COURSEWORK REPORT

Implementation of a Cryptography Algorithm

Semester B

MSc Software Engineering

2003-2004

Revision [1.0.00]

Department of Computing

London Metropolitan University

For:

Dr. Danni Novakovic (CSP006N – Information Security for Enterprise Systems)

By:

Group Name: **CYPHERZ**

Mr. Umer Ejaz Butt (ID: 03028264, ueb001@londonmet.ac.uk)

Mr. Fahad Habib (ID: 02029881, fah086@londonmet.ac.uk)

Mr. Onyekachi Ihedioha (ID: 02011352, oci004@londonmet.ac.uk)

Mr. Shamsuddin Pabani (ID: 02024930, shp082@londonmet.ac.uk)

Confidential and Proprietary Document

DOC: Dev/SDD/1204/4001 (VER: 1.0.00)

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The need to provide security of information has become important especially with the advancement of computing technologies. By using Internet, Information on the Internet has become vulnerable, as is evident from incident reports. Attacks have cost the IT industry, organisations and individuals alike, billions in financial losses. Hackers and malicious codes have exploited vulnerabilities that exist in operating systems running on Internet. No system is a 100% safe as long as it is connected to the web.

Information is transmitted on the Internet every second (or less). Companies, government agencies, individuals, all communicate vital and sensitive information across a vulnerable medium (the Web) that is open to theft. If unduly accessed by an illegitimate user, the confidentiality and integrity of that data can be compromised.

Thus, protecting valuable information that is transferred across the Internet has become vital. If computer systems can be compromised, then the data they hold must be protected (or hidden) from hackers. This can be achieved by using encryption techniques. One of the foremost methods that provide the foundation for computer and communications security is Cryptography **[CN03]**.

Cryptography is “A branch of cryptology (study of secure communications) that deals with the design of algorithms for encryption and decryption to ensure privacy of messages.” **[CN03]**

The history of cryptography is not new. It has extended and interesting history dated back as far as to 1000 BC. So these are not just computer systems that initiate cryptography, but the need of secure communication is apparent from ancient times. Many rulers like Julius Caesar used encryption technique to send official transcripts. Many algorithms were used during the World Wars to transmit classified messages to the troops in the battlefield. Nowadays, cryptography is mostly used in protecting the national assets, government transcripts, classified information and military secrets.

With the online shopping boom, security of personal information is critical. Cryptographic encryption/decryption techniques are playing crucial role in ensuring privacy of the private data. Many algorithms like DES, AES, SHA, RSA are widely used in protecting individual information.

2. AIMS AND OBJECTIVES

The aim of the coursework is to develop a cryptographic algorithm. This can be done either by developing a new algorithm from scratch or by enhancing/modifying the existing cryptographic algorithms.

To accomplish this task, following goals were set

- To look for appropriate cryptographic algorithm that can be enhanced or modified
- To find limitations of existing algorithms and try to overcome them
- To choose suitable programming language to implement the selected algorithm
- To search for the existing variants of the selected algorithm and compare those with our implementation

3. ALGORITHM

To do this assignment, we looked at number of algorithm based upon which we can start building our algorithm. For this purpose, we look into following algorithm and there usages.

- Churchyard Algorithm
- Polybius Chequerboard Algorithm
- Playfair Algorithm
- The ADFGVX Cipher

After analyzing these algorithms, we decided to go with the Playfair Algorithm. Following section will discuss briefly about it.

3.1. Playfair Algorithm

It is a manual symmetric encryption technique invented in 1854 by Charles Wheatstone for telegraph secrecy. But it is named after his friend Lyon PlayfairIt was the first literal digraph substitution cipher. The technique encrypts pairs of letters (digraphs), instead of single letters [**PA04**].

This algorithm was used by British forces in World War 1 and also by the Australians during World War II. The Playfair is significantly harder to break since straight frequency analysis doesn't work with it. “Frequency analysis is a method for "breaking" simple substitution ciphers.” (Wikipedia [**WK04**])

It is a primitive algorithm block cipher by modern standards now. The computational power of today’s computers can break it easily by using quality software. But after its creation in 1854, it was adopted by British Government to use in its official messaging. However, the dogma used by Playfair cipher is used as baseline to many modern computer block ciphers [**PA04a**].

