playfair Cipher was first used by the British in Second World War I [10].

Traditional Playfair algorithm uses 5x5 matrix which is built by placing a user given secret keyword and X is used as a padding character and character J is not included in the matrix. For example, **Keywords** is chosen as a secret keyword. Then the matrix will look like

Table 1 Traditional Playfair Matrix 5x5

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

Now a message is taken for an example, **Playfair**. Playfair is broken down into pairs of two such as **Pl, ay, fa, ir.**

The ciphering of the text will be done in correspondence with the matrix as follows.

2.1 In case letters of a diagram are in the same row the letters to the right of each letter are taken. Wrapping happens in case one of the letters is at the last column.

2.2 In case of letters in the same column the letters to the bottom of each letter are taken. Again wrapping happens in case any letter is in the last row.

2.3 In case the letters are neither in the same row or column a rectangle is made with the letters and the letters at the opposite corners are taken.

And if there is a duplication of letter in the message or odd numbers of letters are used for the message then in the either of the cases padding is used with alphabet X and placed between the letters. Since we do not have a duplication of letters in this example we straight way proceed for ciphering the text based on the 3 rules mentioned above. So we get:

PL	QM
AY	SW
FA	HD
IR	CB

So the cipher text is **QMSWHDCB** for **Playfair**

[10]Explains the traditional Playfair algorithm and uses the basic 3 rules of traditional Playfair to encrypt the message on a 10x9 matrix which has almost all the printable characters.

Table 2[10] Modified Playfair 10x9 matrix

D	u	p	l	i	c	a	t	e	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	2	3	4	5	6	7
8	9		,	.	/	;	'	[	]
<	>	?	:	{	}	-	=	!	@
#	$	%	^	&	*	(	)	_	+

[2] Proposed a variation of modified Playfair algorithm by altering the message to its corresponding ASCII characters denoted by the codes from 0-127. Then a substitution table is constructed for further manipulation followed by interweaving and iteration which makes it further complex to crack.

[1] Has proposed another variation of the modified Playfair algorithm which uses 16x16 matrix instead of the traditional 5x5 matrix to incorporate ASCII values ranging from 0-255. The paper also used multiple array structures for storing information about the spaces and information about repetition of characters.

[5] Proposed a variation where the traditional Playfair algorithm is enhanced by extending the original 5x5 matrix to 6x6 matrix which included alphabets as well as number characters.

Table 3 [5] Modified Playfair Matrix 6x6

M	O	N	A	R	C
H	Y	B	D	E	F
G	I	J	K	L	P
Q	S	T	U	V	W
X	Z	0	1	2	3
4	5	6	7	8	9

[6] In this variation the researchers have used a 8x8 matrix instead of the traditional 5x5 matrix for creating the cipher text. Then the cipher text is converted to their ASCII codes in decimal and then to corresponding binary values of 7 bits. Finally a Linear Shift Register is applied to get the final cipher text.

Table 4 [6] Modified Playfair Matrix 8x8

S	**H**	**I**	**V**	**@**	**A**	**K**	**T**
B	**C**	**D**	**E**	**F**	**G**	**J**	**L**
M	**N**	**O**	**P**	**Q**	**E**	**U**	**W**
X	**Y**	**Z**	**0**	**1**	**2**	**3**	**4**
5	**6**	**7**	**8**	**9**	**!**	#	**$**
%	^	**&**	*****	**(**	**)**	_	+
=	**{**	**}**	**[**	**]**	\	|	**;**
:	**‘**	**,**	<	>	/	**.**	**?**

[7] In this variation the frequency of each alphabet in the text to be encrypted is calculated. The 2 letters with the least frequency are combined instead of combining I and J. The 5 x 5 matrix is formed by inserting the keyword without duplication of letters, the combined letters and lastly the other letters.

[8] Proposed a variation of the Playfair algorithm where random numbers are mapped to secret key of Playfair cipher method and corresponding numbers will be transmitted to the recipient instead of alphabetical letter. This method rapidly increases security of the transmission over an unsecured channel.

