CS 413
Information Security

Course Instructor

Naushad Siddiqui

Head of IT Audit, Sui Southern Gas Company

Visiting Faculty, NED University
Director & Faculty Member, ISACA Karachi

CS413 Information Security

Fall 2020

Week 04

Agenda

- Playfair Cipher
- Playfair Cipher Variants

Playfair Cipher

Playfair Cipher

- Playfair Cipher was the first practical digraph substitution cipher.
- The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher.
- In playfair cipher unlike traditional cipher we encrypt a pair of alphabets (digraphs) instead of a single alphabet.
- It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II.
- This was because Playfair is reasonably fast to use and requires no special equipment.

Playfair Cipher

- The best-known multiple-letter encryption cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext digrams.
- The Playfair algorithm is based on the use of a 5x5 matrix of letters constructed using a keyword.
- The rules for filling in this 5x5 matrix are: L to R, top to bottom, first with keyword after duplicate letters have been removed, and then with the remain letters, with I/J used as a single letter.

Playfair Cipher - Encryption

The Algorithm consists of 2 steps:

Step 1: Generate the key Square(5×5):

- The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext.
- The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.
- Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets).
- All the letters should be written in CAPITAL letter, in pairs, without punctuation, and all Js are replaced with Is
- No repeat letter

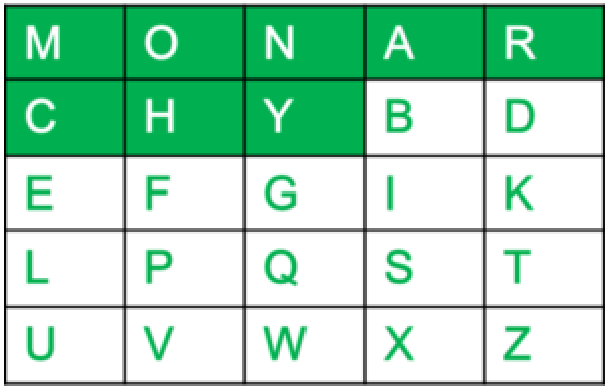
Playfair Cipher - Encryption

- ***For example:***

  If the key is ‘monarchy’ then the initial entries are

  'm', 'o', 'n', 'a', 'r', 'c', 'h', 'y'

  followed by remaining characters of a-z (except 'j') in that order



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

Playfair Cipher - Encryption

Step 2: Algorithm to encrypt the plain text

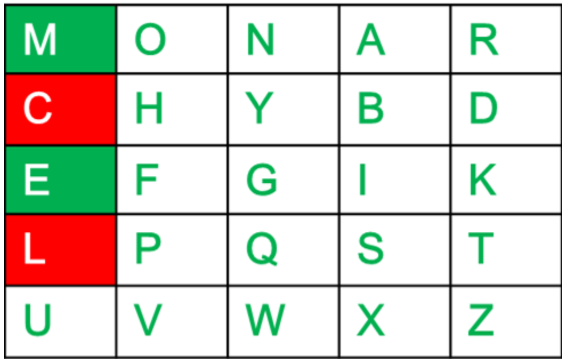
- The plaintext is split into pairs of two letters (digraphs).
- If there is an odd number of letters, a Z is added to the last letter.

For example:

- Plain Text: ‘instruments’
- After Split: 'in' 'st' 'ru' 'me' 'nt' 'sz‘

Three (3) Rules for Encryption

(1) If both the letters are in the same column:
Take the letter below each one
(going back to the top if at the bottom).
Diagraph: ‘me’ Encrypted Text: ‘cl’
Encryption: m → c e → l



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

Playfair Cipher - Encryption

Rules for Encryption (… cont…)

(2) If both the letters are in the same row:

Take the letter to the right of each one (going back to the leftmost if at the rightmost position).

Diagraph: ‘st’ Encrypted Text: ‘tl’

Encryption: s → t t → l

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

(3) If neither of the above rules is true:

Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Diagraph: ‘nt’ Encrypted Text: ‘rq’

Encryption: n → r t → q

Note: The order is important – the first encrypted letter of the pair is the one that lies on the same row as the first plaintext letter

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

Playfair Cipher - Encryption

Complete Example:

Plain Text: ‘instrumentsz’

Encrypted Text: ‘gatlmzclrqtx’

Encryption:

i → g *n → a* *s → t* *t → l*
r → m *u → z* *m → c* *e → l*
n → r *t → q* *s → t* *z → x*

in:

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

st:

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

ru:

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

me:

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

nt:

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

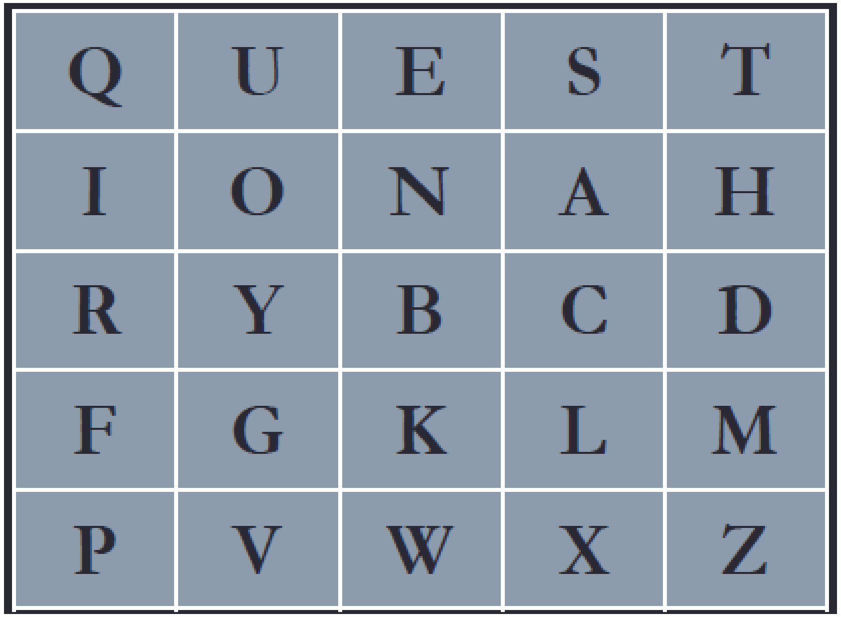
sz:

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

Playfair Cipher - Encryption

Plain Text : Joel enjoys programming on Friday nights! Yah

Key : Question Authority



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

Playfair Cipher - Encryption

We split the text we are encrypting into digraphs. We pull out all non-alphabetical characters (multiple instances of the same letter is okay though!).

Here is what my example would look like: "IO EL EN IO YS PR OG RA MX MI NG ON FR ID AY NI GH TS YA HX".

There are three edge-cases you need to consider.

In digraph MM (from "programming"), we inserted an ‘X’

If there is an odd number of letters in the text phrase, just put an X at the end of the text. In our case, letter ‘H’ (from "Yah") is padded with ‘X’

As we replaced 'J' with 'I' to create the matrix, we need to apply this same rule to input text

So applying the three rules we studied earlier, the resultant Cipher Text:

Cipher Text :

ONSKNBONCUQFYVCILZFHOKNAPFHROCAOMOQTCOAZ

Playfair Cipher - Decryption

- Decrypting the Playfair cipher is as simple as doing the same process in reverse.
- The receiver has the same key and can create the same key table, and then decrypt any messages made using that key.
- Shift up and left instead of down and right
- Drop extra X
- Locate any missing any “I”s that should be “J”s

Playfair Cipher - Decryption

Complete Example:

Plain Text: ‘gatlmzclrqtx’

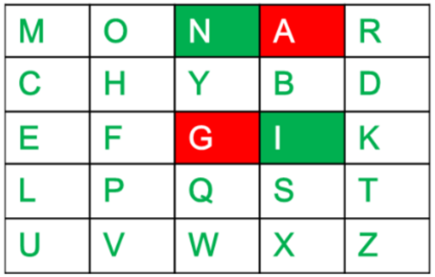
Encrypted Text: ‘instrumentsz’

Encryption:

ga → in *tl → st* *mz → ru*

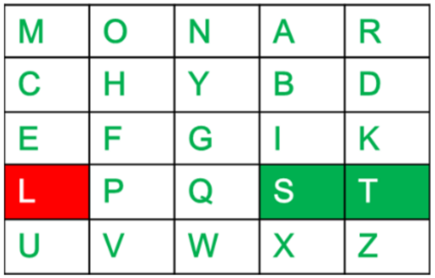
cl → me *rq → nt* *tx → sz*

in:



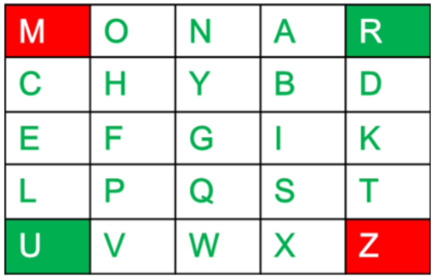
M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

st:



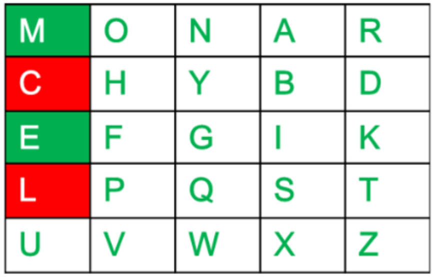
M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

ru:



