❑ Beaufort Cipher

The Beaufort cipher, created by **Sir Francis Beaufort**, is a substitution cipher that is similar to the Vigenère cipher, with a slightly modified enciphering mechanism and tableau.

Encryption:

$$C_i = (k_i - p_i) \bmod 26 \ldots\ldots(11).$$

Locate the plaintext letter in the top row of the table. Search the column immediately under till the key letter is found. Follow the row of the key letter to the left. The crypto letter is found in the leftmost column.

Decryption:

$$p_i = (k_i - C_i) \bmod 26 \ldots\ldots(12).$$

Locate the crypto letter in the leftmost column of the table. Search the row to the right till the key letter is found. Go straight up from the key letter. The plaintext is found in the top row. The Beaufort way of using the table is somewhat easier than standard Vigenère since you only have to follow one route instead of finding an intersection of a row and a column.

Ciphertext \ Plaintext	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	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	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
C	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
D	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
E	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
F	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
G	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
H	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
I	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
J	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
K	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
L	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
M	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
N	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
O	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
P	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
Q	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
R	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
S	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
T	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
U	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
V	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
W	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
X	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
Y	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
Z	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

Table (2.4): Beaufort Table or Tabula Recta or Beaufort Square

Plaintext Alphabet	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
Plaintext Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Example: Encrypt the message "**defend the east wall of the castle**", using Beaufort cipher for the given keyword is **fortification**.

Ans. : Ciphertext: "CKMPVC PVW PIWU JOGI UA PVW RIWUUK"

Key: k	f 5	o 14	r 17	t 19	i 8	f 5		i 8	c 2	a 0		t 19	i 8	o 14	n 13		f 5	o 14	r 17	t 19
Plaintext: p	d 3	e 4	f 5	e 4	n 13	d 3		t 19	h 7	e 4		e 4	a 0	s 18	t 19		w 22	a 0	l 11	l 11
Ciphertext: C $C_i = (k_i - p_i)$ mod 26	2 C	10 K	12 M	15 P	21 V	2 C		15 P	21 V	22 W		15 P	8 I	22 W	20 U		9 J	14 O	6 G	8 I

Key: k	i 8	f 5		i 8	c 2	a 0		t 19	i 8	o 14	n 13	f 5	o 14
Plaintext: p	o 14	f 5		t 19	h 7	e 4		c 2	a 0	s 18	t 19	l 11	e 4
Ciphertext: C $C_i = (k_i - p_i)$ mod 26	20 U	0 A		15 P	21 V	22 W		17 R	8 I	22 W	20 U	20 U	10 K

iii. Vernam Cipher (Stream)

The ultimate defense against such cryptanalysis is to choose a keyword that is as long as the plaintext and has no statistical relationship to it. Such a system was introduced by an AT&T engineer named **Gilbert Vernam**. His system works on binary data (bits) rather than letters. The system can be expressed succinctly as follows (Figure 2.5):

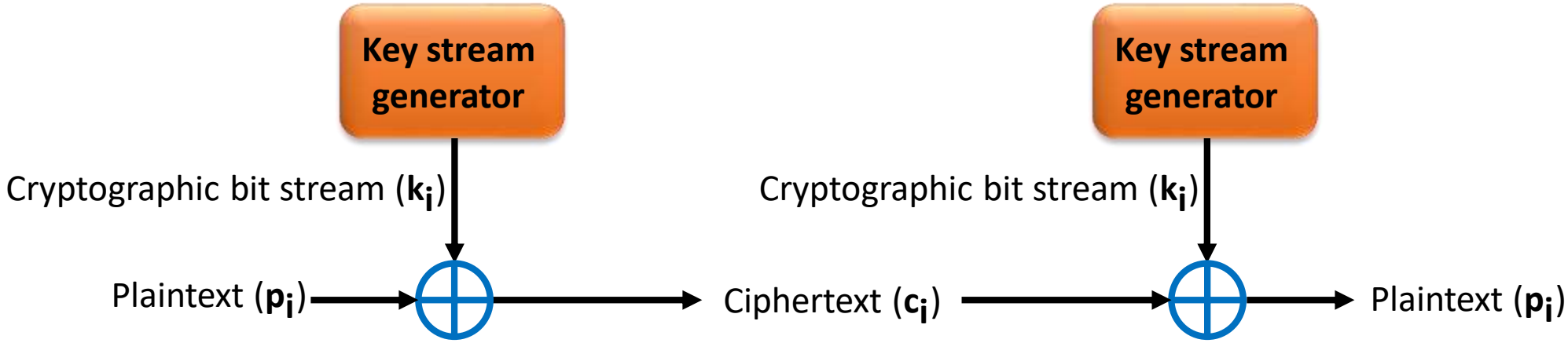


Figure (2.5): Vernam Cipher

Encryption:

$$C_i = p_i \oplus k_i \text{.....(13).}$$

Decryption:

$$p_i = C_i \oplus k_i \text{.....(14).}$$

Where:
p_i = i-th binary digit of plaintext
k_i = ith binary digit of key
C_i = ith binary digit of ciphertext
$\oplus$ = exclusive-or (XOR) operation

The essence of this technique is the means of construction of the key. Vernam proposed the use of a running loop of tape that eventually repeated the key so that in fact the system worked with a very long but repeating keyword. Although such a scheme, with a long key, presents formidable cryptanalytic difficulties, it can be broken with sufficient ciphertext, the use of known or probable plaintext sequences, or both.

Example: Encrypt the message "HELLO", using the key "**PLUTO**".

Ans. : 1. Obtain the binary ASCII character code for each letter of the plaintext

H = 72	E = 69	L = 76	L = 76	O = 79
01001000	01000101	01001100	01001100	01001111

To represent any decimal number by binary	
$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	128 64 32 16 8 4 2 1

2. Obtain the binary ASCII character code for each letter of the key

P = 80	L = 76	U = 85	T = 84	O = 79
01010000	01001100	01010101	01010100	01001111

3. Carry out the XOR operation, applying it to each corresponding pair of bits: Ciphertext is

CAN (cancel)	TAB	EM (end of medium)	CAN (cancel)	NUL (null)
00011000	00001001	00011001	00011000	00000000

Where: each of CAN, TAB, EM, NUL represent one character

iv. One Time Pad

An Army officer, **Joseph Mauborgne**, proposed an improvement to the Vernam cipher that yields the ultimate in security. **Mauborgne** suggested using a random key that is as long as the message so that the key need not be repeated. In addition, the key is to be used to encrypt and decrypt a single message and then is discarded. Each new message requires a new key of the same length as the new message. Such a scheme, known as a **One-Time Pad (OTP),** is unbreakable. It produces random output that bears no statistical relationship to the plaintext. Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code. The **OTP** cipher has:

1- Key is generated random, as long as the message is used, the cipher will be secure

2- Key Size = Message Size

3- Every message has its key

4- Key is not used more than one time

5- Is unbreakable since ciphertext bears no statistical relationship to the plaintext

6- Problems in generation & safe distribution of key

7- Is the only one with Perfect Security

The one-time pad is of limited utility and is useful primarily for low-bandwidth channels requiring very high security. The one-time pad is the only cryptosystem that exhibits what is referred to as **perfect secrecy**.

v- Polygraphic

Polygram substitution ciphers encipher a block of letters at the time, rather than a single letter; this makes cryptanalysis harder, as it destroys the single letter frequency distribution. Polygraphic substitution divides the plaintext into groups of letters. Then, they replace each group of letters by one of the predefined letters, numbers, graphic symbols, or by another group of characters.

❑ Playfair cipher

- This cipher was actually invented by British scientist Sir **Charles Wheatstone**, but it bears the name of his friend **Baron Playfair** who championed the cipher at the British foreign office.
- 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 cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are **26 × 26 = 676 digrams**, so that, the identification of individual digrams is more difficult.
- Furthermore, the relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult. For these reasons, the Playfair cipher was for a long time considered unbreakable.
- It was used as the standard field system by the British Army in World War I and still enjoyed considerable use by the U.S. Army and other Allied forces during World War II.

Playfair Algorithm

- The Playfair algorithm is based on the use of a matrix 5 × 5 of letters constructed using a keyword.
- The matrix is constructed by filling in the letters of the keyword from left to right and from top to bottom, but the repeated letters should be omitted (minus duplicates). For example, use the keyword is **monarchy**.
- Then fill in the remainder of the matrix with the remaining letters in alphabetic order.
- The letters I and J count as one letter (combining I and J in one cell).
- The Plaintext is encrypted two letters at a time, according to the following rules:

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

1. Repeating plaintext letters that are in the same pair are separated with a **filler letter**, such as **x**, so that **balloon** would be treated **as ba lx lo on**.

2. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, **ar** is encrypted as **RM**.

3. Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, **mu** is encrypted as **CM**.

4. Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, **hs** becomes **BP** and **ea** becomes **IM** (or **JM**, as the encipherer wishes). (Find the h and s in the square and locate the letters at opposite corners of the rectangle they form).

Example: Encrypt the message "**instruments**", using Playfair cipher for the given keyword "**monarchy**".

Ans. :

Algorithm to encrypt the plain text:

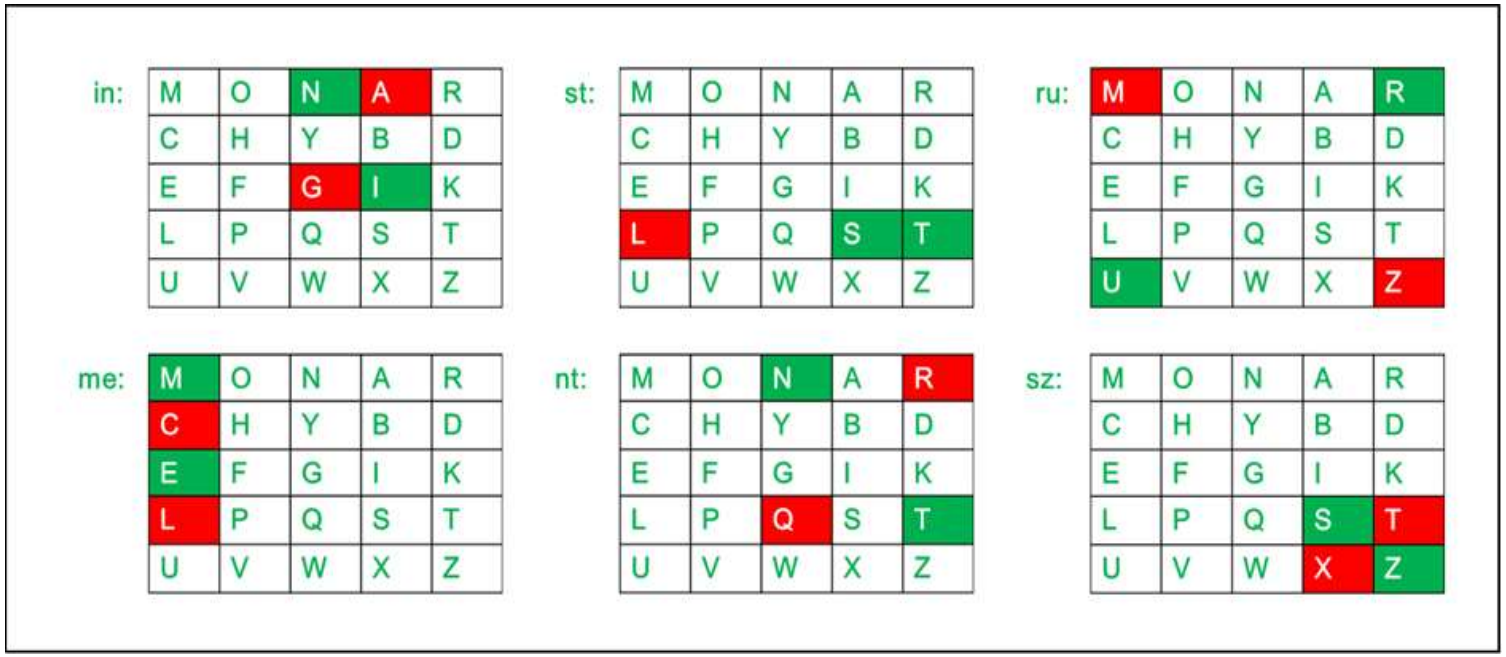
1- Generate the key Square **(5×5):** The key square is a **5×5** grid of alphabets that acts as the key for encrypting the plaintext.

2- 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. Plaintext: in st ru me nt s**z**

Rules for Encryption:

- **If the two letters are in the same column:** Take the letter below each one (going back to the top if at the bottom).
- **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).
- **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.

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



Plaintext	in	st	ru	me	nt	sz
Ciphertext	GA	TL	MZ	CL	RQ	TX

Example: Decrypt the ciphertext "**GATLMZCLRQTX**", using Playfair cipher for the given keyword "**monarchy**".

Ans. : The ciphertext is split into pairs of two letters (digraphs).