3. PROPOSED METHOD

3.1 Modified Playfair Algorithm Using Dynamic Matrix

In this method a dynamic multi-dimensional array is used. The array has a standard size of 15x15 which can use **printable** 225 extended ASCII values. The password or the Key is inserted first from left to right and top to bottom. Then rest of the table is filled with remaining characters according to their ASCII values from left to right and top to bottom. All repeatable letters in the key are deleted before placing into the table. And all repeatable letters as well as odd number of letters in the message are padded with Ã .Then the encryption and decryption process is taken from traditional Playfair method.

3.2 Algorithm

3.2.1 *Fill The Matrix*

This algorithm fills the matrix with the password and the rest with the other ASCII values. For the rest of the document, it will be referred as **FILL-MATRIX.**

3.2.1.1 To make this algorithm work, two parameters need to be passed:
3.2.1.1.1 The value integer (VAL).
3.2.1.1.2 The password string(PASSWORD)
3.2.1.2 Create an array FILL and fill it with values from 0 to POW (2, VAL).
3.2.1.3 Delete the repeated characters from PASSWORD.
3.2.1.4 Delete the characters from the array FILL that are already present in the PASSWORD.
3.2.1.5 The value of VAL is calculated as POW (2, VAL/2).
3.2.1.6 Create a 2-d array ARRAY with dimensions VAL X VAL.
3.2.1.6.1 Fill ARRAY with the PASSWORD in the beginning.
3.2.1.6.2 Fill the rest of the ARRAY from the array FILL.

3.2.1.7 Return the resulting ARRAY.

3.2.2 *Find The Two Characters From The Matrix*

This algorithm finds the two characters from the matrix sequentially by matching the ASCII values of the characters. For the rest of the document, it will be referred as **FIND-TWO-CHAR.**

3.2.2.1 To make this algorithm work, four parameters need to be passed:
3.2.2.1.1 The value integer (VAL).
3.2.2.1.2 The first character (CHAR1)
3.2.2.1.3 The second character (CHAR2)
3.2.2.1.4 The array (ARRAY)

3.2.2.2 Loops from i=0 to VAL

3.2.2.2.1 Loops from j=0 to VAL

3.2.2.2.1.1 If ARRAY[i][j] == CHAR1, store RETURN[‘x1’] = i and RETURN[‘y1’] = j and flag1 = 1.

3.2.2.2.1.2 If ARRAY[i][j] == CHAR2, store RETURN[‘x2’] = i and RETURN[‘y2’] = j and flag2 = 1.

3.2.2.2.1.3 If flag1 == flag2 == 1, return RETURN.

3.2.3 *Encryption Engine*

The workflow of encryption engine is as follows:

3.2.3.1 *Getting The Input*

3.2.3.1.1 Get the source text from the User, which is referred as ***TOENCRYPT***.

3.2.3.1.2 Get the password from the user, which is referred as ***PASS***.

3.2.3.2 *Checks The Maximum Value Of The Element From TOENCRYPT And PASS*

3.2.3.2.1 Calls MAX-ELEMENT with TOENCRYPT as well as PASS.

3.2.3.2.2 The max of both is kept in MAX.

3.2.3.3 *Calculating The Dimensions Of Matrix To Make It Dynamic*

3.2.3.3.1 The value of MAX is checked such that its value is just less than Power (2, i).

3.2.3.3.2 That value of i is returned.

3.2.3.4 *Fill The Matrix*

3.2.3.4.1 Calculate the RETURN matrix by calling FILL-MATRIX with parameters i and PASS.

3.2.3.4.2 ‘Success’ is stored in RETURN [‘type’].

3.2.3.4.3 i is stored in RETURN[‘dim’].