3.1.1. How it works?

Following steps can be observed in encrypting a message using this algorithm

1. The algorithm works in a specific way. The first step is to select a key that will be used in creating the Playfair matrix. A Playfair matrix is a 5*5 matrix consisting of all alphabets in such a way that no letter should repeat in it with I and J treated as one letter. The selected cipher key is important as it helps in creating matrix and encoding, decoding of the message. The key must be private and be known by both sender and receiver in order to encode or decode the messages.

2. After choosing a key, the message is divided in the form of digraphs in such a way that no diagraph consist of similar letter. If so, replace the repeated letter with some other letter like ‘X’.

3. Next step is to replace the digraphs with the encoded pair of letters from the matrix. Each digraph will replace with a specific pair of the matrix. Following rules are applied in replacing the original digraph [**WK04p**]:

    a. If the letters appear in the same row of the matrix, replace them with the letters to their immediate right respectively.

    b. If the letters appear on the same column of the generated matrix, replace them with the letters immediately below respectively.

    c. If the letters 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.

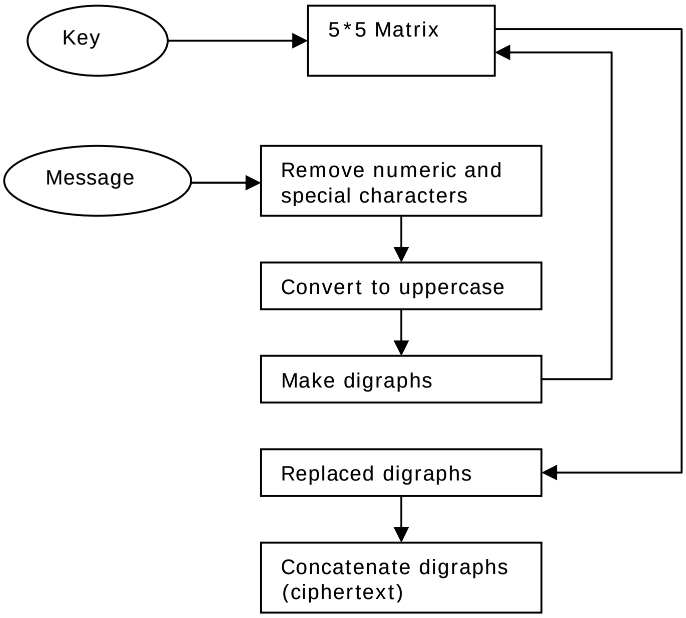

Key
5*5 Matrix
Message
Remove numeric and special characters
Convert to uppercase
Make digraphs
Replaced digraphs
Concatenate digraphs (ciphertext)

Fig 1. Playfair Algorithm Block Diagram

3.1.2. Example

Suppose, we have a message

“*We will meet at the Hyde park corner. Bring all the required documents with you.*”

And the key selected for this message is “Cricket match”.

Based on this key, following matrix will be created,

$$\begin{bmatrix} C & R & I & K & E \\ T & M & A & H & B \\ D & F & G & L & N \\ O & P & Q & S & U \\ V & W & X & Y & Z \end{bmatrix}$$

The next step is to create digraphs of the message in UPPERCASE and without any punctuation and J will become I, if any

WE WI LL ME ET AT TH EH YD
EP AR KC OR NE RB RI NG AL
LT HE RE QU IR ED DO CU ME
SW IT HY OU

Now replace the repeated letters with X, then the plain text will become

WE WI LX LM EX ET AT TH
EH YD EP AR KC OR NE RB
RI NG AL LT HE RE QU IR
ED DO CU ME SW IT HY OU

Now we will replace this plaintext with the pairs from the matrix according to the rules.

ZR XR GY FH IZ CB HM MB
KB VL RU MI ER PC UB AT
IK DL HG DH BK IC SO KI
CN OV EO BR PY CA LK PO

So the encrypted message will become

ZRXRGYFHIZCBHMMB
KBVLRUMIERPCUBAT
IKDLHGDHBKICSOKI
CNOVEOBRPYCALKPO

This is our final cipher text, which now can be transmitted. But to get the original message again, the receiver should know the key, “Cricket match”

3.1.3. Strengths

The benefits of using this algorithm are,

- It is simple to operate
- Only 1 key is required to remember by both sender and receiver
- Creates 26*26 = 676 digraphs which requires a frequency table of size 676 to analyse encrypted messages [**CN03**]
- Substitution of letter depends on the key selection. As 5*5 matrix is generated according to the key, so the plain text digraphs are replaced with different cipher text digraph with different key selection. It results in improved security of the algorithm.
- Simple cryptanalysis techniques may not work easily to break it.