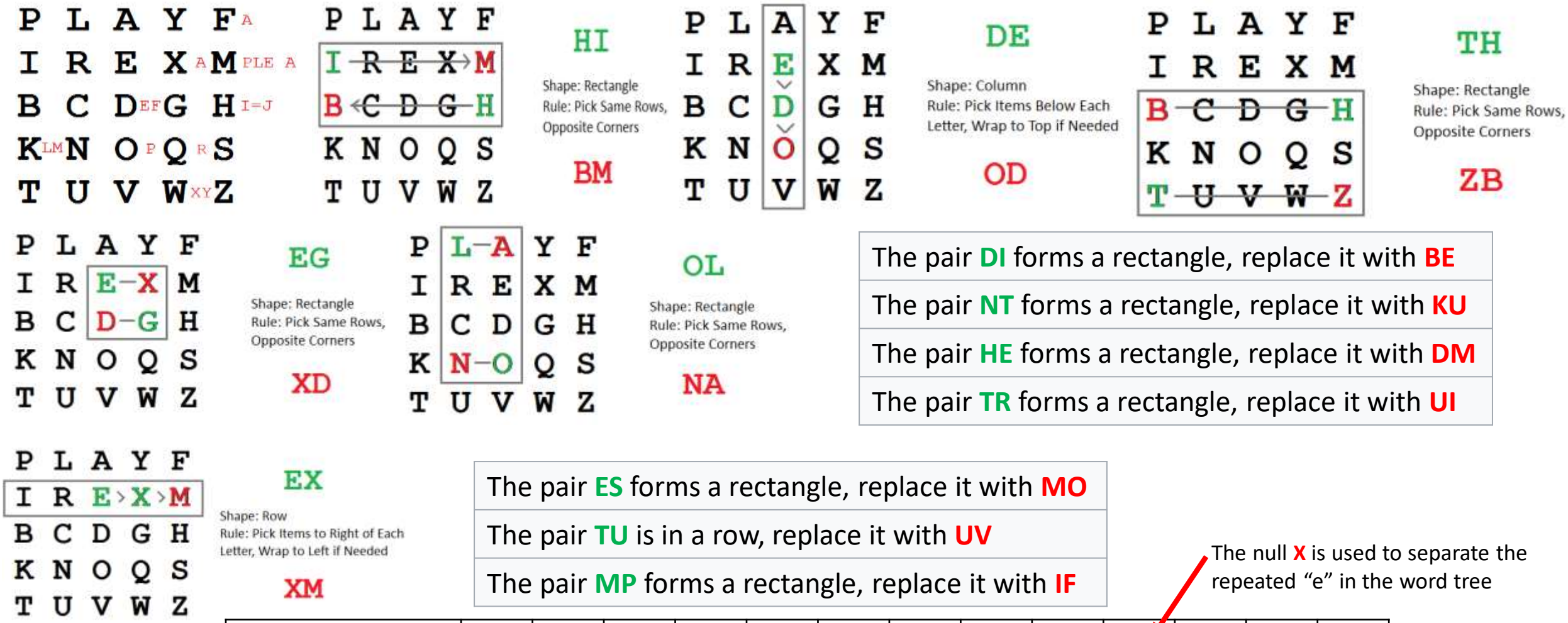
Note: The ciphertext always has an **even** number of characters.

Rules for Decryption:

- **If both the letters are in the same column**: Take the letter above each one (going back to the bottom if at the top).
- **If both the letters are in the same row**: Take the letter to the left of each one (going back to the rightmost if at the leftmost position).
- **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.

Example: Encrypt the message "**hide the gold in the tree stump**", using Playfair cipher for the given keyword "**playfair example**". (note the null "X" used to separate the repeated "E"s) :

Ans. : the table becomes (omitted letters in red): The Plaintext pairs: hi de th eg ol di nt he tr ex es tu mp



The pair **DI** forms a rectangle, replace it with **BE**
The pair **NT** forms a rectangle, replace it with **KU**
The pair **HE** forms a rectangle, replace it with **DM**
The pair **TR** forms a rectangle, replace it with **UI**

The pair **ES** forms a rectangle, replace it with **MO**
The pair **TU** is in a row, replace it with **UV**
The pair **MP** forms a rectangle, replace it with **IF**

The null **X** is used to separate the repeated "e" in the word tree

Plaintext	hi	de	th	eg	ol	di	nt	he	tr	ex	es	tu	mp
Ciphertext	BM	OD	ZB	XD	NA	BE	KU	DM	UI	XM	MO	UV	IF