3.2.3.4.4 RETURN is returned.

3.2.3.5 *Encrypting The TOENCRYPT*

3.2.3.5.1 To make this algorithm work, three parameters need to be passed:

3.2.3.5.1.1 The encryption string (TOENCRYPT).

3.2.3.5.1.2 The array (ARRAY)

3.2.3.5.1.3 The value integer (VAL)

3.2.3.5.2 Checks if the length of the TOENCRYPT is odd. If yes adds '×' to make it even.

3.2.3.5.3 Splits TOENCRYPT into an array TEMP containing groups of 2 characters.

3.2.3.5.4 Loop through each element (STR) of TEMP:

3.2.3.5.4.1 FIND-TWO-CHAR is called with parameters VAL, first character of STR, second character of STR and ARRAY and store it in TEMP3.

3.2.3.5.4.2 Checks if the characters are on:

3.2.3.5.4.2.1 Same row, replace it with the next element on that row.

3.2.3.5.4.2.2 Same column, replace it with the next element on that column.

3.2.3.5.4.2.3 Different column and row, replace it with the other corner elements formed by the rectangle.

3.2.3.5.4.3 Concatenate the characters with TEMPENCRYPT.

3.2.3.5.5 Return TEMPENCRYPT.

3.2.4 *Decryption Engine*

The workflow of decryption engine is as follows:

3.2.4.1 *Getting the input*

Two inputs from the user is taken:

3.2.4.1.1 Get the encrypted text from the User, which is referred as ***TODECRYPT***.

3.2.4.1.2 Get the password from the user, which is referred as ***PASS***.

3.2.4.2 *Checks The Maximum Value Of The Element From TODECRYPT And PASS*

3.2.4.2.1 Calls MAX-ELEMENT with TODECRYPT as well as PASS.

3.2.4.2.2 The max of both is kept in MAX.

3.2.4.3 *Calculating The Dimensions Of Matrix To Make It Dynamic*

3.2.4.3.1 The value of MAX is checked such that its value is just less than Power (2, i).

3.2.4.3.2 That value of i is returned.

3.2.4.4 *Fill The Matrix*

3.2.4.4.1 Calculate the RETURN matrix by calling FILL-MATRIX with parameters i and PASS.

3.2.4.4.2 ‘Success’ is stored in RETURN [‘type’].

3.2.4.4.3 i is stored in RETURN[‘dim’].

3.2.4.4.4 RETURN is returned.

3.2.4.5 *Decrypting The TODECRYPT:*

3.2.4.5.1 To make this algorithm work, three parameters need to be passed:

3.2.4.5.1.1 The encrypted string (TODECRYPT).

3.2.4.5.1.2 The array (ARRAY)

3.2.4.5.1.3 The value integer (VAL)

3.2.4.5.2 Splits TODECRYPT into an array TEMP containing groups of 2 characters.

3.2.4.5.3 Loop through each element (STR) of TEMP:

3.2.4.5.3.1 FIND-TWO-CHAR is called with parameters VAL, first character of STR, second character of STR and ARRAY and store it in TEMP3.

3.2.4.5.3.2 Checks if the characters are on:

3.2.4.5.3.2.1 Same row, replace it with the previous element on that row.

3.2.4.5.3.2.2 Same column, replace it with the previous element on that column.

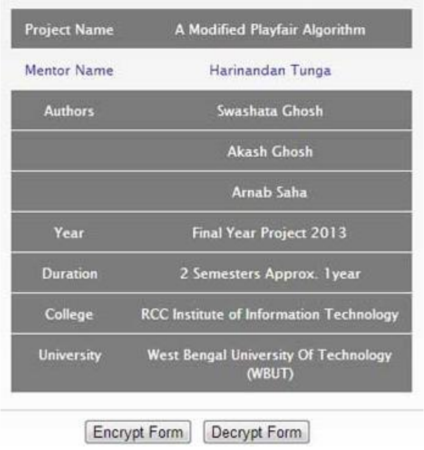
3.2.4.5.3.2.3 Different column and row, replace it with the other corner elements formed by the rectangle.