3.1.4. Limitations

Some of the observed limitations of the cipher are:

- No encryption of the numeric data or punctuation symbols.
- Each digraph is substituted with the same diagraph within one message.
- If someone, other than sender or receiver, gets access to the key, then message can be decrypted easily.

3.1.5. Existing variations of Playfair Algorithm

After searching for some variants of Playfair algorithm, we came across with following:

1. Seriated Playfair Algorithm [**SP04**]
2. 3D Playfair Algorithm [**3DP**]

3.1.5.1. Seriated Playfair Algorithm

In the Seriated algorithm, it writes the letters horizontally in two lines, making group of 5 letters in such a way that diagraphs creates vertically rather than horizontally. It will then encipher vertical pairs according to the Playfair rules. Next example [**SP04**] shows an arrangement of letters done by this algorithm.

Key: LOGARITHM

Original message: Come quickly we need help immediately

Arranged text:

C O M E Q U E N E E D H M E D I A T

I C K L Y W (X)E L P I M E L Y T O M

Matrix:

L O G A R

I T H M B

C D E F K

N P Q S U

V W X Y Z

Cipher:

N L B C S P Q Q C D C M H C F T R H

C D F G X Z G C G Q T B F G W H G B

Final Cipher text:

NLBCS PCDFG XZQQC DCMGC GQTBH CFTRH FGWHG B

3.1.5.2. 3D Playfair Algorithm

In this algorithm, each letter will replace with a specific letter of the given table. For each letter, it checks the coordinates and does some calculations and replace with letter present at that calculated place. Following table [**3DP**] and examples [**3DP**] shows the 3D Playfair algorithm working.

3D Playfair Algorithm table:

X	**n**	**p**	**5**	**c**	**D**	**I**	**m**	**G**	**h**	**2**	**Q**	**8**	**a**	**E**	**y**
1	**b**	**B**	**l**	**N**	**4**	**9**	**0**	**r**		**L**	**7**	**j**	**p**	**6**	**z**
A	C	F	H	R	S	T	U	d	e	f	g	s	t	u	v
J	K	M	O	V	W	X	Z	i	k	o	q	w	x	3	n

Original Message: Make this

Process of encryption:

1. Take the first three letters, “Mak”
2. The coordinates of
   a. M is (0,3,2)
   b. a is (3,0,1)
   c. k is (2,3,1)
3. In order to encrypt ‘M’ we give i_M a displacement $i_{M'}$ = (0+(3+0+1)) % 4 =0

4. Thus, ‘M’ is replaced by the character at (0,3,2), which is ‘M’
5. Similarly the entire sentence “Make this” after encryption becomes “MtoekuaGs”

3.1.6. Summary

After analyzing Playfair algorithm and its existing variations, the Seriated algorithm does not improve original algorithm regarding its way of working and selection of digraphs. It simply chooses digraphs in different way. The 3D algorithm caters for the white space present in the message, but replaces each letter with specific letter and does not works as the original Playfair algorithm works.

3.2. Our Algorithm

Based on the Playfair algorithm and trying to overcome the limitations of it and its variations, we enhance and modify the way it works. Using the idea of Playfair cipher, we propose following algorithm.

We name our algorithm as “Snuffer Cipher”.

3.2.1. Improvements

After analyzing the Playfair cipher, following enhancements are done

- Increase in the size of the matrix from 5*5=25 to 7*6=42
- Inclusion of the digits 0-9 and commonly used punctuation symbols like space & , . ; @
- Randomizing the process of digraph replacement from plaintext to cipher text.
- Separate use of alphabets I and J
- No repetition of the characters within a key

3.2.2. How it works?

Following steps can be observed in encrypting a message using our algorithm

1. The algorithm works in a similar way to Playfair cipher. The first step is to select a key that will be used in creating the 7*6 matrix. The matrix will contain alphabets A-Z, number 0-9 and punctuation symbols (space & , . ; @) in such a way that no letter should repeat. The selected cipher key is important as it helps in creating matrix and encoding, decoding of the message.

2. The key may or may not be private but should be known to both sender and receiver in order to encode or decode the messages.

3. After choosing a key, the message will be divided in pairs in such a way that no diagraph consist of similar letter, number or symbol. If so, replace the repeated letter with white space.

