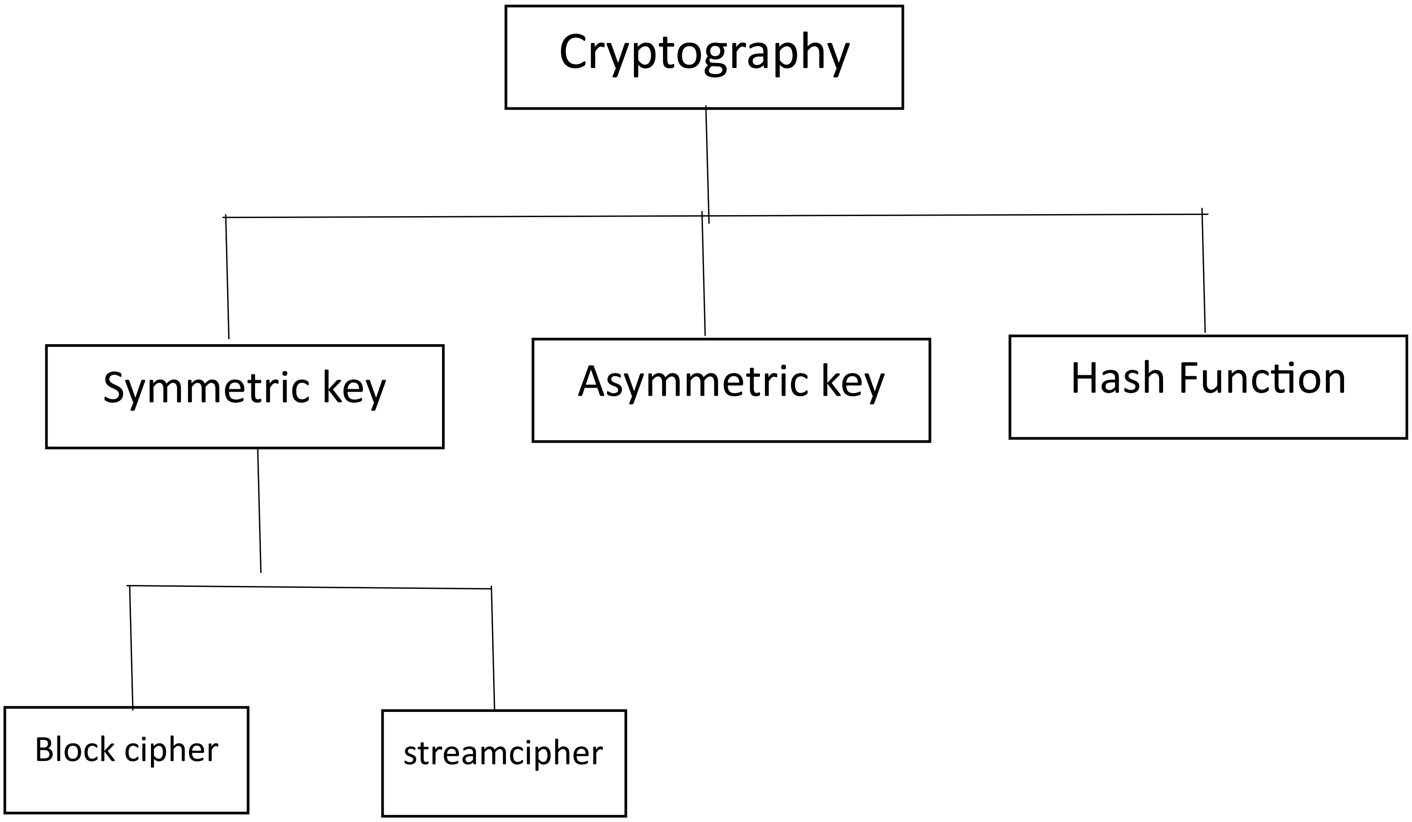
UNIT-II

Introduction to Modern Symmetric Key Ciphers, classical cryptography shift cipher, mono alphabetic substitution cipher, poly alphabetic substitution cipher.



Cryptography

Symmetric key

Asymmetric key

Hash Function

Block cipher

streamcipher

Symmetric cipher model

Symmetric Encryption is the most basic and old method of encryption. It uses only one key for the process of both the encryption and decryption of data. Thus, it is also known as Single-Key Encryption.

Before going into details, we need to know some of the terms,

Plain text- An original message to be communicated between the sender and the receiver is known as plain text.
Cipher text-The coded format of the original message that cannot be understood by the other third party is known as cipher text.
Encryption (or enciphering)-The process of converting plain text into the cipher text is known as encryption.
Decryption (or deciphering)-The process of converting cipher text into plain text is known as decryption.
Cryptography-the study of encryption is known as cryptography.
Cryptoanalysis-The techniques used for deciphering a message without any knowledge of the enciphering text is known as cryptanalysis.

There are two different kinds of cryptanalysis approaches-

- Brute-Force attack: The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On an average, half of all possible keys must be tried to achieve success.
- Cryptanalytic attack: cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some samples plaintext-ciphertext pairs.

Cryptology-The areas of cryptography and cryptoanalysis together called cryptology.

