5- Polyalphabetic Ciphers:

Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message. The general name for this approach is **polyalphabetic substitution cipher**.

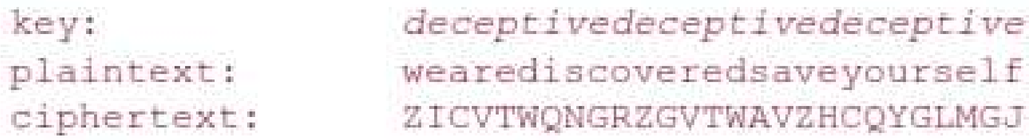
The best known, and one of the simplest, such algorithm is referred to as the **Vigenère cipher**.

1- Vigenère cipher:

use Vigenère tableau is constructed (Table below). Each of the 26 ciphers is laid out horizontally, with the key letter for each cipher to its left. The process of encryption is simple: Given a key letter ***x*** and a plaintext letter **y**, the ciphertext letter is at the intersection of the row labeled *x* and the column labeled y; in this case the ciphertext is V.

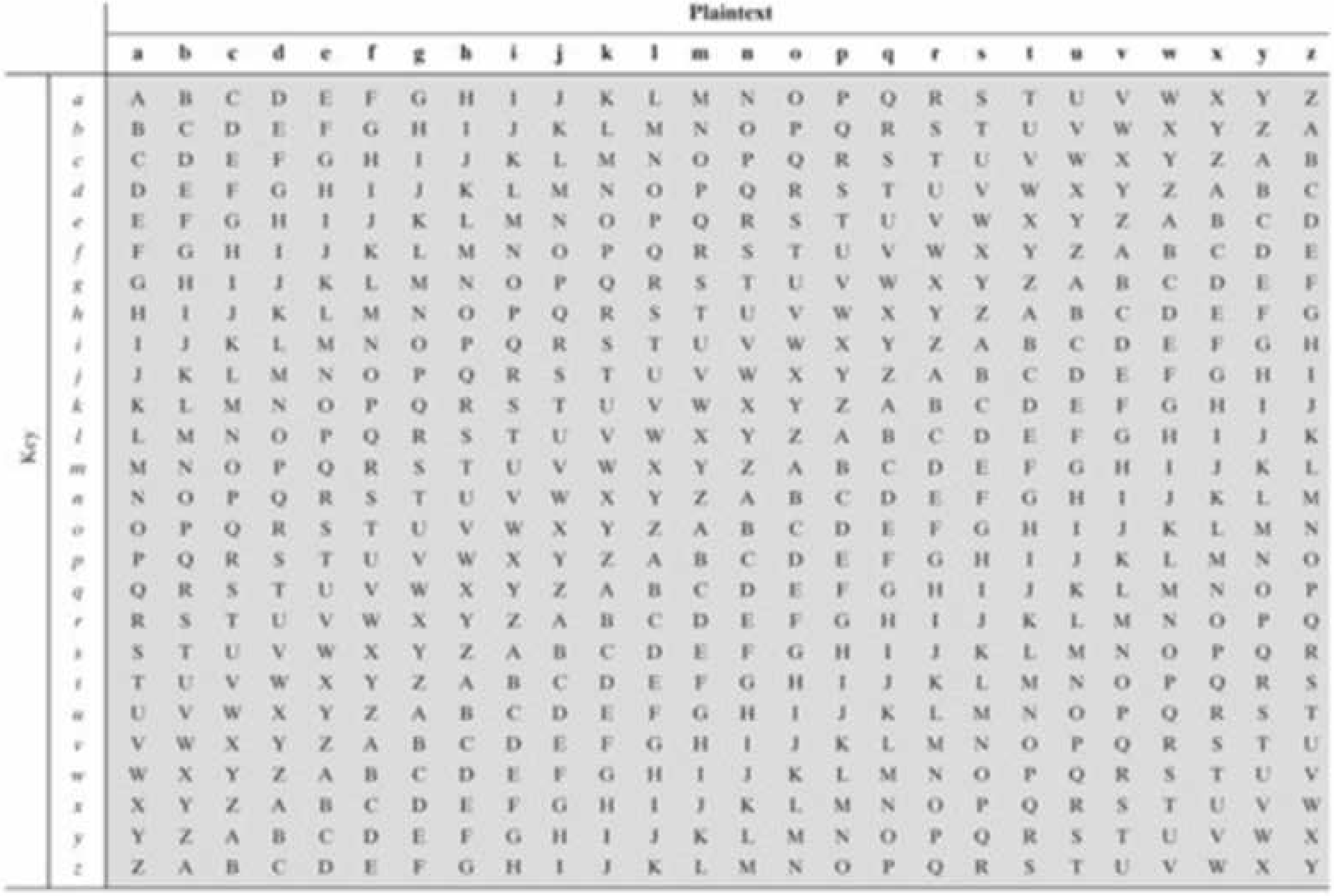
A key is needed that is as long as the message. Usually, the key is a repeating keyword.