3.2.4.5.3.3 Concatenate the characters with TEMPDECRYPT.

3.2.4.5.4 Return TEMPDECRYPT.

4. RESULTS AND DISCUSSION

In this paper the simulation of the algorithm is done by making an application using PHP.



Project Name	A Modified Playfair Algorithm
Mentor Name	Harinandan Tunga
Authors	Swashata Ghosh
	Akash Ghosh
	Arnab Saha
Year	Final Year Project 2013
Duration	2 Semesters Approx. 1year
College	RCC Institute of Information Technology
University	West Bengal University Of Technology (WBUT)

Figure 1 : Interface

This page contains the basic project information. The user can select from the two options:

Encrypt Form
Decrypt Form

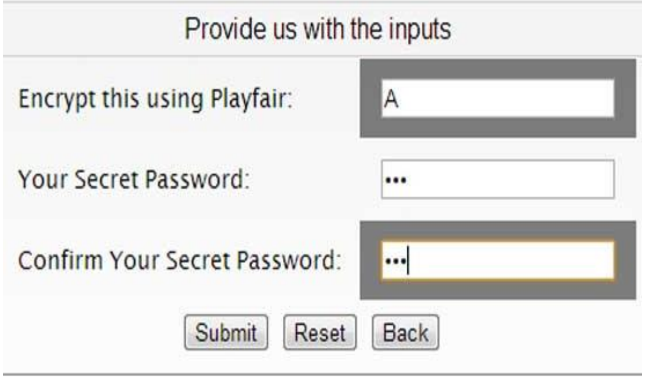
According to the option selected the user will get access to the encryption or the decryption form.

Threemodels are used for testing the product:

4.1 Case 1:

A simplealphabet is taken as an example;let's say **A** with password as **RED.**

Message**: A**
Password**: RED**



Provide us with the inputs

Encrypt this using Playfair: A

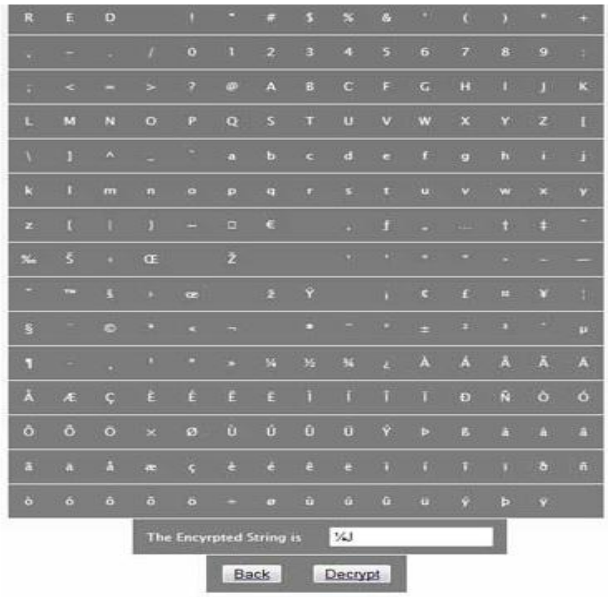
Your Secret Password: •••

Confirm Your Secret Password: •••

Figure 2: Encryption Input Interface

The encryption form button is selected. Then the appropriate inputs for the message and password key are entered.

The cipher text is generated and the process used to generate this cipher text is done by using modified Playfair method from the table shown below.

