Secure Key Exchange using RSA in Extended Playfair Cipher Technique

Surendra Singh Chauhan
M. Tech Scholar, (CSE)
SMEC, Neemrana, Rajasthan

Hawa Singh
Assistant Professor (CSE)
SMEC, Neemrana, Rajasthan

Ram Niwas Gurjar
Assistant Professor, (CSE)
SMEC, Neemrana, Rajasthan

ABSTRACT
In today's digital world cryptography is used to secure information in order to provide the privacy for the intended sender and receiver by managing the message with the public key. The objective of this work is to securing the key of Playfair cipher using RSA algorithm. It is a two stage application, in first stage the existing methods of Playfair cipher modified by increasing in the size of matrix, so that the restrictions of earlier works of PF cipher using 5×5 matrix were overcome in the proposed work. The proposed method use a 12×8 matrix which contain all alphabetic, numeric and special character use in keyboard as input. This work is an enhancement to the existing algorithms that uses 5×5 matrix to pick cipher characters. It makes use of alphabets both lower and upper case characters, number and special characters for constructing the contents of the matrix. In the second stage, RSA public key encryption technique is used for sending the key of the PF ciphers securely. Finally, the security strength of the whole system has been analyzed and tried to fulfil the requirement of security.

Keywords
Playfair=PF, Plaintext=PT, Plaintext1=PT1, Plaintext2=PT2, Cipher text= CT, Cipher text1= CT1, Cipher text2= CT2, Encryption Process =ET, Decryption process=DT

1. INTRODUCTION
1.1 Introduction about PlayfairTechnique
The basic Playfair cipher uses a matrix of 5×5 containing a key or phrase. Memorization of the key is achieved by generating a 5×5 key table and cipher text is created by applying four simple rules on this key table [8]. To generate the key table, one would first fill in the spaces in the table with the letters of the key (dropping any duplicate letters), then fill the remaining spaces with the rest of the letters of the alphabet in order (usually omitting "Q" to reduce the alphabet to fit; other versions put both "I" and "J" in the same space) [8]. The key can be written in the top rows of the table, from left to right, or in some other pattern, such as a spiral beginning in the upper-left-hand corner and ending in the center. The key together with the conventions for filling in the 5×5 table constitute the cipher key [8].

To encrypt a message, one would break the message into digraphs (groups of 2 letters) for example; "HELLO WORLD" becomes "HE, LL, OW, OR, LD", and maps them out on the key table. If needed, append a "Z" to complete the final digraph. The two letters of the digraph are considered as the opposite corners of a rectangle in the key table [2]. Note the relative position of the corners of this rectangle. Then apply the following 4 rules, in order, to get encrypted message for each pair of letters in the PT [8].

1.1.1 Rules for making CT using PF Matrix
1. Add an "X" after the first letter, if both letters are the same (or only one letter is left). Encrypt the new pair and continue doing this. Some variants of PF use "Q" instead of "X", but any uncommon monograph will do.
2. If both alphabets appear on the same row in table, replace them with the letters to their immediate right respectively (wrapping around to the left side of the row if a letter in the original pair was on the right side of the row).
3. If both alphabets appear on the same column in table, replace them with the letters immediately below respectively (wrapping around to the top side of the column if a letter in the original pair was on the bottom side of the column).
4. If both alphabets are not on the same row or column, replace them with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. The order is important – the first letter of the encrypted pair is the one that lies on the same *row* as the first letter of the PT pair.

The decryption process, use the INVERSE (opposite) of the last 3 (2,3,4) rules, and the 1st as-is (dropping any extra "X"s (or "Q"s) that do not make sense in the final message when finished) [2].

1.1.2 Limitations of 5×5 Matrix
1. The 5×5 Matrix considers the letters 'I' and 'J' as one character.
2. Only 26 letters alone can take as key without duplicates.
3. The Space between two words in the PT is not considered as one character.
4. The special characters cannot use as and numbers.
5. The uppercase alphabets are only used in 5x5 Matrix.
6. An extra letter 'X' is added when the PT word consists of odd number of characters. In the DP this 'X' is ignored. 'X' is a valid character and creates confusion because it could be a part of PT, so we cannot simply remove X in DP.
7. X is used a filler letter while repeating letter falls in the same pair are separated.