Symmetric cipher model

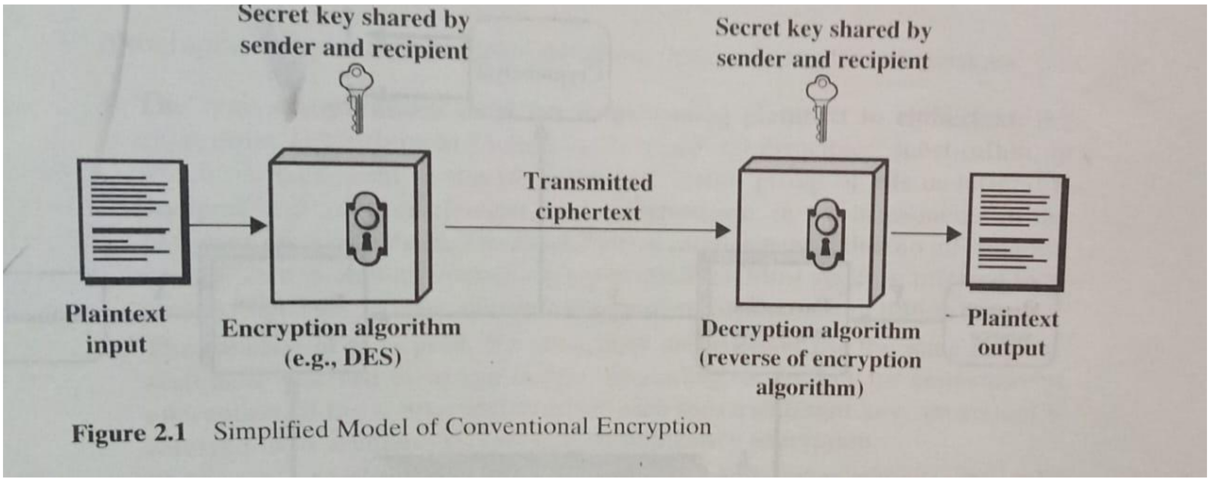

Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Figure 2.1 Simplified Model of Conventional Encryption

A symmetric cipher model is composed of five ingredients (Figure 2.1)-

- **Plain text:** This is the original message or data that is to be communicated to the receiver by the sender. It is fed into the algorithm as input.
- **Encrypting algorithm:** It takes the plain text and the secret key as inputs and produces Cipher Text as output. It performs various substitution and transformations on the plaintext using the secret key.
- **Secret Key:** The secret key is also an input into the algorithm. It is a value/string/text file used by the encryption and decryption algorithm to encode and decode the plain text to cipher text and vice-versa respectively. It is independent of the encryption algorithm.
- **Cipher text:** This is the scrambled message produced as output. It depends on the plain text and secret key. Each unique secret key produces a unique cipher text.
- **Decrypting algorithm:** This is essentially the encrypting algorithm run in reverse. It takes the cipher text and secret key as input and produces the original plain text as output.

Ther are two requirements for secure use of encryption-

1. **Encryption algorithm**-There is a need for a very strong encryption algorithm that produces cipher texts in such a way that the attacker should be unable to crack the secret key even if they have access to one or more cipher texts.
2. **Secure way to share secret key**-Sender and receiver must have obtained copies of the secret key in a secure way and keep the key secure. If someone can discover the key and knowns the algorithm, all communication using the key is readable.

Let us take a closer look at the essential elements of a symmetric encryption scheme, using Figure 2.2. A source produces a message in plaintext, X= [X1, X2…., XM]. The M elements of X are letters in some finite alphabet. Traditionally, the alphabet usually consisted of the 26 capital letters.

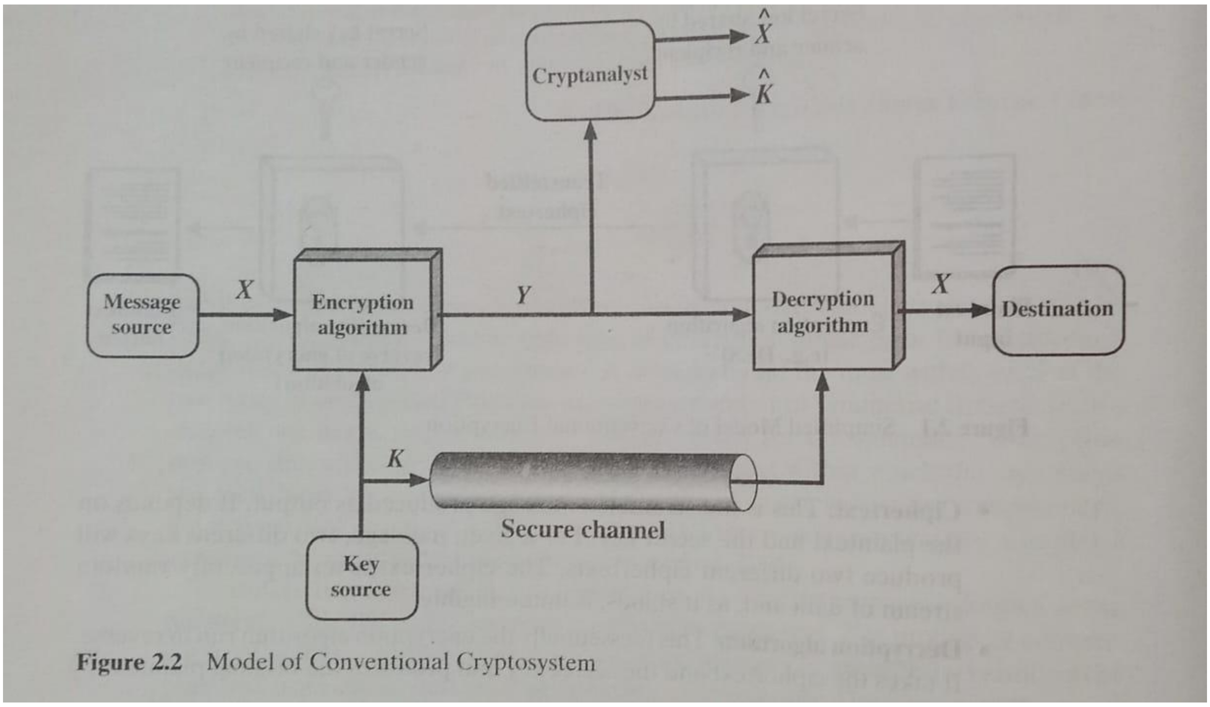


Figure 2.2 Model of Conventional Cryptosystem

For encryption, a key of the form K = [K1, K2….., KJ] is generated. If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel. Alternatively, a third party could generate the key and securely deliver it to both source and destination.

With the message X and the encryption key K as input, the encryption algorithm forms the ciphertext Y = [Y1, Y2….., YN]. We can write this as

Y = EK(X)

This notation indicates that Y is produced by using encryption algorithm E as a function of the plaintext X, with the specific function determined by the value of the key K.

The intended receiver, in possession of the key, is able to invert the transformation:

$$X = D(Y)$$

An opponent, observing Y but not having access to K or X, may attempt to recover X or K or both X and K. It is assumed that the opponent knows the encryption (E) and decryption (D) algorithms. If the opponent is interested in only this particular message, then the focus of the effort is to recover X by generating a plaintext estimate X^. Often, however, the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover K by generating an estimate K^.

