2- Running Key Ciphers

In order to make the key size approach infinity, the Tx selects a key (K) with the same size of the message. This key is usually selected from say your library by specifying a **book number** plus a **page number** plus a **line number**. If both the Tx and Rx agree on that protocol, both will read the same sentence from that book on that page starting from that line number. This sentence will represent the key K. Then:

Encryption:

$C_i = (P_i + K_i) \bmod n$ **...........(17)**

Decryption:

$P_i = (C_i - K_i) \bmod n$ **............(18)** , where **n** = alphabet size = 26 for English language.

P_i = a number corresponding to the plaintext character. C_i = a number corresponding to the Ciphertext character. K_i = a number corresponding to the key character.

Above P_i, C_i and K_i number are obtained by numbering the characters from say A to Z as 0 to 25, i.e.

P_i (C_i or K_i)	A	B	C	D	E	.	.	.	Z
Number	0	1	2	3	4	.	.	.	25

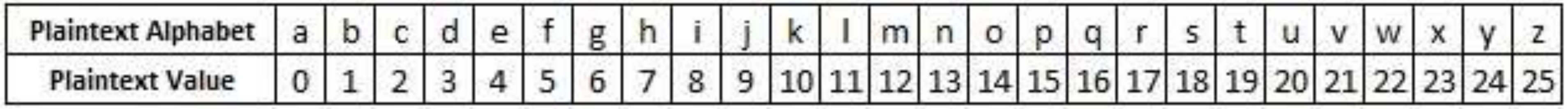
The key size here is almost **infinite** or **very large** since there is no limit to the size of the number of books available at your library.

In classical cryptography, the running key cipher is a type of polyalphabetic substitution cipher in which a text, typically from a book, is used to provide a very long keystream. Usually, the book to be used would be agreed upon ahead of time, while the passage to be used would be chosen randomly for each message and secretly indicated somewhere in the message.

Example:

Suppose we have agreed to use The C Programming Language (1978 edition), as our book to select the key, and we are using the **tabula recta** also called a **Vigenere square**, **Vigenere table**, or **tabula recta** or **tableau**). The plaintext is "Flee at once". Page 63, line 1 is selected as the running key: **errors can occur in several places**.

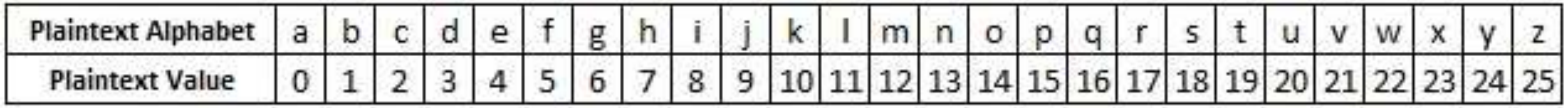
A label that has the running key is then written under the plaintext:



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

Plaintext	**f**	**l**	**e**	**e**	**a**	**t**	**o**	**n**	**c**	**e**
Running key	**e**	**r**	**r**	**o**	**r**	**s**	**c**	**a**	**n**	**o**
Ciphertext	**J**	**C**	**V**	**S**	**R**	**L**	**Q**	**N**	**P**	**S**

The message is then sent as **"JCVSR LQNPS".** However, unlike a Vigenere cipher, if the message is extended**,** **the key is not repeated**; the key text itself is used as the key. If the message is extended, such as, "Flee at once. We are discovered", then the running key continues as before:



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

P	**f**	**l**	**e**	**e**	**a**	**t**	**o**	**n**	**c**	**e**	**w**	**e**	**a**	**r**	**e**	**d**	**i**	**s**	**c**	**o**	**v**	**e**	**r**	**e**	**d**
K	**e**	**r**	**r**	**o**	**r**	**s**	**c**	**a**	**n**	**o**	**c**	**c**	**u**	**r**	**i**	**n**	**s**	**e**	**v**	**e**	**r**	**a**	**l**	**p**	**l**
C	**J**	**C**	**V**	**S**	**R**	**L**	**Q**	**N**	**P**	**S**	**Y**	**G**	**U**	**I**	**M**	**Q**	**A**	**W**	**X**	**S**	**M**	**E**	**C**	**T**	**O**

To determine where to find the running key, a fake block of five Ciphertext characters is subsequently added, with **three** denoting the **page number**, and **two** the **line number**, using A=0, B=1, etc., to encode digits. Such a block is called an **indicator block**. The indicator block will be inserted as the second last of each message. (Many other schemes are possible for hiding indicator blocks.) Thus page 63, line 1 encodes as "AGDAB" (06301).

This yields a final message of "JCVSR LQNPS YGUIM QAWXS **AGDAB** MECTO".