1.2 Introduction about RSA algorithm
The RSA is a public key cryptographic algorithm that is used to help ensure data communication security. It is simply based on two main cryptographic processes [3]. First, using a public key it converts an input data called the PT into an unrecognizable encrypted output called CT (EP), such that it is impossible to recover the original PT without the encryption password in a reasonable amount of time. Second,

using a private key, the RSA then converts the unrecognizable data back to its original form (DP) [3]. Today it is used in web browsers, email programs, mobile phones, virtual private networks and secure shells [3]. Until recently, the use of RSA was very much restricted by patent and export laws. However, the patent has now expired and US export laws have been relaxed [3]. The challenge of RSA is to develop an algorithm in which it is impossible to determine the private key. This algorithm is based on one-way function. As the name implies, the function is only one-way i.e. given some input values it is relatively easy to compute the result. However, it is extremely difficult, nearly impossible to determine the input values given the result .In the mathematical terms, given *x*, computing *f (x)* is relatively easy, but given *f (x)*, computing *x* is extremely difficult. The one-way function used by RSA is the multiplication of two very large prime numbers. It is relatively easy to multiply them but extremely difficult, rather impossible and time consuming to factorize them [3]

.

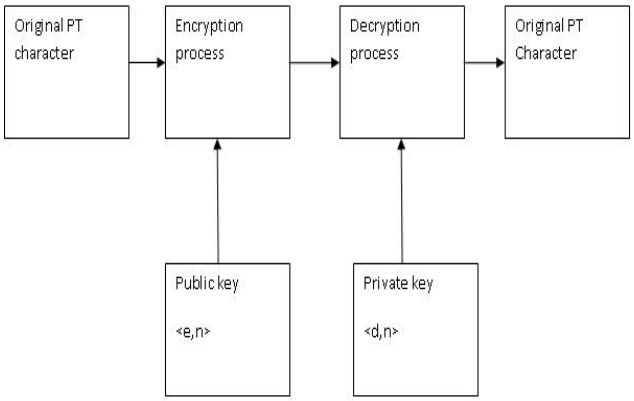

Fig. 1 RSA Model

1.3 RSA Algorithm [7]

1. Choose two large prime numbers P and Q Calculate N=P*Q.
2. Select the public key e (encryption key) such that it is not a factor of (P-1) and (Q-1).
3. Select the private key d (decryption key) such that the following equation is true:

   (d*e) mod (P-1)*(Q-1)=1
4. For encryption , calculate the cipher text

   $CT = PT^{e} \bmod N$
5. Send CT as the cipher text to receiver.
6. For decryption, calculate the plain text PT from the cipher text CT as follows:

   $PT = CT^{d} \bmod N$.

2. PROPOSED WORK

The proposed work consists of the following two steps:

1. In first step, construct a modified table of PF cipher technique, which contain all the alphabets (including lower and upper case), all numeric values (from 0 to 9), and all the special characters which are on the keyboard. The PF encryption technique is divided into two phases:
   a) First phase is creation and population of Matrix (by using the key).
   b) The second phase is encryption process.
2. In the second step, use the key as a PT in RSA algorithm to make the PT of the key and send to the receiver. At the receiver end decrypt the PT into PT (key). By this key make the PF Matrix and decrypt the message.

3. METHODOLOGY

The methodology is divided into two phases:

i. In first phase for Matrix construction use all the rules of 5×5 Matrix with these changes:
ii. The letters I and J are considered as two different letters.
iii. It allows more than 26 characters as key.
iv. It is case sensitive; it uses the upper case and lower case characters.

TABLE 1 List of upper case letters

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

TABLE 2 List of lower case letter

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

v. It considers the space between two words in PT as one character.
vi. The user can easily encrypt and decrypt the combinations of numbers efficiently.

TABLE 3 List of numeric values

0	1	2	3	4
5	6	7	8	9

vii. The user can easily encrypt and decrypt the combinations of operator efficiently.