Classic ciphers

- Substitution cipher techniques
  - Ceaser cipher
  - Mono alphabetic cipher
  - Playfair cipher
  - Poly alphabetic cipher
  - One-time padding
- Transposition cipher techniques
  - Rail fence cipher
  - Columnar transposition cipher
- Product cipher techniques

Substitution cipher

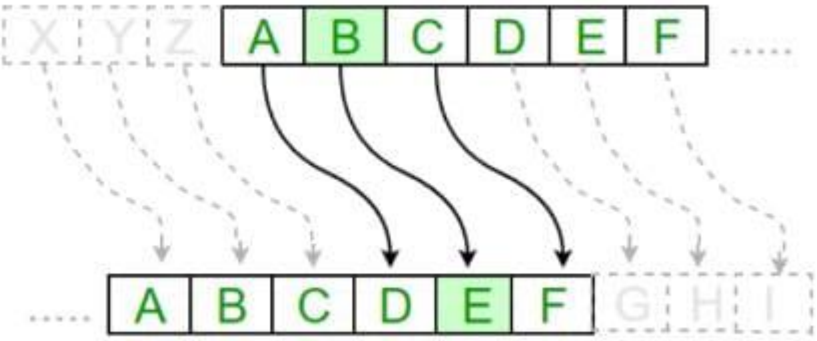
In a Substitution cipher, any character of plain text from the given fixed set of characters is substituted by some other character from the same set depending on a key. For example, with a shift of 1, A would be replaced by B, B would become C, and so on.

Ceaser cipher:

The earliest known use of a substitution cipher, and the simplest, way by Julius Caesar. It works by shifting the letters in the plaintext message by a certain number of positions, known as the “shift” or “key”.

The Caesar Cipher technique is one of the earliest and simplest methods of encryption technique. It’s simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet. Thus, to cipher a given text we need an integer value, known as a shift which indicates the number of positions each letter of the text has been moved down. The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1…, Z = 25.

For example, if the shift is 3, then the letter A would be replaced by the letter D, B would become E, C would become F, and so on. The alphabet is wrapped around so that after Z, it starts back at A.



Let us consider the following example,

Plain: meet me after the party

Cipher: PHHW PH DIWHU WKH SDUWB

We can define the transformation by listing all possibilities, as follows:

plain: 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

cipher: 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

Let us assign a numerical equivalent to each letter,

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

Then the algorithm expressed as follows. For each plain text letter p substitute the cyphertext C.

$$C = E(p) = (p+3) \bmod (26)$$

A shift may be any amount, so that the general Caeser algorithm is

$$C = E(p) = (p+k) \bmod (26)$$

Where k takes on a value in the range 1 to 25. The decryption algorithm is simply

$$P = D(C) = (C-k) \bmod (26)$$

If it is known that a given cipher text is a Caeser cipher, then a brute-force cryptanalysis is easily performed by simply trying all the 25 possible keys.

	PHHW	PH	DIWHU	WKH	SDUWB
KEY					
1	oggv	og	chvgt	vjg	rctva
2	nffu	nf	bgufs	uif	qbsuz
3	meet	me	after	the	party
4	ldds	ld	zesdq	sgd	ozqsx
5	kccr	kc	ydrcp	rfc	nyprw
6	jbbq	jb	xcqbo	qeb	mxoqv
7	iaap	ia	wbpan	pda	lwnpu
8	hzzo	hz	vaozm	ocz	kvmot
9	gyyn	gy	uznyl	nby	julns
10	fxxm	fx	tymxk	max	itkmr
11	ewwl	ew	sxlwj	lzw	hsjlq
12	dvvk	dv	rwkvi	kyv	grikp
13	cuuj	cu	qvjuh	jxu	fqhjo
14	btti	bt	puitg	iwt	epgin
15	assh	as	othsf	hvs	dofhm
16	zrrg	zr	nsgre	gur	cnegl
17	yqqf	yq	mrfqd	ftq	bmdfk
18	xppe	xp	lqepc	esp	alcej
19	wood	wo	kpdob	dro	zkbdi
20	vnnc	vn	jocna	cqn	yjach
21	ummb	um	inbmz	bpm	xizbg
22	tlla	tl	hmaly	aol	whyaf
23	skkz	sk	glzkx	znk	vgxze
24	rjjy	rj	fkyjy	ymj	ufwyd
25	qiixzrrg	qi	ejxiv	xli	tevxc

Three important characteristics of this problem enabled to use a brute-force attack cryptanalysis:

1. The encryption and decryption algorithms are known.
2. There are only 25 keys to try.
3. The language of the plaintext is known and easily recognizable.

Program code:

```
//This function receives text and shift and
// returns the encrypted text
string encrypt (string text, int s)
{
    string result = "";

    // traverse text
    for (int i = 0; i < text.length(); i++) {
        // apply transformation to each character
        // Encrypt Uppercase letters
        if (isupper(text[i]))
            result += char(int(text[i] + s - 65) % 26 + 65);

        // Encrypt Lowercase letters
        else
            result += char(int(text[i] + s - 97) % 26 + 97);
    }

    // Return the resulting string
    return result;
}
```

Examples:

```
Text : ABCDEFGHIJKLMNOPQRSTUVWXYZ
Shift: 23
Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW

Text : ATTACKATONCE
Shift: 4
Cipher: EXXEGOEXSRGI
```

Advantages:

- Easy to implement and use thus, making suitable for beginners to learn about encryption.
- Can be physically implemented, such as with a set of rotating disks or a set of cards, known as a scytale, which can be useful in certain situations.
- Requires only a small set of pre-shared information.
- Can be modified easily to create a more secure variant, such as by using a multiple shift values or keywords.

Disadvantages:

- It is not secure against modern decryption methods.
- Vulnerable to known-plaintext attacks, where an attacker has access to both the encrypted and unencrypted versions of the same messages.
- The small number of possible keys means that an attacker can easily try all possible keys until the correct one is found, making it vulnerable to a brute force attack.
- It is not suitable for long text encryption as it would be easy to crack.
- It is not suitable for secure communication as it is easily broken.
- Does not provide confidentiality, integrity, and authenticity in a message.