4. Next step is to replace the digraphs with the encoded pair of letters from the matrix according to the size of the selected key. Each digraph will be replaced

with a specific pair from the matrix. Following rules are applied in replacing the original digraph:

a. Calculate the size of the selected key.

b. If the letters, numbers or symbols appear in the same row of the matrix, replace them with the letter, number or symbols on the right side, *Total Size of the key places away* to them respectively. This change will make it different from the original Playfair algorithm rules.

c. Similarly, if the letters appear on the same column of the generated matrix, replace them with the letters below, *Total Size of the key places down* respectively.

d. If the letters 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.

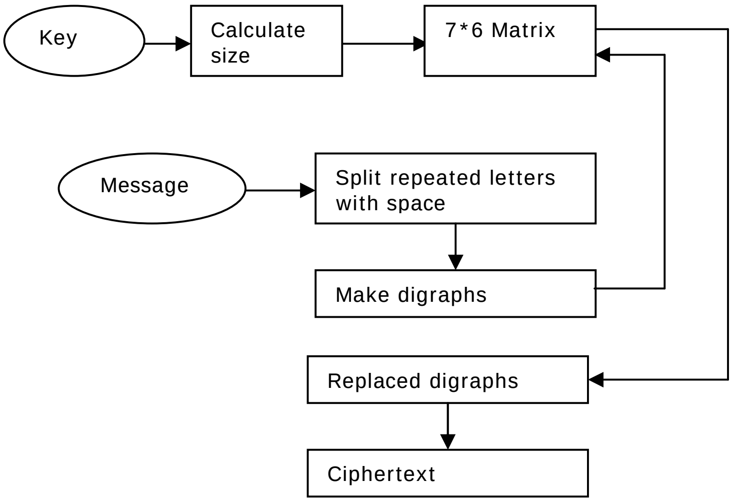


Fig 2. Snuffer Cipher Block diagram

3.2.3. Example

Suppose, we have the same message as above with little modification

“*We will meet at the 10 Hyde Park corner at 2pm.*”

And the key selected for this message is “sh12,”. The key size is 5. Based on this key, following matrix will be created,

$$\begin{bmatrix} s & h & 1 & 2 & , & a \\ b & c & d & e & f & g \\ i & j & k & l & m & n \\ o & p & q & r & t & u \\ v & w & x & y & z & 0 \\ 3 & 4 & 5 & 6 & 7 & 8 \\ 9 & 0 & & ? & @ & ; \end{bmatrix}$$

The next step is to create digraphs of the message and with numbers and punctuation symbols and separate repeated letter with space. Also add extra space in the end if last letter left alone.

we _w il l_ me et

_a t_ th e_ 10 _h

yd e_ pa rk _a t_

2p m.

* white space shown with _

Next step is to arrange digraphs, then we get

we _w il _l _m e_

et _a t_ th e_ 10

_h yd e_ pa rk _a

t_ 2p m.

Now we will replace this plaintext with the pairs from the matrix according to the rules. Size of the key is 5.

Xb wj nk &l .ib &e q;

so .o 2b &, ww 4w db

&t sp k; so .s qk ._

So the encrypted message will become:

Xbwjnk&l.ib&eq;so.o2b&,ww4wdb&tspk;so.sqk._

This is our final cipher text which now can be transmitted. But to get the original message again, the receiver should know the key, “sh12,” and size of the key.

When the receiver applies the key on the encrypted text, the original text will be

we wil l me et at the 10 hyde park at 2pm.

3.2.4. Strengths

1. Handles alphabets as well as Numeric data.
2. Has additional Shift Characters facility in Encryption/Decryption.
3. Since Key Size is dynamic since it is based on key itself, this makes the algorithm more secure.
4. Apart from alphanumeric characters, this algorithm also caters the most common special characters that include comma, space, ampersand, period, semi colon, and question mark.
5. In the original algorithm, repeated characters were distributed with any fixed character and when recombined, it made the original text a bit more difficult to read due to the additional character, however in this algorithm period is used making it more readable.
6. Is suitable for sending large data due to easy data structure.