TABLE 4 List of operators

^	*	/	%	+	-
<	=	>	!	|	&

viii. The user can easily encrypt and decrypt the ombination of brackets efficiently.

TABLE 5 List of brackets

(	)	{
}	[	]

ix. The user can easily encrypt and decrypt the combination of special character efficiently.
x. To compare with the previous algorithms, here the key length is very large, so it is very difficult to find the PT from CT without knowing a key.

TABLE 6 List of Special characters

Space	Null	"	#	$	'	,	`
:	;	@	_	.	?	~	\

xi. This algorithm adds the Null character to omplete the pair, because the "Null" character cannot affect the PT at the end of the word or sentence.

xii. This algorithm cannot separate a repeating PT letters with a filter letter.

4. ALGORITHM

Step 1. In the first phase:

1. Add the Null in the last of the key and in PT if it has odd number of character.
   i. Use a Matrix size of 12×8.
   ii. Fill up the key in the Matrix with out duplicate from left to right and from top to bottom.
   iii. Now fill the remaining blocks according the priority (Give the priority to all the tables).
      a) First priority is gives to the upper case letter table.
      b) Second priority is gives to lower case letter table.
      c) Third priority is gives to numeric value table.
      d) Forth priority is gives to operator table.
      e) Fifth priority is gives to brackets table.
      f) Sixth priority is gives to special character table.
2. If both alphabets appear on the same row of your table, replace them with the letters to their immediate right respectively (wrapping around to the left side of the row if a letter in the original pair was on the right side of the row) [1].
3. If both alphabets appear on the same column of your table, replace them with the letters immediately below respectively (wrapping around to the top side of the column if a letter in the original pair was on the bottom side of the column) [1].
4. If both alphabets are not on the same row or column, replace them with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. The order is important – the first letter of the encrypted pair is the one that lies on the same row as the first letter of the PT pair [1].

Step 2. In the second phase:

1. Create the array of key.
2. Take the ASCII code of every character in the array (array of key k[ ]).
3. Apply the RSA algorithm on the array of key (on every character one by one).
4. Store this result in the array of CT2 (CT2 [ ]).
5. Send this CT1 and CT2 at the receiver end.
6. Apply deception process on CT2 to get the original key by RSA algorithm.
7. Construct the Matrix at the receiver end.
8. Decrypt the CT1 to get the original PT by the extended PF Matrix.

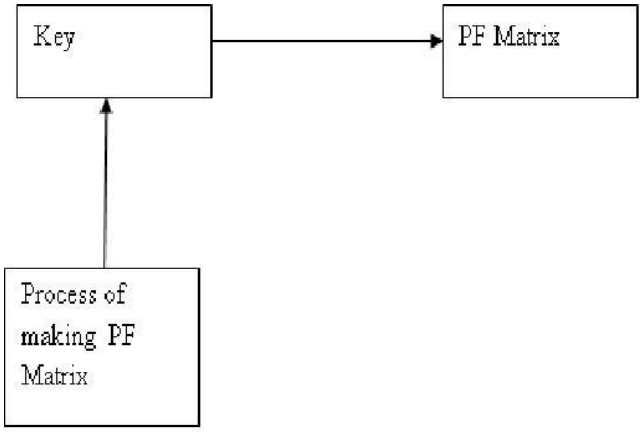

Fig. 2 Process of making of PF Matrix

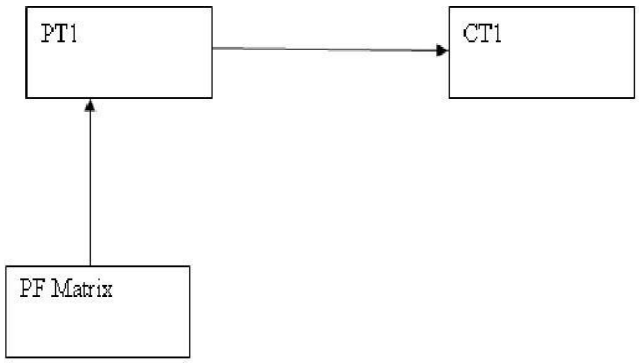

Fig. 3 Process of creating the CT1 of PT

For sending the key with the help of RSA algorithm

1. Use the key of PF Matrix as the PT2.

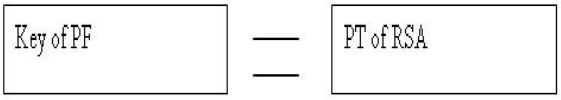

Fig. 4 Key of PF and PT of RSA are equal.