Mono alphabetic cipher:

With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution.

If, instead, the cipher line can be any permutations of the 26 alphabetic characters, then there are 26! Or greater than 4 x 10^26 possible keys.

In mono-alphabetic ciphers, each symbol in plain-text (e.g.; ‘o’ in ‘follow’) is mapped to one cipher-text symbol. No matter how many times a symbol occurs in the plain-text, it will correspond to the same cipher-text symbol. For example, if the plain-text is ‘follow’ and the mapping is:

- f -> g
- o -> p
- l -> m
- w -> x

The cipher-text is ‘gpmmpx’.

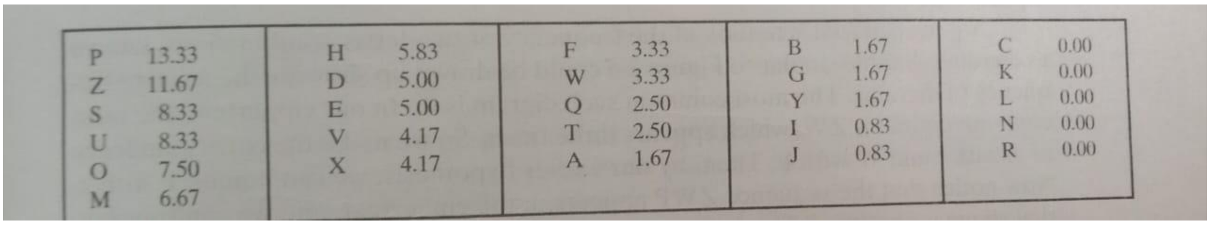
There is, however, another line of attack. If the cryptanalyst knows the nature of the plain text, then the analyst can exploit the regularities of the language.

Let us consider the following example,

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDEMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX

EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

As a first step, the relative frequency of the letters can be determined and compared to a standard frequency distribution for English. If the message were long enough, this technique alone might be sufficient, but because this is a relatively short message, we cannot expect an exact match. In any case, the relative frequencies of the letters in the ciphertext (in percentages) are as follows.



P	13.33	H	5.83	F	3.33	B	1.67	C	0.00
Z	11.67	D	5.00	W	3.33	G	1.67	K	0.00
S	8.33	E	5.00	Q	2.50	Y	1.67	L	0.00
U	8.33	V	4.17	T	2.50	I	0.83	N	0.00
O	7.50	X	4.17	A	1.67	J	0.83	R	0.00
M	6.67								

Comparing this breakdown with Figure 2.5, it seems likely that cipher letters P and Z are the equivalents of plain letters e and t. but it is not certain which is which. The letters S, U, O, M, and H are all of relatively high frequency and probably correspond to plain letters from the set [a, h, i, n, o, r. s). The letters with the lowest frequencies (namely, A, B, G, Y, I, J) are likely included in the set {b, j, k, q, v, x, z}.