3.2.5. Limitations

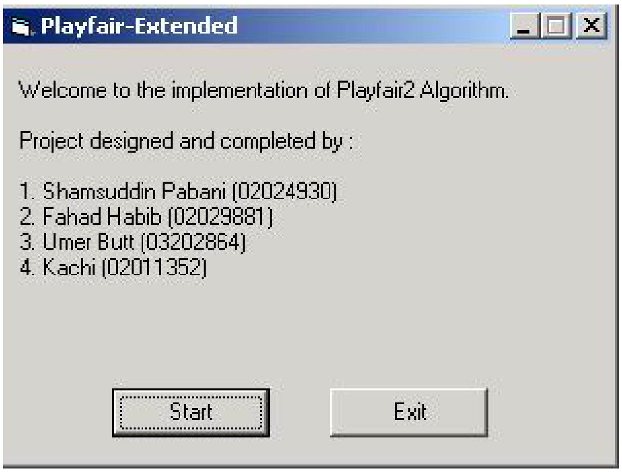
1. This algorithm does not differentiate between small and capital letters.
2. It does not cater special characters apart from the ones stated above.
3. On decryption, it does not any mechanism to remove periods that were added due to concurrent repetition of characters.
4. It only allows 200 characters to be encoded due to limitation in implementation.

4. IMPLEMENTATION

To implement Snuff Cipher, we use Visual Basic 6 as our programming language. The following section will describe step by step of it as implemented.

4.1. Welcome Screen:

This is the welcome screen for the Implementation giving the introduction to the team members.



Playfair-Extended

Welcome to the implementation of Playfair2 Algorithm.

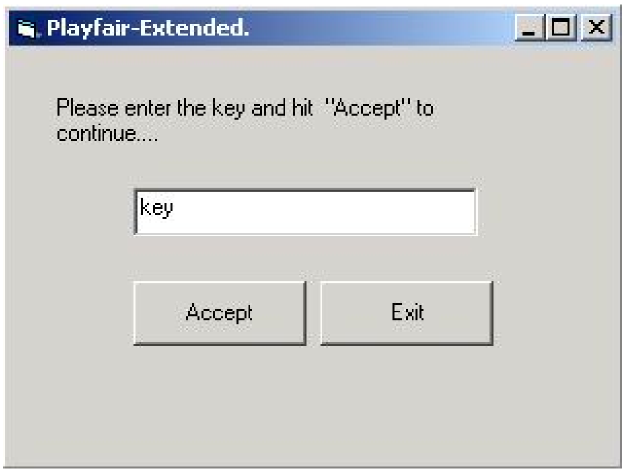
Project designed and completed by :

1. Shamsuddin Pabani (02024930)
2. Fahad Habib (02029881)
3. Umer Butt (03202864)
4. Kachi (02011352)

Fig 3. Welcome screen

4.2. Key Input Screen:

The key is taken as an input in the second screen and is checked against the rules, which include checking for repetitive characters, out-of-range characters (not within the domain covered by the algorithm) and empty strings.



Playfair-Extended.