2. Give the number to every character of PT2 or use the ASCII code of them numbers.
3. The CT of PF is called CT1.
4. The CT2 calculated with the help of RSA algorithm.

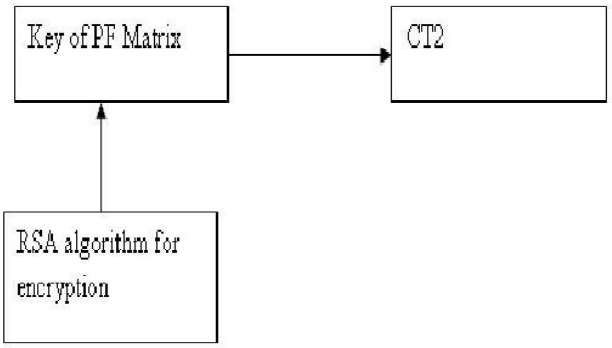

Fig. 5 Creation of CT2 With the help of RSA algorithm.

5. Send these two CT (CT1 and CT2) to the receiver.

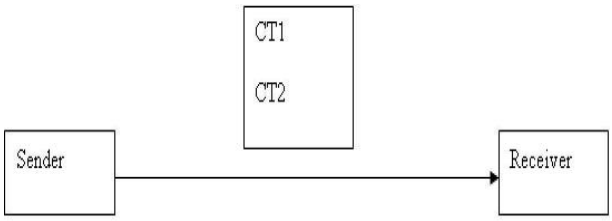

Fig. 6 Sending CT1 and CT2 at Receiver side.

6. The receiver decrypts the CT2 with the help of the RSA algorithm and calculates of the key of the PF Matrix.

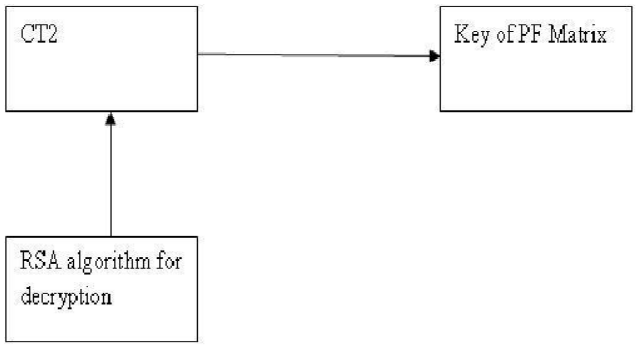

Fig. 7 Find the key of PF Matrix with the RSA lgorithm.