Example:



key:	*deceptivedeceptivedeceptive*
plaintext:	wearediscoveredsaveyourself
ciphertext:	ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Decryption is equally simple. The key letter again identifies the row. The position of the ciphertext letter in that row determines the column, and the plaintext letter is at the top of that column.



		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
Key	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

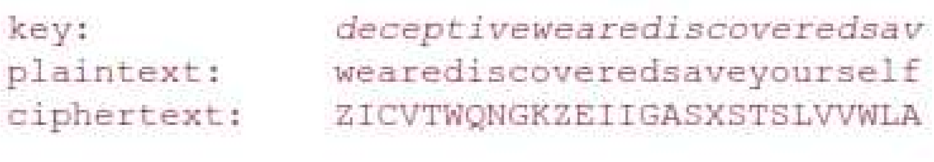
The Modern Vigenère Tableau

By Marwa Al-Musawy

2- Autokey system

The periodic nature of the keyword can be eliminated by using anonrepeating keyword that is as long as the message itself.

Example :

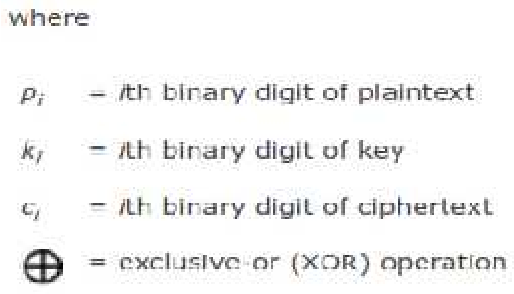


```
key:          deceptivewearediscoveredsav
plaintext:    wearediscoveredsaveyourself
ciphertext:   ZICVTWQNGKZEIIGASXSTSLVVWLA
```

3- Vernam cipher:

The system works on binary data rather than letters , the keyword is chosen as long as the plaintext and has no statistical relationship to it Vernam systems work with very long message but repeating keywrd. The system can be expressed succinctly as follows:

$$c_i = p_i \oplus k_i$$



where

p_i = *i*th binary digit of plaintext

k_i = *i*th binary digit of key

c_i = *i*th binary digit of ciphertext

$\oplus$ = exclusive-or (XOR) operation

The decryption simply involves the same bitwise operation:

$$p_i = c_i \oplus k_i$$

Example :

Pi	$\oplus$	Ki	= Ci		Ci	$\oplus$	Ki =	Pi
0		0	0		0		0	0
0		1	1		1		1	0
1		0	1		1		0	1
1		1	0		0		1	1

6- One-Time Pad

It 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. length as the new message.

One-time pad, is unbreakable. It produces random output that bears no statistical relationship to the plaintext.

In fact, given any plaintext of equal length to the ciphertext, there is a key that produces that plaintext.

The one-time pad offers complete security but, in practice, has two fundamental difficulties:

1- There is the practical problem of making large quantities of random keys. Any heavily used

2- Even more daunting is the problem of key distribution and protection. For every message to be sent, a key of equal length is needed by both sender and receiver.