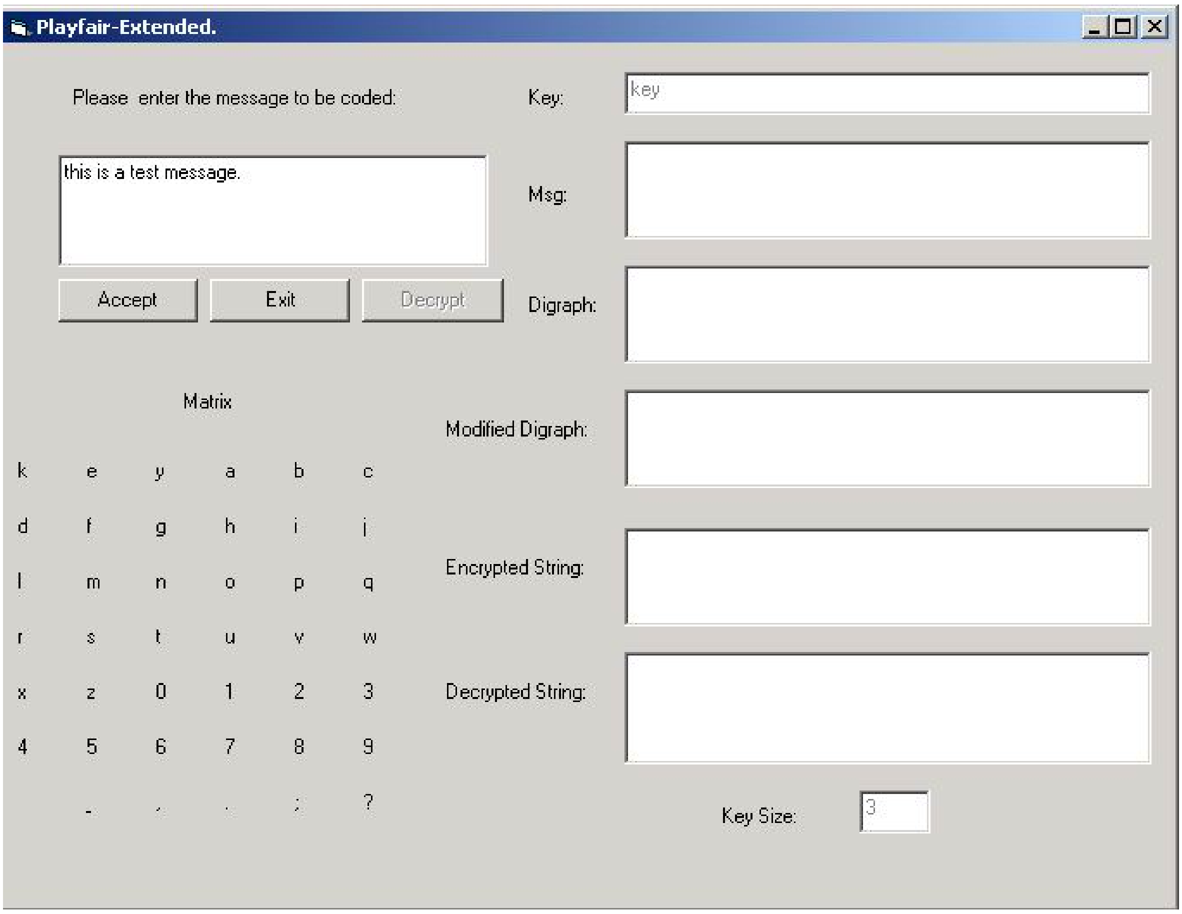
Please enter the key and hit "Accept" to continue....

key

Fig 4. Enter Key

4.3. Main Screen:

This screen shows the key entered, the key size at the bottom right and the matrix that has been developed using the given values. It also displays a text area for allowing the user to enter the