7. With this key construct the PF Matrix at the receiver end.
8. Now decrypt the CT1 with this PF Matrix and get the PT.

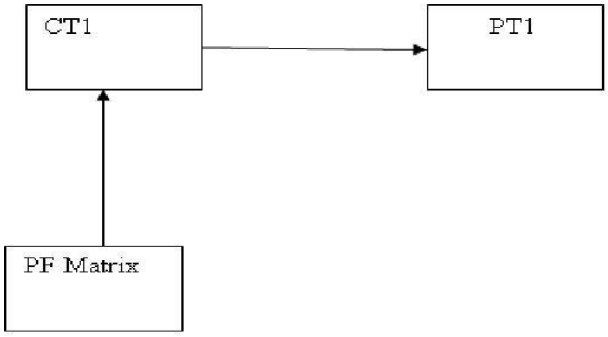

Fig. 8 Decryption of CT1

5. FLOW CHART

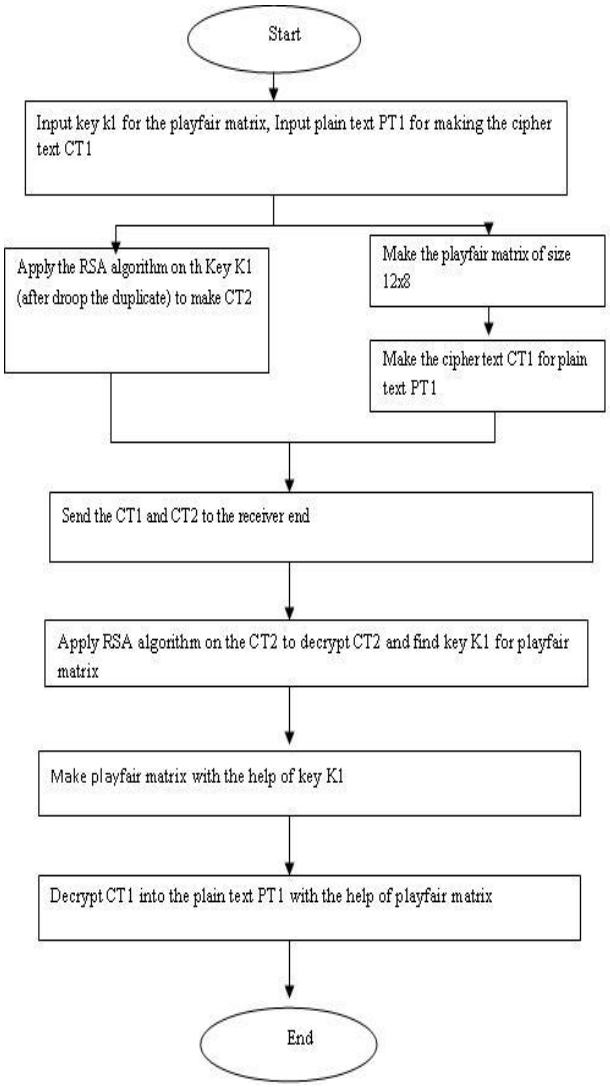

Fig. 9 Flow Chart of Algorithm

6. EXAMPLE AND SNAPSHOT

6.1 Example

Key = playfairexample

Key after dropping the duplicate = playfirexm

The key array at the sender end is:

TABLE 7 The key array at the sender end

p	l	a	y	f	i	r	e	x	m

By the help of the key array shown in TABLE 7, make the matrix of size 12×8 for play fair cipher as shown in TABLE 8, first insert the character of TABLE 7 from left to right and then from top to bottom As shown in TABLE 8. When all element of TABLE 7 are filled, then insert all the element of TABLE 1, TABLE 2, TABLE 3, TABLE 4, TABLE 5, and TABLE 6 by dropping all duplicates. By this TABLE 8 will be created at the sender end (according to proposed algorithm).

p	l	a	y	f	i	r	e
x	m	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	b	c	d	g
h	j	k	n	o	q	s	t
u 4	v 5	w 6	z 7	0 8	1 9	2 ^	3 *
/	%	+	-	<	=	>	!
|	&	(	)	{	}	[	]
Space	"	#	$	,	:	;	?
@	\	_	`	~	.	'	NULL

Plain text = i am ram

The length of the plain text is 8, so there is no need to add null in the last of the plain text.

1. Select first two characters (first character is 'i' and second character is blank space), 'i' is in the first row and sixth column blank space is in eleventh row and first column. So the corresponding cipher text for 'i' will be 'p' and cipher text for the blank space will be ':'.
2. Select the next two characters 'a' and 'm'. Character 'a' is in first row and third column and character 'm' is in second row second column. So the corresponding cipher text for 'a' will be 'l', and corresponding cipher text for 'm' will be 'A'.
3. Select next two characters (fifth character is blank space, and sixth character is 'r'), the blank space is in eleventh row and first column of table, and the r is first row and seventh column. So the corresponding cipher text for blank space will be ';', and corresponding cipher text for 'r' will be 'p'.
4. Select the next two characters 'a' and 'm', 'a' is in first row and third column, and 'm' is in second row and second column. So the corresponding ciphers text for 'a' will be 'l', and corresponding cipher text for 'm' will be 'A'.