Because of these difficulties, the one-time pad is of limited utility, and is useful primarily for low bandwidth channels requiring very high security.

Transposition Techniques:

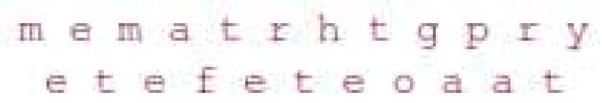
This technique is achieved by performing some sort of permutation on the plaintext letters.

1-Rail fence technique:

The plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.

Example:

To encipher the message "meet me after the toga party" with a rail fence of depth 2, we write the following:



```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

The encrypted message is:

MEMATRHTGPRYETEFETEOAAT

For depth 3 we write:

```
m       t       a       e       o      p      t
   e       m       f       r       g      a      y
     e       e       t       t       a       r
```

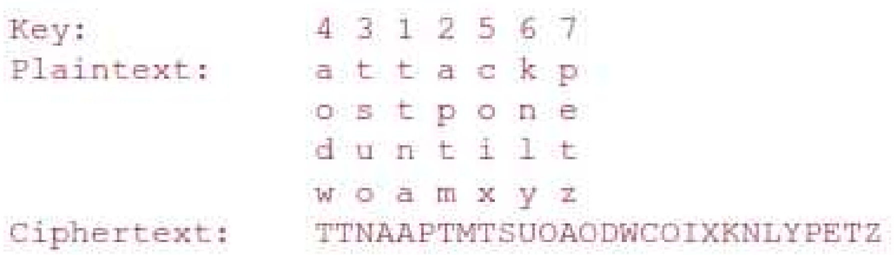
The encryption message is :

MTAEOPTEMFRGAYEETTAR

2- Columnar transposition :

is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm.

Example :

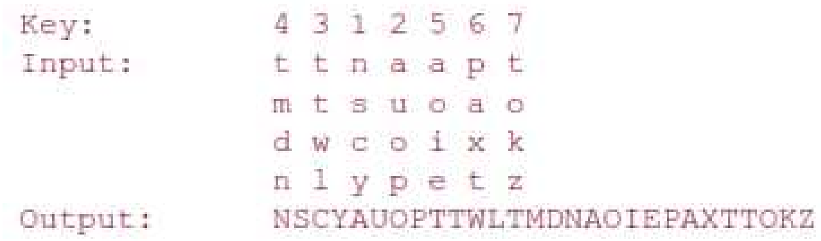


```
Key:           4 3 1 2 5 6 7
Plaintext:     a t t a c k p
               o s t p o n e
               d u n t i l t
               w o a m x y z
Ciphertext:    TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

3- Double columnar transposition:

The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is a more complex permutation that is not easily reconstructed. Thus, if the foregoing message is re-encrypted using the same algorithm.

Example :



```
Key:          4 3 1 2 5 6 7
Input:        t t n a a p t
              m t s u o a o
              d w c o i x k
              n l y p e t z
Output:       NSCYAUOPTTWLTMDNAOIEPAXTTOKZ
```

Block Cipher Principles

A **block cipher** is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.

- Many block ciphers have a Feistel structure. Such a structure consists of a number of identical rounds of processing. In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round.

Most symmetric block encryption algorithms in current use are based on a structure referred to as a Feistel block cipher.

Stream Ciphers and Block Ciphers

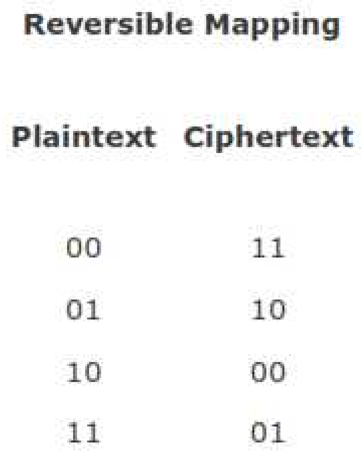
A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the autokeyed Vigenère cipher and the Vernam cipher.

A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Typically, a block size of 64 or 128 bits is used. A block cipher can be used to achieve the same effect as a stream cipher.