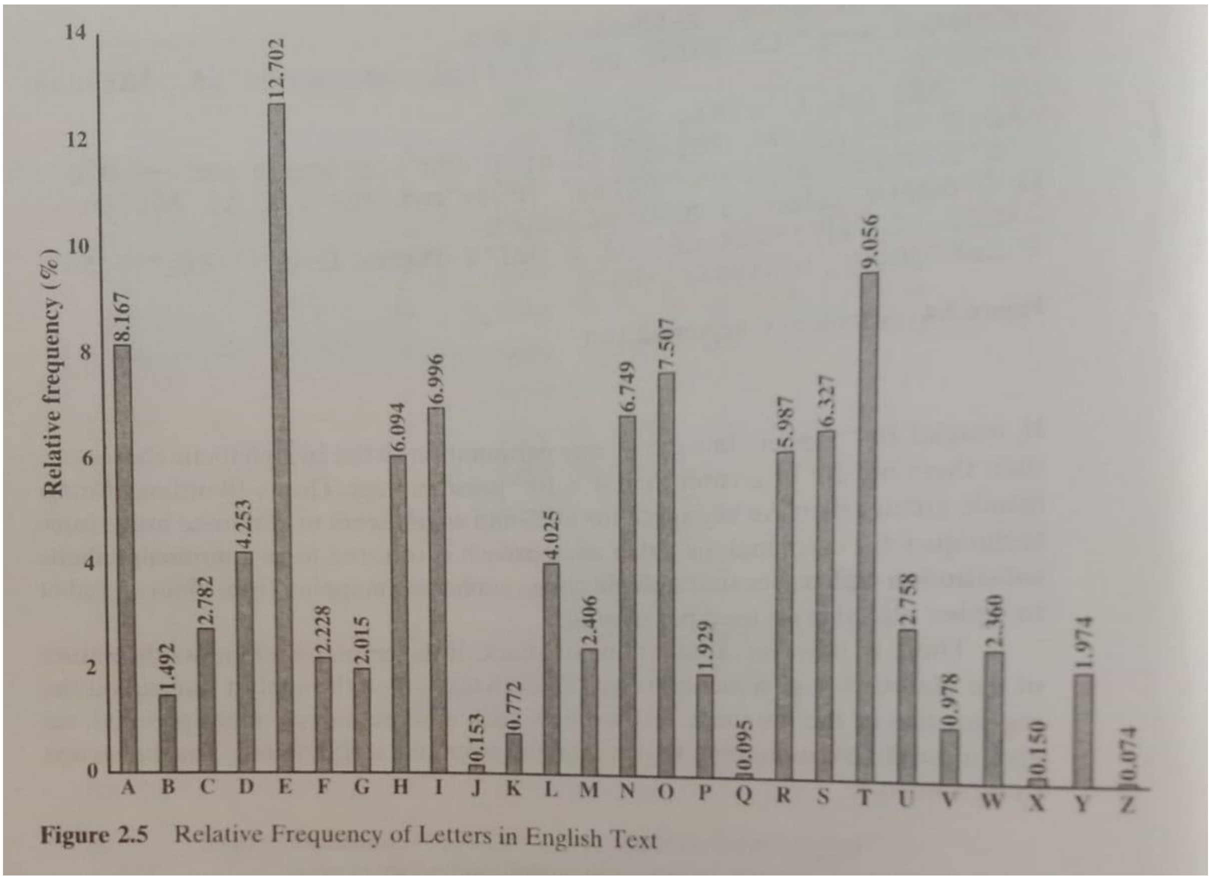


Figure 2.5 Relative Frequency of Letters in English Text

So far, then, we have

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSKAIZ
 t a        e  e te  a that e e a       a
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
   e t   ta t ha e ee  a e  th    t  a
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
 e  e e tat  e   the   t
```

Only four letters have been identified, but already we have quite a bit of the message. Continued analysis of frequencies plus trial and error should easily yield a solution from this point. The complete plaintext, with spaces added between words, follows:

it was disclosed yesterday that several informal but
direct contacts have been made with political
representatives of the viet cong in Moscow

Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet. A countermeasure is to provide multiple substitutes, known as homophones, for a single letter.

Playfair cipher:

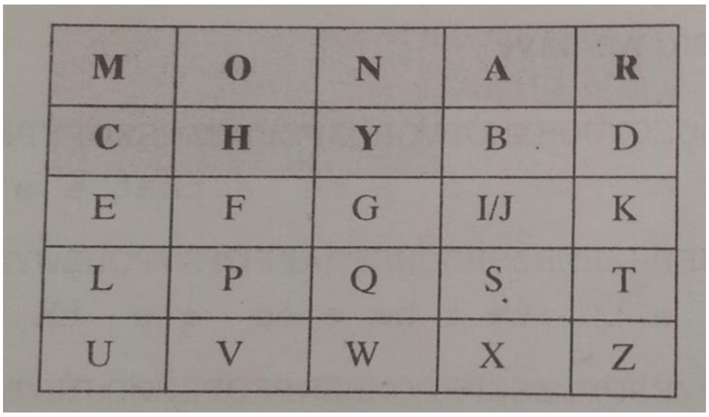
The best-known multiple-letter encryption cipher is the playfair. The **Playfair cipher** was the first practical digraph substitution cipher. The scheme was invented in **1854** by **Charles Wheatstone** but was named after Lord Playfair who promoted the use of the cipher.

The playfair algorithm is based on the use of a 5 x 5 matrix of letters constructed using a keyword.

For the encryption process let us consider the following example:

Key: monarchy

Plain text: instruments



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

Encryption Algorithm:

The Algorithm consists of 2 steps:

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. 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). If the plaintext contains J, then it is replaced by I.

    - 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.

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:

```
Plaintext: "instruments"
After Split: 'in' 'st' 'ru' 'me' 'nt' 'sz'
```

1. Pair cannot be made with same letter. Break the letter in single and add a bogus letter to the previous letter.

Plain Text: “hello”
After Split: ‘he’ ‘lx’ ‘lo’

Here **‘x’** is the bogus letter.

2. If the letter is standing alone in the process of pairing, then add an extra bogus letter with the alone letter

Plain Text: “helloe”
After Split: ‘he’ ‘lx’ ‘lo’ ‘ez’

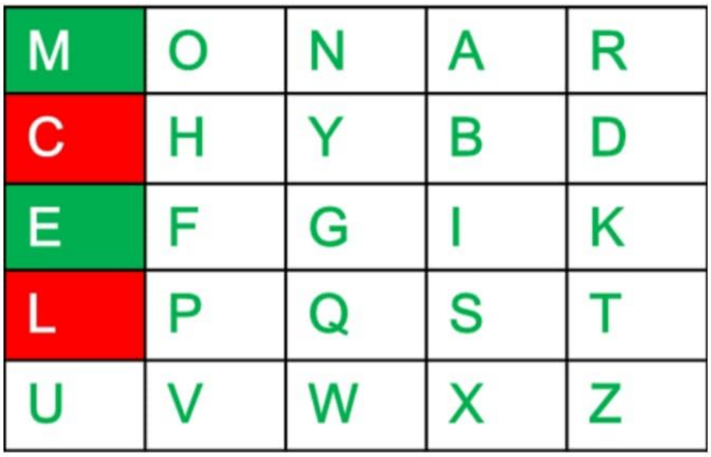
Here **‘z’** is the bogus letter.

Rules for encryption:

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

For example:

```
Diagraph: "me"
Encrypted Text: cl
Encryption:
  m -> c
  e -> l
```

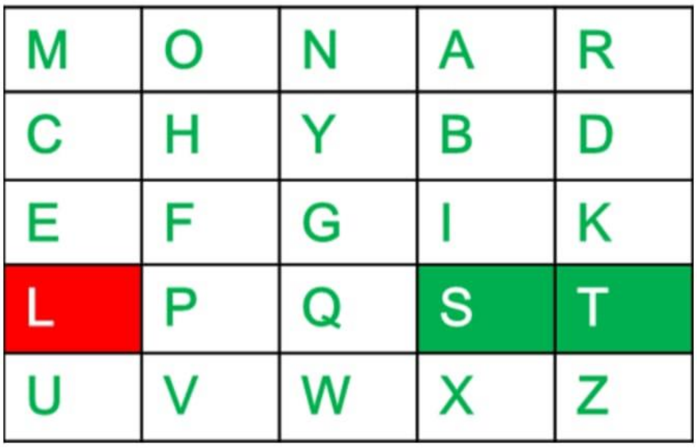


M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

- **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).

For example:

```
Diagraph: "st"
Encrypted Text: tl
Encryption:
  s -> t
  t -> l
```

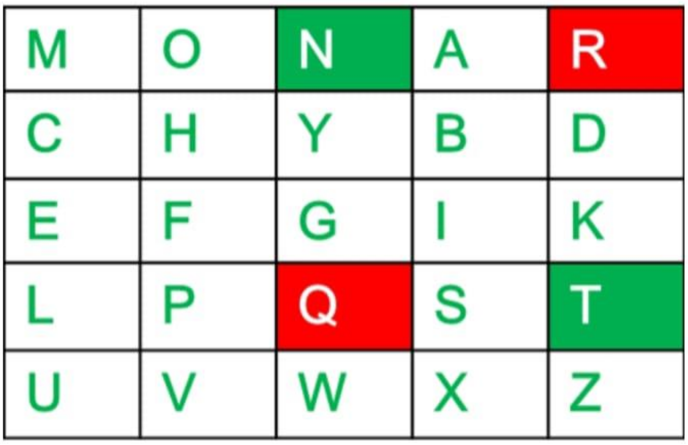


M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

- **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.

For example:

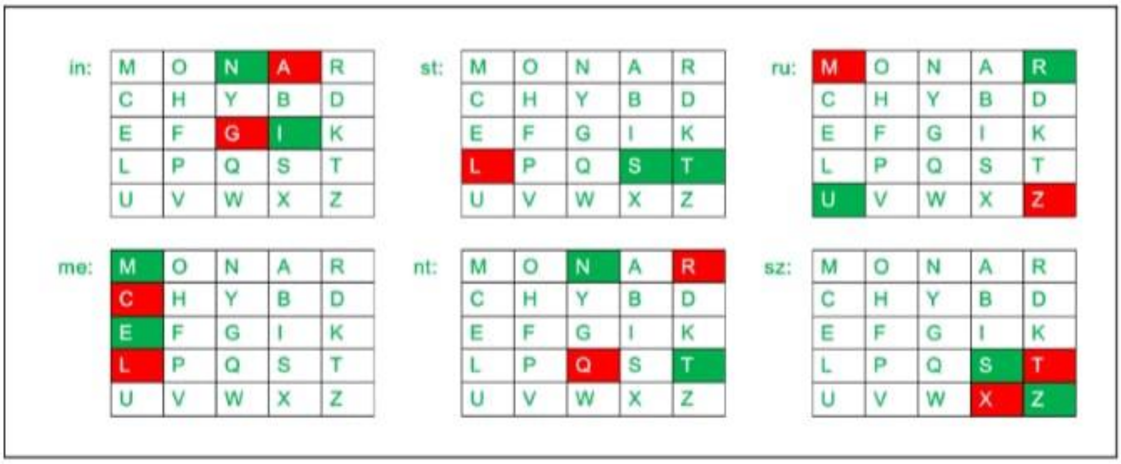
```
Diagraph: "nt"
Encrypted Text: rq
Encryption:
  n -> r
  t -> q
```



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

For 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

Decryption Algorithm:

The Algorithm consists of 2 steps:

1. **Generate the key Square (5×5) at the receiver's end:**
   - The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. 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). If the plaintext contains J, then it is replaced by I.

   - 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.

2. **Algorithm to decrypt the ciphertext:** The ciphertext is split into pairs of two letters (digraphs).

*Note: The **ciphertext** always have **even** number of characters.*

For example:

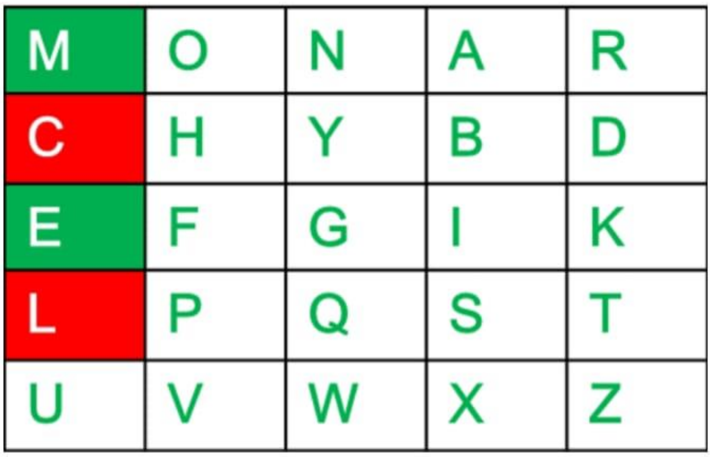
```
Cipher Text: "gatlmzclrqtx"
After Split: 'ga' 'tl' 'mz' 'cl' 'rq' 'tx'
```

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).

For example:

```
Diagraph: "cl"
Decrypted Text: me
Decryption:
  c -> m
  l -> e
```

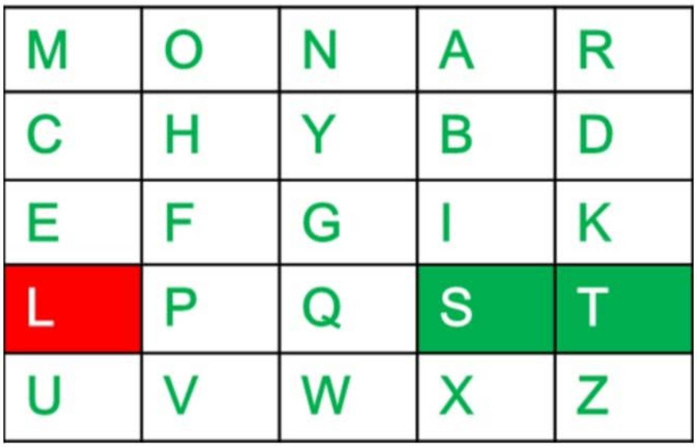


M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

- **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)

For example:

```
Diagraph: "tl"
Decrypted Text: st
Decryption:
  t -> s
  l -> t
```

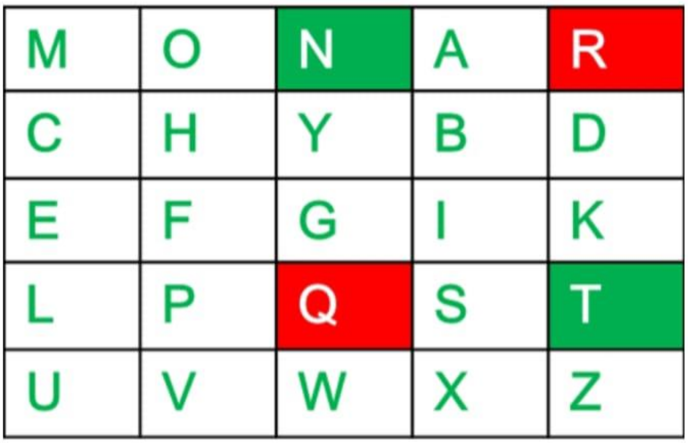


M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

- **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.

For example:

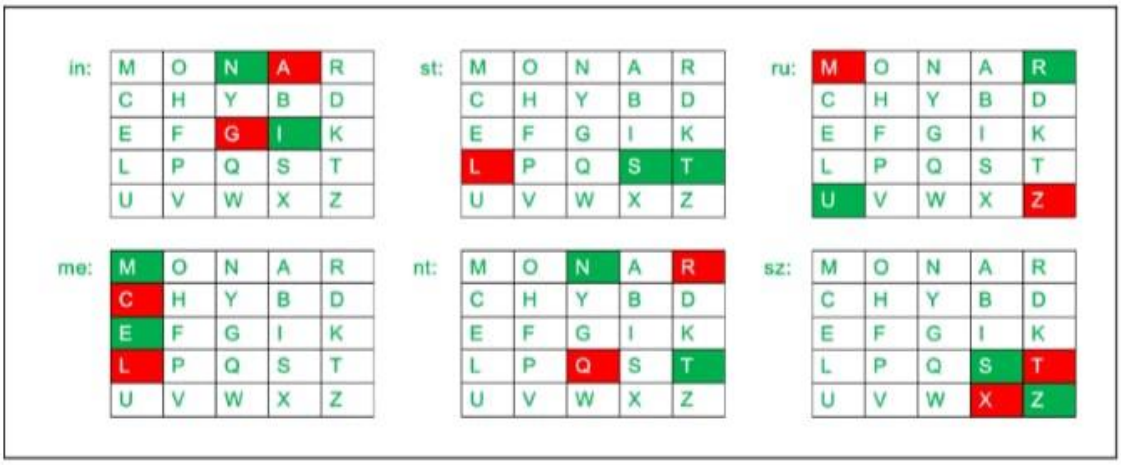
```
Diagraph: "rq"
Decrypted Text: nt
Decryption:
  r -> n
  q -> t
```



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

For example:

```
Plain Text: "gatlmzclrqtx"
Decrypted Text: instrumentsz
Decryption:
(red)-> (green)
  ga -> in
  tl -> st
  mz -> ru
  cl -> me
  rq -> nt
  tx -> sz
```



Advantages:

1. It is significantly harder to break since the frequency analysis technique used to break simple substitution ciphers is difficult but still can be used on (25*25) = 625 digraphs rather than 25 monographs which is difficult.
2. Frequency analysis thus requires more cipher text to crack the encryption.

Disadvantages:

1. An interesting weakness is the fact that a digraph in the ciphertext (AB) and it’s reverse (BA) will have corresponding plaintexts like UR and RU (and also ciphertext UR and RU will correspond to plaintext AB and BA, i.e. the substitution is self-inverse). That can easily be exploited with the aid of frequency analysis, if the language of the plaintext is known.
2. Another disadvantage is that play fair cipher is a symmetric cipher thus same key is used for both encryption and decryption.

Polyalphabetic cipher:

Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plain text message. The general name for this approach is polyalphabetic substitution cipher All these techniques have the following features in common.
1. A set of related monoalphabetic substitution rules is used
2. A key determines which particular rule is chosen for a given transformation

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.
To encrypt a message, a key is needed that is as long as the message. Usually. the key is a repeating keyword. For example, if the keyword is deceptive, the message "we are discovered save yourself" is encrypted as follows:

key: deceptivedeceptivedeceptive
plaintext: wearediscoveredsaveyourself
ciphertext: Z1CVTWONGRZGVTWAVZHCQYGLMGJ

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

The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword. Thus, the letter frequency information is obscured. However, not all knowledge of the plaintext structure is lost.

One-Time padding:

One-time pad cipher is a type of Vigenère cipher which includes the following features −

- It is an unbreakable cipher.
- The key is exactly same as the length of message which is encrypted.
- The key is made up of random symbols.
- As the name suggests, key is used one time only and never used again for any other message to be encrypted.

Due to this, encrypted message will be vulnerable to attack for a cryptanalyst. The key used for a one-time pad cipher is called **pad**, as it is printed on pads of paper.

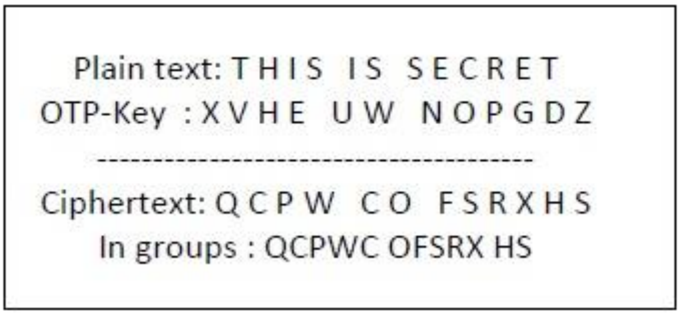
Why is it Unbreakable?

The key is unbreakable owing to the following features −

- The key is as long as the given message.
- The key is truly random and specially auto-generated.
- Key and plain text calculated as modulo 10/26/2.
- Each key should be used once and destroyed by both sender and receiver.
- There should be two copies of key: one with the sender and other with the receiver.

Encryption:

To encrypt a letter, a user needs to write a key underneath the plaintext. The plaintext letter is placed on the top and the key letter on the left. The cross section achieved between two letters is the plain text. It is described in the example below