So the cipher text at the sender end will be:

Cipher text 1 (CT1) = p:lA;plA

Now the RSA algorithm will be used on the key exchange it securely.

At the sender's end: After dropping duplicates from the key the key will become:

Key = playfirexm

The ASCII code of this key character by character is

TABLE 9 The ASCII code of this key character by character

112	108	97	121	102	105	114	101	120	109

Now select two large prime numbers 'P' and 'Q'. To make the calculation simple two small prime no. have been taken.

P= 13 , Q= 17

N=13 × 17= 221

Select any integer value of encryption key 'e' let it be 7.

Then apply encryption process on each (cipher text) values provided in TABLE 9. It will give cipher text 2(CT2)

$CT2 = (key)^e \bmod N$

$CT2 = 112^7 \bmod 221 = 5$

$CT2 = 108^7 \bmod 221 = 82$

$CT2 = 97^7 \bmod 221 = 7$

$CT2 = 121^7 \bmod 221 = 43$

$CT2 = 102^7 \bmod 221 = 119$

$CT2 = 105^7 \bmod 221 = 79$

$CT2 = 114^7 \bmod 221 = 75$

$CT2 = 101^7 \bmod 221 = 101$

$CT2 = 120^7 \bmod 221 = 120$

$CT2 = 109^7 \bmod 221 = 216$

So the CT2 at the sender end will be in the ASCII code as shown in the TABLE 10.

TABLE 10 The ASCII code of CT2 character by character at sender end

5	82	7	43	119	79	75	101	120	216

Sender will send the CT1 and CT2 to the receiver.

At the receiver end, the value of decryption key'd' will be 247

So the CT2 received at the receiver end will be as shown in TABLE 10.

Apply the decryption process on cipher text values of key

$Key = (CT2)^d \bmod N$

$Key = 5^{247} \bmod 221 = 112$

$Key = 82^{247} \bmod 221 = 108$

$Key = 7^{247} \bmod 221 = 97$

$Key = 43^{247} \bmod 221 = 121$

$Key = 119^{247} \bmod 221 = 102$

$Key = 79^{247} \bmod 221 = 105$

$Key = 75^{247} \bmod 221 = 114$

$Key = 101^{247} \bmod 221 = 101$

Key = 120^{247} mod 221= 120

Key = 216^{247} mod 221= 109

So the total key at the receiver end (ASCII Code) will be

TABLE 11 The ASCII code of this key character by character

112	108	97	121	102	105	114	101	120	109

Now convert these ASCII values into the character stream that will be the key for the play fair matrix.

Key=playfirexm

The key array at the receiver end will be:

TABLE 12 The key array at the receiver end

p	l	a	Y	F	i	r	e	x	m

By the help of this key array we make the play fair matrix of size 12×8 size.

TABLE 13 table of 12 × 8 playfair matrix at receiver end

p	l	a	y	f	i	r	e
x	m	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	b	c	d	g
h	j	k	n	o	q	s	t
u	v	w	z	0	1	2	3
4	5	6	7	8	9	^	*
/	%	+	-	<	=	>	!
|	&	(	)	{	}	[	]
Space	"	#	$	,	:	;	?
@	\	_	`	~	.	'	NULL

Cipher text 1 (CT1) = p:lA;plA

There the length of the cipher text is 8 (even), so there is no need to add null in the last of the plain text.

1. Select first two characters ‘p’ and ‘:’, ‘p’ is in the first row and first column, and ‘:’ is in eleventh row and sixth column. So he corresponding plain text for ‘p’ is ‘i’, and corresponding plain text for ‘:’ is blank space.
2. Select the next two characters ‘l’ and ‘A’. The ‘l’ is in first row and second column, and ‘A’ is in second row third column. So the corresponding plain text for ‘l’ is ‘a’, and plain text for ‘A’ is ‘m’.
3. Select next two characters ‘;’ and ‘p’, the ‘;’ is eleventh row and seventh column, and ‘p’ is first row and first column. So the corresponding plain text for ‘;’ is blank space, and plain text for ‘p’ is ‘r’.
4. Select the next two characters ‘l’ and ‘A’. The character ‘l’ is in first row and second column, and ‘A’ is in second row third column. So the corresponding plain text for ‘l’ is ‘a’, and plain text for ‘A’ is ‘m’.