The idea of block cipher :

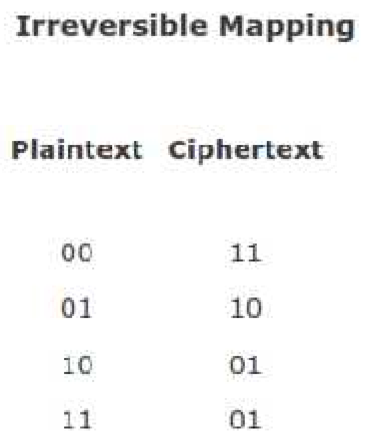
A block cipher operates on a plaintext block of ***n* bits** to produce a ciphertext block of ***n* bits**.

There are 2^n possible different plaintext blocks and, for the encryption to be reversible (i.e., for decryption to be possible), each must produce a unique ciphertext block. Such a transformation is called reversible, or nonsingular. The following examples illustrate nonsingular and singular transformation for *n* = 2.



Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01



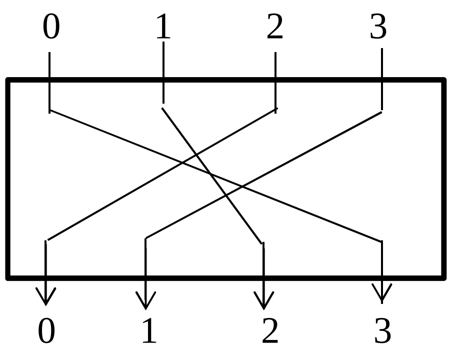
Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

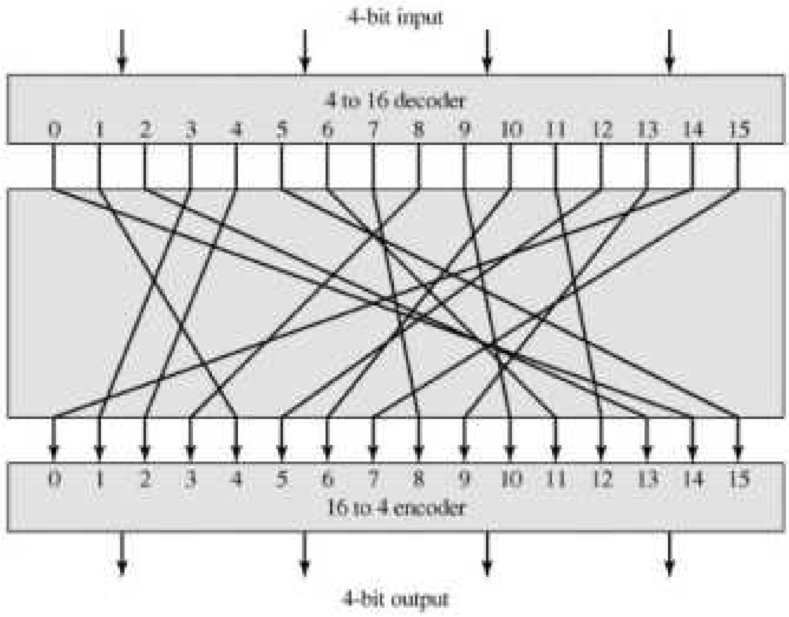
In the latter case, a ciphertext of 01 could have been produced by one of two plaintext blocks.

So if we limit ourselves to reversible mappings, the number of different transformations is $2^n!$.

The circuit diagram of the table (Reversible) above is shown below (n=2)



While with n=4, the circuit diagram (Reversible) is shown below:



General *n*-bit-*n*-bit Block Substitution (shown with *n* = 4)

The key is used here to determine the specific mapping from among all possible mapping (show the value of the ciphertext for each plaintext block).

For n=2 the key length is :

Key length = (2bits) x (4-row)= 8 bits.

Generally , for an n-bit ideal block cipher the key length is (n x 2^{n}).

If n is large ,the key will be very large ,as example:

If n= 64 ⟹ K= 64 x $2^{64} = 2^{70} \approx 10^{21}$bits.