```
  Plain text: T H I S   I S   S E C R E T
 OTP-Key  : X V H E   U W   N O P G D Z
    --------------------------------------
 Ciphertext: Q C P W   C O   F S R X H S
    In groups : QCPWC OFSRX HS
```

Decryption:

To decrypt a letter, user takes the key letter on the left and finds cipher text letter in that row. The plain text letter is placed at the top of the column where the user can find the cipher text letter.

Transposition cipher

Transposition technique is an encryption method which is achieved by performing **permutation over the plain text**. Mapping plain text into cipher text using transposition technique is called **transposition cipher**.

Rail-fence cipher:

Given a plain-text message and a numeric key, cipher/de-cipher the given text using Rail Fence algorithm.

The rail fence cipher (also called a zigzag cipher) is a form of transposition cipher. It derives its name from the way in which it is encoded.

Examples:

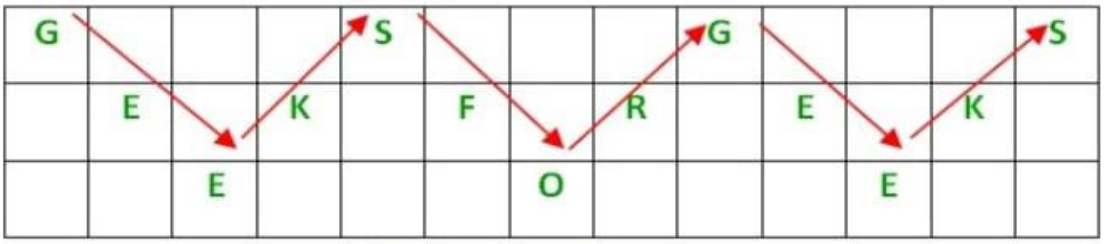
```
Encryption
Input :  "GeeksforGeeks "
Key = 3
Output : GsGsekfrek eoe
Decryption
Input : GsGsekfrek eoe
Key = 3
Output :  "GeeksforGeeks "

Encryption
Input :  "defend the east wall"
Key = 3
Output : dnhaweedtees alf  tl
Decryption
Input : dnhaweedtees alf  tl
Key = 3
Output : defend the east wall

Encryption
Input : "attack at once"
Key = 2
Output : atc toctaka ne
Decryption
Input : "atc toctaka ne"
Key = 2
Output : attack at once
```

Encryption:

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

- In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.
- When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus, the alphabets of the message are written in a zig-zag manner.
- After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

For example, if the message is “GeeksforGeeks” and the number of rails = 3 then cipher is prepared as:



G				S				G				S
	E		K		F		R		E		K	
		E				O				E		

Decryption:

As we’ve seen earlier, the number of columns in rail fence cipher remains equal to the length of plain-text message. And the key corresponds to the number of rails.

- Hence, rail matrix can be constructed accordingly. Once we’ve got the matrix we can figure-out the spots where texts should be placed (using the same way of moving diagonally up and down alternatively).
- Then, we fill the cipher-text row wise. After filling it, we traverse the matrix in zig-zag manner to obtain the original text.

Columnar -Transposition cipher:

Given a plain-text message and a numeric key, cipher/de-cipher the given text using Columnar Transposition Cipher. The Columnar Transposition Cipher is a form of transposition cipher just like Rail fence cipher, Columnar Transposition involves writing the plaintext out in rows, and then reading the ciphertext off in columns one by one.

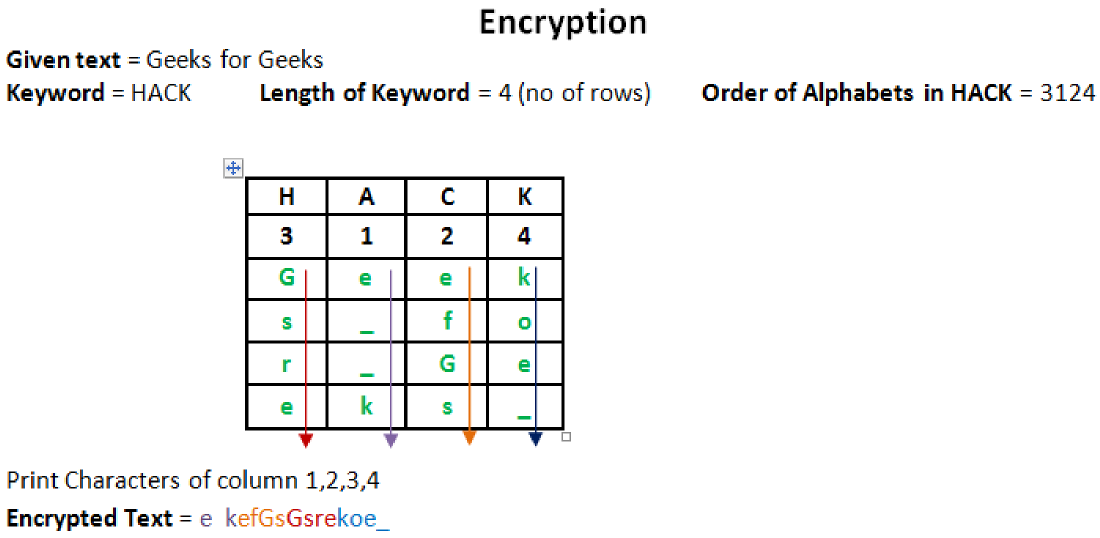
Examples:

```
Encryption
Input : Geeks for Geeks
Key = HACK
Output : e  kefGsGsrekoe_
Decryption
Input : e  kefGsGsrekoe_
Key = HACK
Output : Geeks for Geeks
Encryption
Input :  Geeks on work
Key = HACK
Output : e w_eoo_Gs kknr_
Decryption
Input : e w_eoo_Gs kknr_
Key = HACK
Output : Geeks on work
```

Encryption:

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

1. The message is written out in rows of a fixed length, and then read out again column by column, and the columns are chosen in some scrambled order.
2. Width of the rows and the permutation of the columns are usually defined by a keyword.
3. For example, the word HACK is of length 4 (so the rows are of length 4), and the permutation is defined by the alphabetical order of the letters in the keyword. In this case, the order would be “3 1 2 4”.
4. Any spare spaces are filled with nulls or left blank or placed by a character (Example: _).
5. Finally, the message is read off in columns, in the order specified by the keyword.



Encryption

Given text = Geeks for Geeks

Keyword = HACK **Length of Keyword** = 4 (no of rows) **Order of Alphabets in HACK** = 3124

H	A	C	K
3	1	2	4
G	e	e	k
s	_	f	o
r	_	G	e
e	k	s	_

Print Characters of column 1,2,3,4

Encrypted Text = e kefGsGsrekoe_

Decryption:

1. To decipher it, the recipient has to work out the column lengths by dividing the message length by the key length.
2. Then, write the message out in columns again, then re-order the columns by reforming the key word.