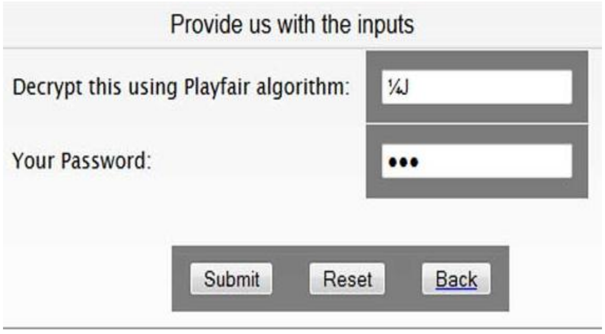


R	E	D		!	"	#	$	%	&	'	(	)	*	+
,	-	.	/	0	1	2	3	4	5	6	7	8	9	:
;	<	=	>	?	@	A	B	C	F	G	H	I	J	K
L	M	N	O	P	Q	S	T	U	V	W	X	Y	Z	[
\	]	^	_	`	a	b	c	d	e	f	g	h	i	j
k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
z	{	|	}	~	□	€		‚	ƒ	„	…	†	‡	ˆ
‰	Š	‹	Œ		Ž			'	'	"	"	•	–	—
˜	™	š	›	œ		ž	Ÿ		¡	¢	£	¤	¥	¦
§	¨	©	ª	«	¬		®	¯	°	±	²	³	´	µ
¶	·	¸	¹	º	»	¼	½	¾	¿	À	Á	Â	Ã	Ä
Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó
Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á	â
ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	ð	ñ
ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	

The Encyrpted String is ¼J

Figure 3: Output of the Encryption Engine

Next the decrypt form is selected and entering appropriate inputs for the password and the cipher text is done as follows:



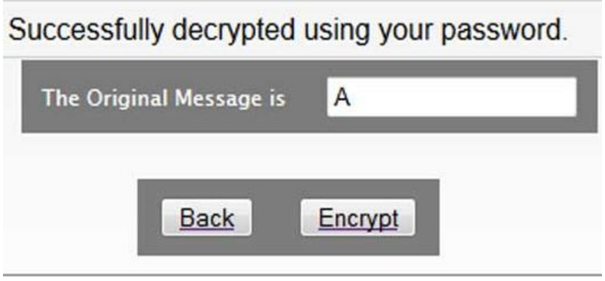
Provide us with the inputs

Decrypt this using Playfair algorithm: ¼J

Your Password: •••

Figure 4: Decryption Interface

The decryption machine outputs us with the original message.



Successfully decrypted using your password.

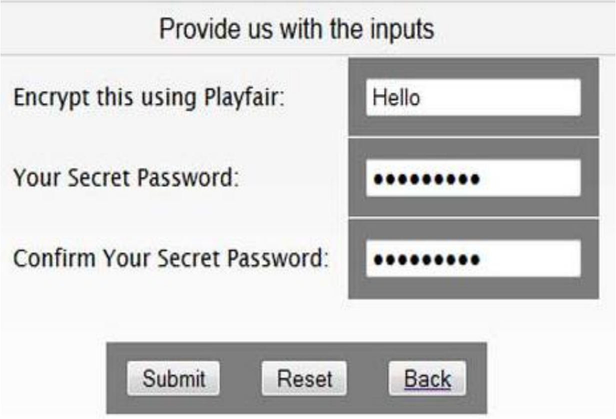
The Original Message is A

Figure 5: Output of Decryption Engine

4.2 Case 2:

Here a word is taken with repeated letters as an example for demonstrating the project.