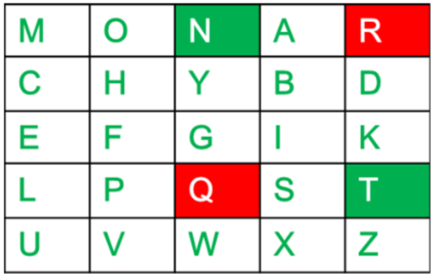
M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

me:



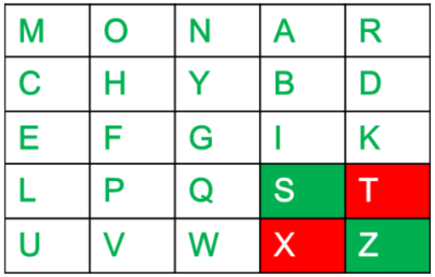
M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

nt:



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

sz:

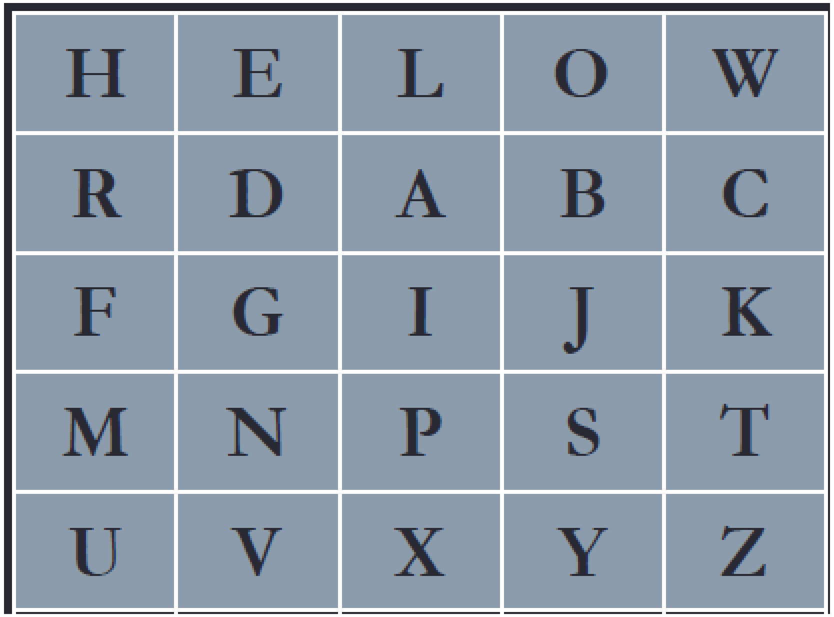


M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

Playfair Cipher - Decryption

Cipher Text : LF GD MW DN WO CV

Key : Hello World (Remove Q)



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

Plain Text : hide the gold

Security of Playfair Cipher

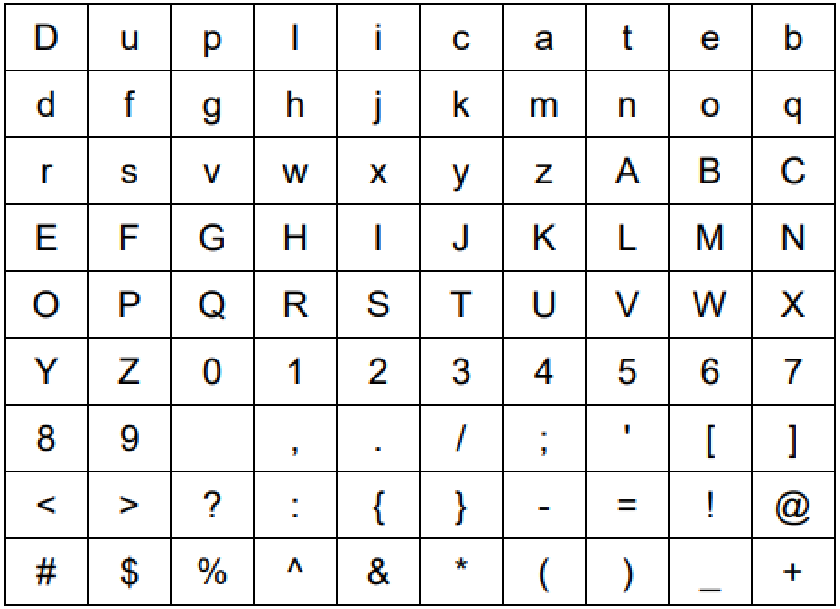
- The Playfair cipher is a great advance over simple monoalphabetic ciphers, since there are 26*26=676 digrams (vs 26 letters), so that identification of individual digrams is more difficult.
- The relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult.
- The Playfair cipher was for a long time considered unbreakable. However, its simple transposition offering a relatively weak method of encryption
- Despite this level of confidence in its security, the Playfair cipher is relatively easy to break because it still leaves much of the structure of the plaintext language intact.
- Playfair is now regarded as insecure for any purpose because modern hand-held computers could easily break the cipher within seconds

Playfair Cipher Variants

10 x 9 Matrix

- The 10 x 9 matrix contains almost all the printable characters.
- This includes lowercase and uppercase alphabets, punctuation marks, numbers and special characters.
- The order of placement of different groups of characters can also be done so that the matrix formed by using the same secret keyword depends on the order of placement.
- This means that the ciphertext will also depend on the order of placement of different groups of characters.