Now after decryption, the cipher text is converted into the plain text with the help of the playfair matrix and applies the proposed algorithm that is:

PT= i am ram

Snapshot of example 1

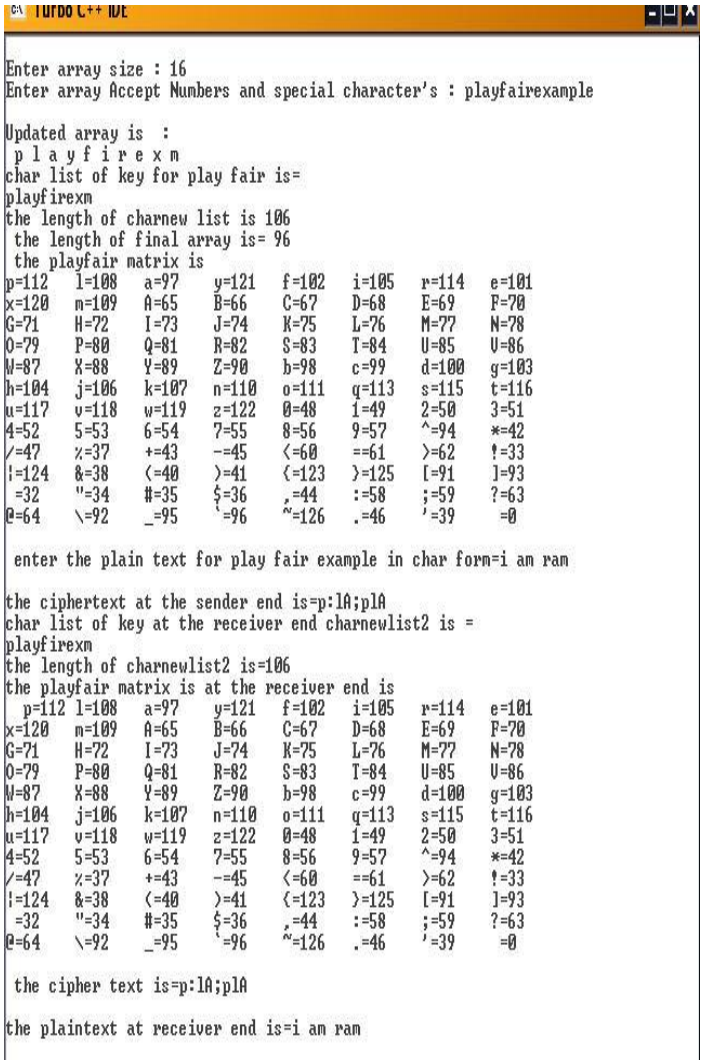

Fig. 10 Output Snapshot of Example 1 at Sender side

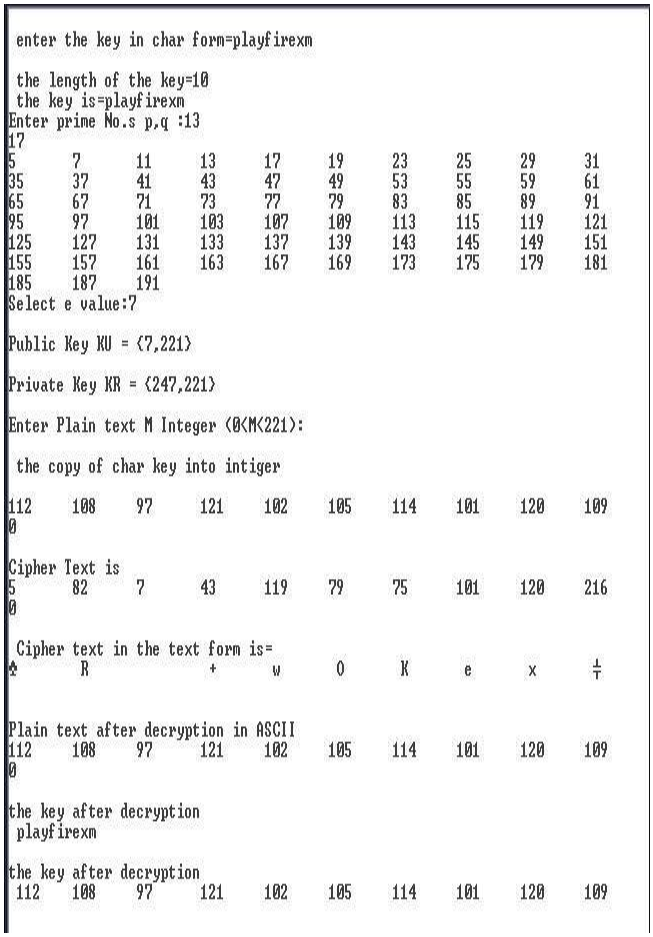

Fig. 11 Output Snapshot of Example 1 at Receiver side

7. ADVANTAGE OF ALGORITHM

1. The 'I' and 'J' character are in different cell.
2. The Space between two words in the PT is considered as one character.
3. The special characters are used in this algorithm.
4. The uppercase and lower case alphabets are in this algorithm.
5. An extra letter NULL is added when the PT word consists of odd number of character. In the DP this NULL is ignored.
6. In this algorithm 96 character are used so it takes advantage on 5x5 matrix which used the 26 characters.
7. The proposed 12×8 Playfair cipher can be said to be safe from Brute Force Attack, as the attacker has to find in a 96×96 = 9216 digraphs. Also the use of a 7-bit Linear Feedback Shift Register to generate pseudorandom numbers allows a key space of 2^7. So the altogether $96 \times 96 \times 2^7 = 1179648$ is quite a huge alternative to search for the proper key.
8. Increasing the key size also reduces the chances to break the cipher by Frequency Analysis. The probability of occurrence of an element in the original Playfair matrix of size 5×5 was 1/26 = 0.0384, whereas in the extended 12×8 Playfair matrix the probability is 1/96 = 0.010416666666666666666666666666667, which is far less when compared and it makes the frequency analysis a tougher job.

8. CONCLUSION