Message**: Hello**
Password**: Duplicate**



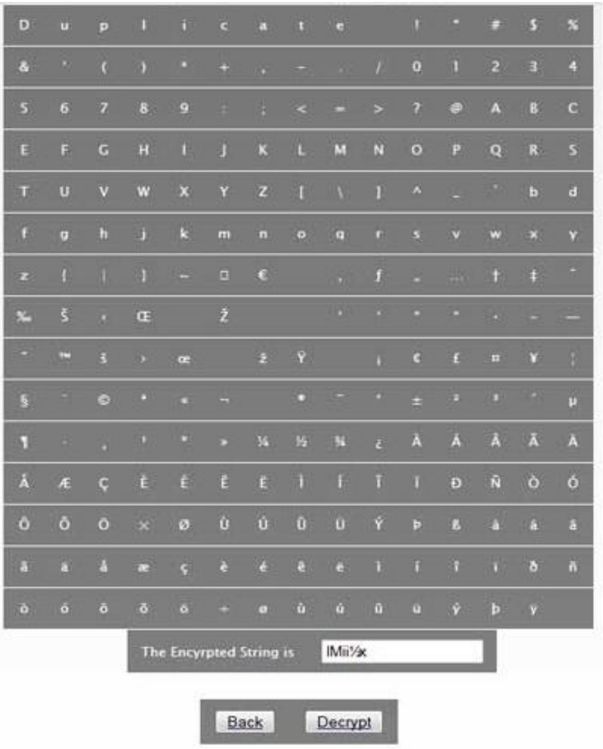
Provide us with the inputs

Encrypt this using Playfair: Hello

Your Secret Password: •••••••••

Confirm Your Secret Password: •••••••••

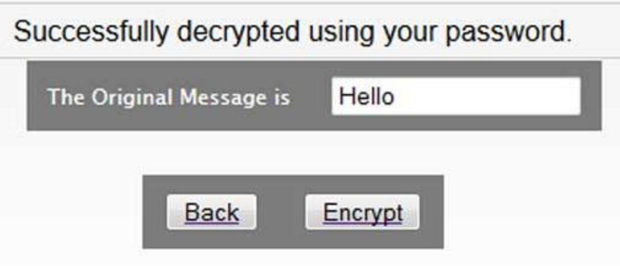
Figure 6: Encryption Input Interface



D	u	p	l	i	c	a	t	e		!	"	#	$	%
&	'	(	)	*	+	,	–	.	/	0	1	2	3	4
5	6	7	8	9	:	;	<	=	>	?	@	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	[	\	]	^	_	`	b	d
f	g	h	j	k	m	n	o	q	r	s	v	w	x	y
z	{	|	}	~	□	€		‚	ƒ	„	…	†	‡	ˆ
‰	Š	‹	Œ		Ž			'	'	"	"	•	–	—
˜	™	š	›	œ		ž	Ÿ		¡	¢	£	¤	¥	¦
§	¨	©	ª	«	¬		®	¯	°	±	²	³	´	µ
¶	·	¸	¹	º	»	¼	½	¾	¿	À	Á	Â	Ã	Ä
Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó
Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á	â
ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	ð	ñ
ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	

The Encyrpted String is IMii½x

Figure 7: Output of the Encryption Engine



Successfully decrypted using your password.

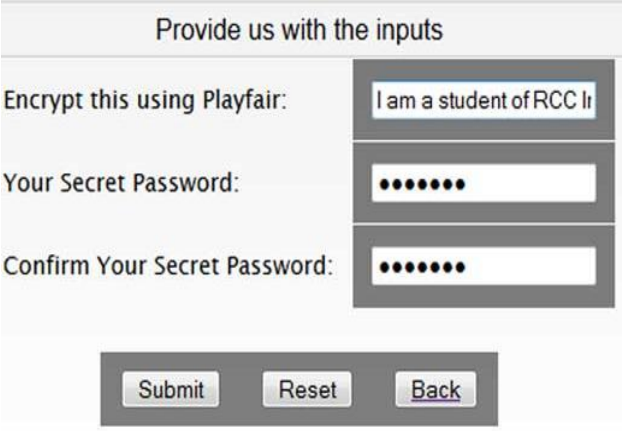
The Original Message is Hello

Figure 8: Output of Decryption Engine

4.3 Case 3:

Here a sentence is taken as an input.

Message**: I am a student of RCC Institute of Information Technology.**
Password**: student**