The matrix with the secret keyword as ***Duplicate*** *and a particular placement order is given in below Table*



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		,	.	/	;	'	[	]
<	>	?	:	{	}	-	=	!	@
#	$	%	^	&	*	(	)	_	+

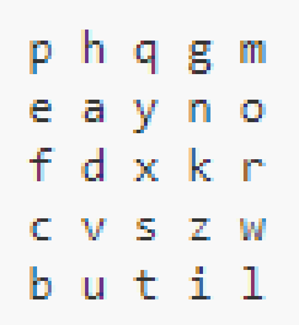
ADFGX Ciphers

- The ADFGX cipher was a field cipher used by the German Army during World War I.
- ADFGX is a fractionating transposition cipher
- The cipher is named after the five possible letters used in the ciphertext: A, D, F, G and X.
- These letters were chosen deliberately because they sound very different from each other when transmitted via morse code.
- The intention was to reduce the possibility of operator error.

ADFGX Ciphers

Algorithm

- The 'key' for a ADFGX cipher is a 'key square' and a key word. e.g.



```
p h q g m
e a y n o
f d x k r
c v s z w
b u t i l
```

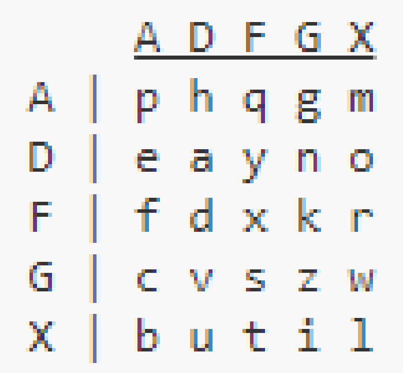
- The Key Square is a 5 by 5 square containing all the letters except 'j'.
- The Key Word is any word e.g. GERMAN

- There are a number of steps involved.

ADFGX Ciphers

Algorithm (2)

- ***Step 1:*** Build a table like the following with the key square. This is known as a polybius square.



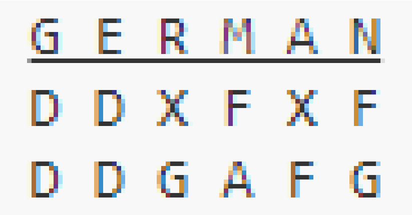
	A	D	F	G	X
A	p	h	q	g	m
D	e	a	y	n	o
F	f	d	x	k	r
G	c	v	s	z	w
X	b	u	t	i	l

- ***Step 2:*** Encode the plaintext using this matrix. To encode the letter 'a', locate it in the matrix and read off the letter on the far left side on the same row, followed by the letter at the top in the same column.
- In this way each plaintext letter is replaced by two cipher text letters. E.g. 'attack' -> 'DD XF XF DD GA FG'.
- The ciphertext is now twice as long as the original plaintext. Note that so far, it is just a simple substitution cipher, and trivial to break.

ADFGX Ciphers

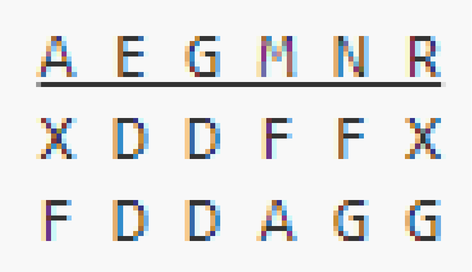
Algorithm (3)

- ***Step 3:*** Write the code word with the enciphered plaintext underneath



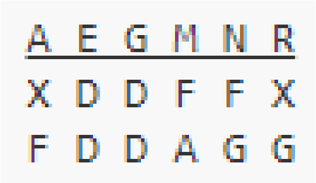
G	E	R	M	A	N
D	D	X	F	X	F
D	D	G	A	F	G

- ***Step 4:*** Perform a columnar transposition. Sort the code word alphabetically, moving the columns as you go. i.e. shuffle the columns into alphabetical order
- Note that the letter pairs that make up each letter get split apart during this step, this is called fractionating.



A	E	G	M	N	R
X	D	D	F	F	X
F	D	D	A	G	G

- ***Step 5:*** Read the final ciphertext off in columns.
- -> XF DD DD FA FG XG



A	E	G	M	N	R
X	D	D	F	F	X
F	D	D	A	G	G

End of Week 04