message that has to be encrypted as well as different text boxes that would be showing result as the algorithm proceeds.



Please enter the message to be coded:

this is a test message.

Key: key

Matrix

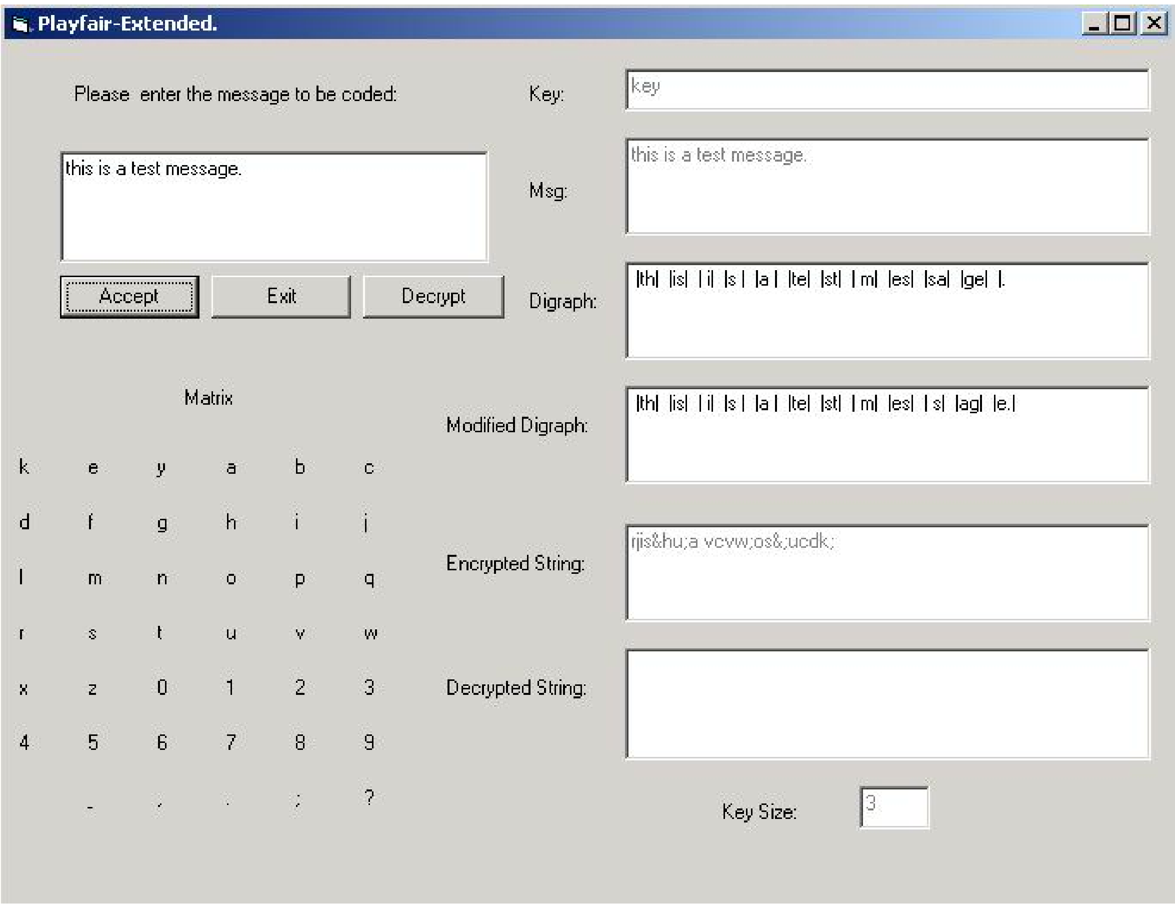
k	e	y	a	b	c
d	f	g	h	i	j
l	m	n	o	p	q
r	s	t	u	v	w
x	z	0	1	2	3
4	5	6	7	8	9
	-	,	.	;	?