Provide us with the inputs

Encrypt this using Playfair: I am a student of RCC Ir

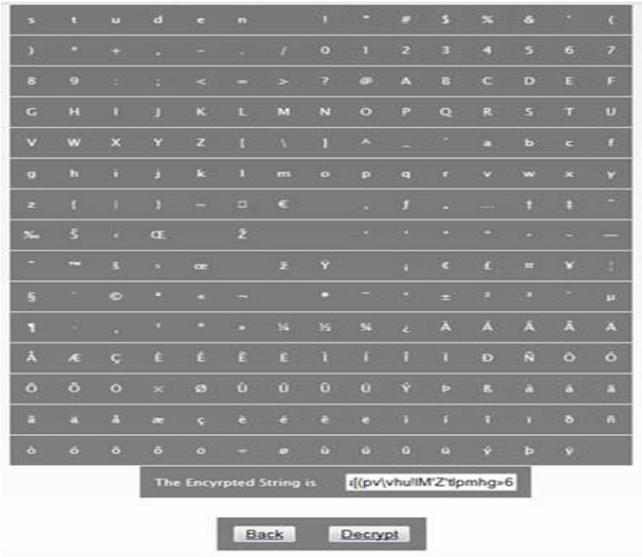
Your Secret Password: •••••••

Confirm Your Secret Password: •••••••

Figure 9: Encryption Input Interface

The Cipher text is

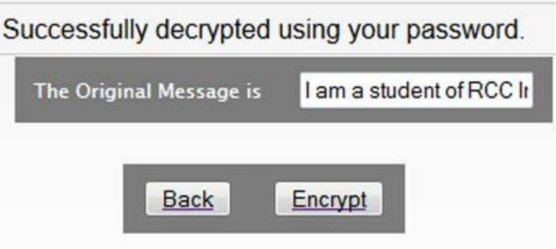
uMv\\%!tuden um!(\aR%>uLtuuhdun!]yMu[(pv\vhu!lM'Z'tlpmhg»6



s	t	u	d	e	n		!	"	#	$	%	&	'	(
)	*	+	,	-	.	/	0	1	2	3	4	5	6	7
8	9	:	;	<	=	>	?	@	A	B	C	D	E	F
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
V	W	X	Y	Z	[	\	]	^	_	`	a	b	c	f
g	h	i	j	k	l	m	o	p	q	r	v	w	x	y
z	{	|	}	~	□	€		‚	ƒ	„	…	†	‡	ˆ
‰	Š	‹	Œ		Ž			'	'	"	"	•	–	—
˜	™	š	›	œ		ž	Ÿ		¡	¢	£	¤	¥	¦
§	¨	©	ª	«	¬		®	¯	°	±	²	³	´	µ
¶	·	¸	¹	º	»	¼	½	¾	¿	À	Á	Â	Ã	Ä
Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó
Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á	â
ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	ð	ñ
ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	

The Encyrpted String is ![(pv\vhu!lM'Z'tlpmhg»6

Figure 10 : Output of the Encryption Engine



Successfully decrypted using your password.

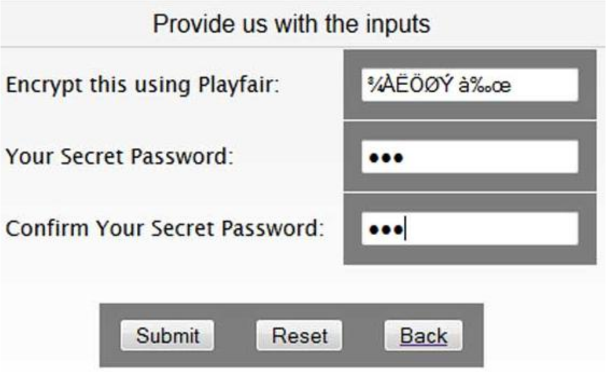
The Original Message is I am a student of RCC Ir

Figure 11: Output of Decryption Engine

4.4 Case 4:

Here an extended ASCII symbols is taken as a input.