So far encryption technique adopting the concept of PLAY-FAIR CIPHER MATRIX of size 5X5 has been programmed for calculating the CT CT1. Finally, we have pointed the merits and demerits of traditional PF algorithm. In order to overcome the demerits, we have proposed an extension to traditional PF cipher algorithm; which can be used more efficiently even for the PT containing alphanumeric values and special characters. Then a public key encryption system has been designed which provides both confidentiality and authentication, but there are some limitations. Complete mathematical derivation is given to show the exact result at both sender and receiver sides the previous encryption technique is also a part of this system. After completion of the program the strength of the technique has been checked and this encryption technique can also be used for other networks. In this algorithm playfair matrix is used for creating the cipher text and the RSA algorithm is used for providing the secure channel.

9. REFERENCES

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

[2] S.S.Dhenakaran, M. Ilayaraja, "Extension of PF Cipher using 16X16 Matrix", International Journal of Computer Applications (0975 – 888), Volume 48– No.7, June 2012.

[3] Chandra M. Kota and Cherif Aissi1," Implementation of the RSA algorithm and its cryptanalysis".

[4] R. L. Rivest. A. Shamir and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," Commun. ACM, Vol. 21, No. 2, pp. 158-164, Feb 1978.

[5] Andrew S. Tanenbaum, Networks Computer, 5th Edition, Pearson Education, ISBN-10: 0132553171. Diffie, W., and Hellman, M. New directions in cryptography. IEEE Trans. Inform. Theory IT-22,(Nov. 1976), 644-654.

[6] Diffie, W., and Hellman, M. Exhaustive cryptanalysis of the NBS data encryption standard. Computer 10 (June 1977), 74-84.

[7] William Stallings, "Cryptography and Network Security: Principles and Practice", 4th edition, Prentice Hall, 2006.

[8] Atul Kahate, "Cryptography and Network Security", 2nd edition, McGraw-Hill, 2010.

[9] Bernard Menezes, "Network Security and Cryptography", CENGAGE Learning, 2014.