Key Size: 3

Fig 5. Main Screen

4.4. Encryption Screen:

This screen shows the initial digraph, the modified digraph (when the rules are applied) and the encrypted string as created by the matrix. The character "|" delimits the digraphs in both the initial and the modified digraphs.



Please enter the message to be coded:

this is a test message.

Key: key

Msg: this is a test message.

Digraph: |th| |is| | i| |s | |a | |te| |st| | m| |es| |sa| |ge| |.

Modified Digraph: |th| |is| | i| |s | |a | |te| |st| | m| |es| | s| |ag| |e.|

Encrypted String: rjis&hu;a vcvw;os&;ucdk;

Decrypted String:

Key Size: 3

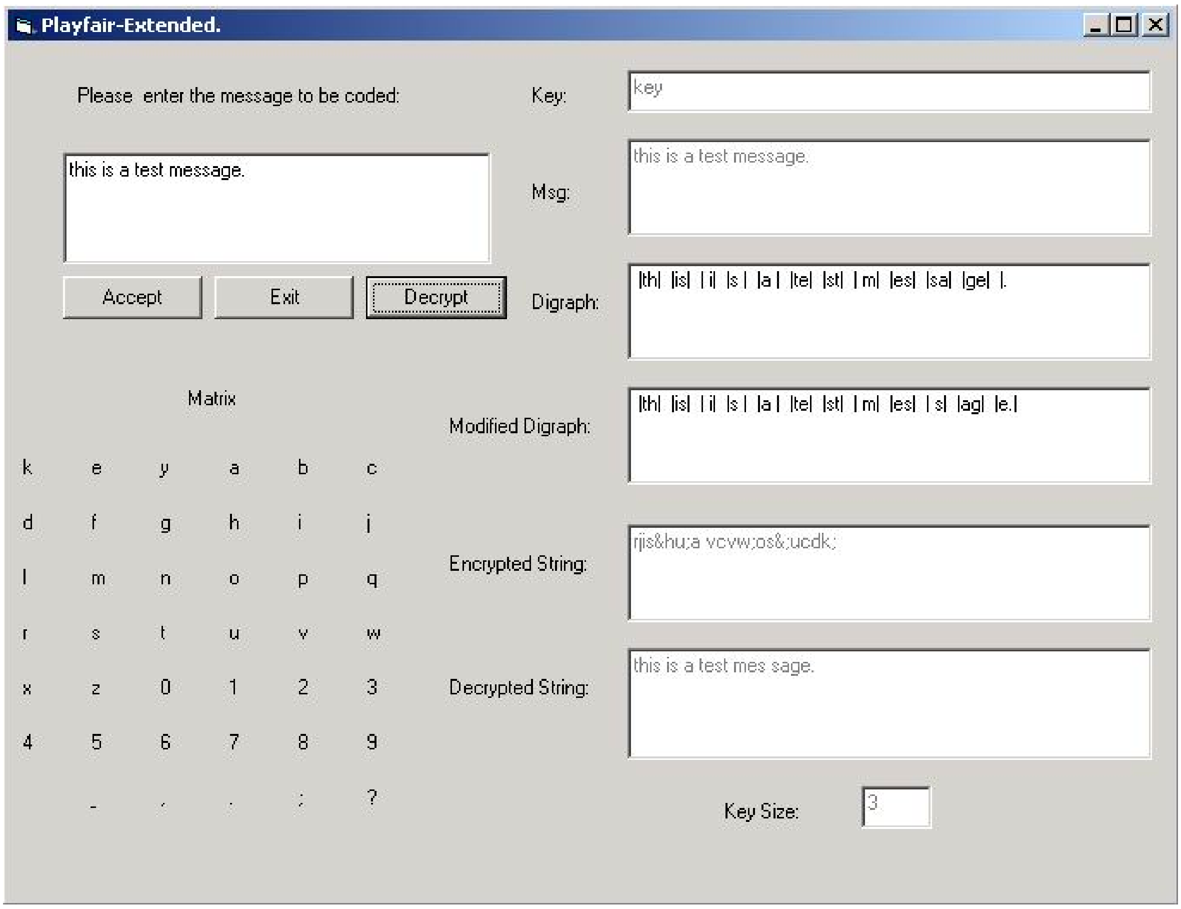
Matrix

k	e	y	a	b	c
d	f	g	h	i	j
l	m	n	o	p	q
r	s	t	u	v	w
x	z	0	1	2	3
4	5	6	7	8	9
	-	,	.	;	?

Fig 6. Encryption screen

4.5. Decryption Screen:

This is the final screen when the user decrypts the message, showing all the steps in the screen. The decrypted string does not cater for the periods that have been introduced during encryption process when there were consequent repetitions of characters.



Please enter the message to be coded:

this is a test message.

Key: key

Msg: this is a test message.

Digraph: |th| |is| | i| |s | |a | |te| |st| | m| |es| |sa| |ge| |.

Modified Digraph: |th| |is| | i| |s | |a | |te| |st| | m| |es| | s| |ag| |e.|

Encrypted String: rjis&hu;a vcvw;os&;ucdk;

Decrypted String: this is a test mes sage.

Key Size: 3

Matrix

k	e	y	a	b	c
d	f	g	h	i	j
l	m	n	o	p	q
r	s	t	u	v	w
x	z	0	1	2	3
4	5	6	7	8	9
	-	,	.	;	?

Fig 7. Decryption Screen

5. REFERENCES AND BIBLIOGRAPHY

[**3DP**] **Atal Chaudhuri**, *3D Playfair Cipher, Department Of Computer Science & Engineering, Jadavpur University* (Electronic Source: URL: http://www.eissa.org/Zip/Network%20Security.pps, Visited on: March 25, 2004)

[**CN03**] **William Stallings**, *Playfair Cipher, Chapter 2: Classical Encryption Techniques, Cryptography and Network Security: Principles and Practice 3/e*, Prentice Hall Publishers, 2003

[**PA04**] *Playfair Cipher* (Electronic Source: URL: http://www.fact-index.com/p/pl/playfair_cipher.html, Visited on March 18, 2004, 16:34)

[**PA04a**] **Ben Goren**, *The Playfair Cipher* (Electronic Source: URL: http://www.trumpetpower.com/Papers/Crypto/Playfair, Visited on: March 17, 2004)

[**WK04**] *Frequency Analysis,* Wikipedia Encyclopaedia (Electronic Source: URL: http://en.wikipedia.org/wiki/Frequency_analysis, Visited on: March 23, 2004)

[**WK04p**] *Playfair Cipher,* Wikipedia Encyclopaedia (Electronic Source: URL: http://en.wikipedia.org/wiki/Playfair_cipher, Visited on: March 23, 2004)

[**SP04**] *Seriated Playfair Cipher, Department of Mathematics and Computer Sciences, Weizmann Institute* (Electronic Source: URL: http://www.wisdom.weizmann.ac.il/~albi/cryptanalysis/lect3.htm, Visited on: March 25, 2004)