Message**: ¾ÀËÖØÝ à‰œ**
Password**:Ã½Å**



Provide us with the inputs

Encrypt this using Playfair: ¾ÀËÖØÝ à‰œ

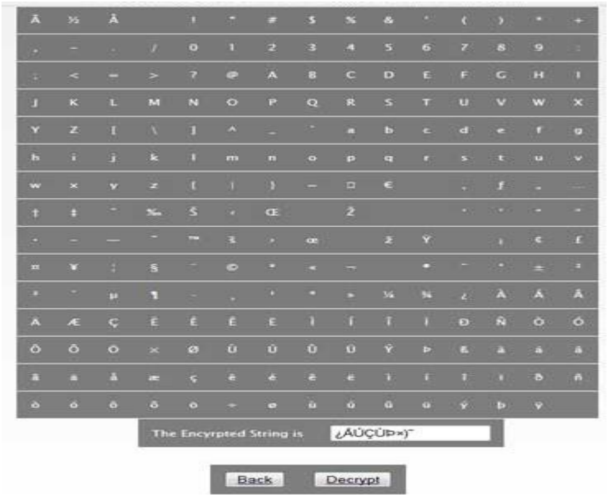
Your Secret Password: •••

Confirm Your Secret Password: •••

Figure 12: Encryption Input Interface

The Cipher text is:

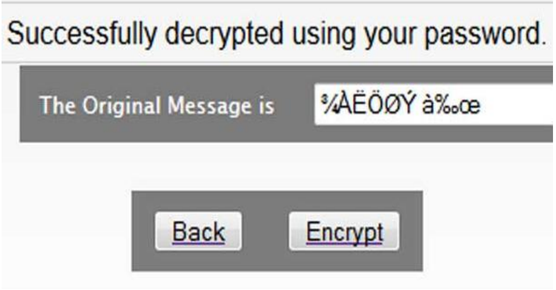
¿ÁÚÇÙÞ×)˜•



Ã	½	Å		!	"	#	$	%	&	'	(	)	*	+
,	–	.	/	0	1	2	3	4	5	6	7	8	9	:
;	<	=	>	?	@	A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Y	Z	[	\	]	^	_	`	a	b	c	d	e	f	g
h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
w	x	y	z	{	|	}	~	□	€		‚	ƒ	„	…
†	‡	ˆ	‰	Š	‹	Œ		Ž			‘	’	“	”
•	–	—	˜	™	š	›	œ		ž	Ÿ		¡	¢	£
¤	¥	¦	§	¨	©	ª	«	¬		®	¯	°	±	²
³	´	µ	¶	·	¸	¹	º	»	¼	¾	¿	À	Á	Â
Ä	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó
Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á	â
ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	ð	ñ
ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	

The Encyrpted String is ¿ÁÜÇÜÞ×)¯

Figure 13: Output of the Encryption Engine



Successfully decrypted using your password.

The Original Message is ¾ÀËÖØÝ à‰œ

Figure 14: Output of Decryption Engine

5. CONCLUSION

Any printable ASCII characters can be encrypted and decrypted. This algorithm does not support audio encryption or image encryption. This is one of the fastest ways of encryption and decryption method. The cipher cannot be easily broken down by any cryptanalytic attack although brute force makes it vulnerable. The purpose of this paper was to enhance traditional Playfair cipher. The Playfair cipher uses a 15x15 matrix which includes all the printable extended ASCII values and therefore the process makes it harder to break and also is the fastest way of encryption.

6. REFERENCES

[1] 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)

[2] 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

[3] William Stallings, Cryptography and Network Security Principles and Practices, 4th Edition, Pearson Education.

[4] Atul Kahate, Cryptography and Network Security, 2nd Ed., Tata McGraw-Hill Publishing Company Limited, New Delhi.

[5] Ravindra Babu K, S. Uday Kumar, A. Vinay Babu, I.V.N.S Aditya , P. Komuraiah “An Extension to Traditional Playfair Cryptographic Method”, International Journal of Computer Applications (0975 – 8887) Volume 17– No.5, March 2011.

[6] 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

[7] 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

[8] Packirisamy Murali and Gandhidoss Senthilkumar “Modified Version of Playfair Cipher using Linear Feedback Shift Register”, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.12, December 2008

[9] Behrouz A. Forouzan, Cryptography and Network Security. Special Indian Edition 2007, Tata McGraw-Hill Publishing Company Limited, New Delhi.

[10] Sanjay Basu and Utpal Kumar Ray “Modified Playfair Cipher using Rectangular Matrix”, IJCA International Journal of Computer Applications, Volume 46- No.9, May 2012