3- Rotor Machine Cipher

- ❑ The most important application of the principle of multiple stages.
- ❑ Multiple stages of encryption can produce an algorithm that is significantly more difficult to Cryptanalyze.
- ❑ Before modern ciphers, rotor machines were the most common complex ciphers in use.

- ❑ Widely used in the Second World War (WW2).

German Enigma, **Allied Hagelin**, **Japanese Purple**

- ❑ Used a series of rotating cylinders.

- ❑ Implemented a polyalphabetic substitution cipher of period K.

With 3 cylinders, K = 26^3 =17,576.

With 5 cylinders, K = 26^5 =12 x 10^6.

The basic principle of the rotor machine

- The machine consists of a set of independently rotating cylinders through which electrical pulses can flow.

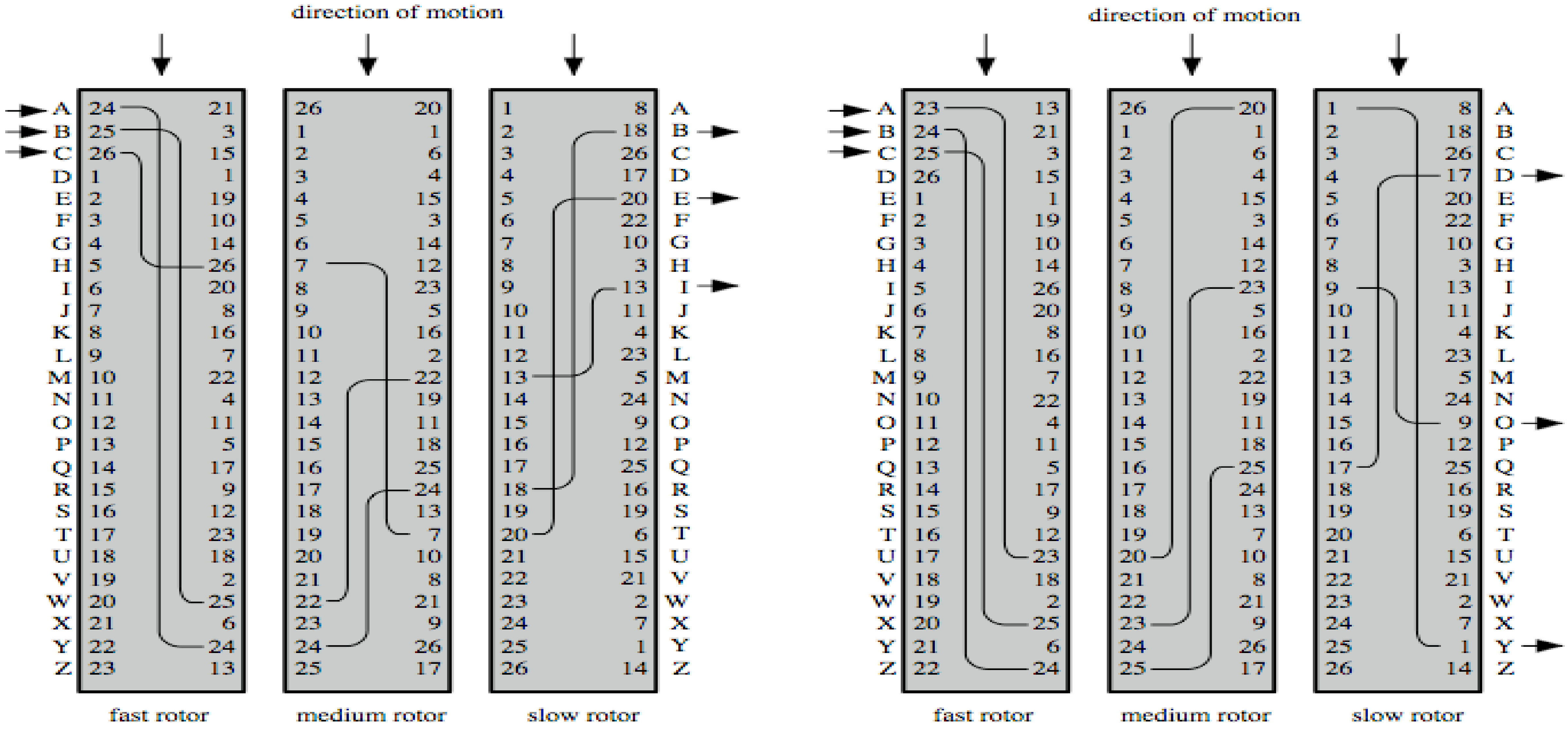
- Each cylinder has 26 input pins and 26 output pins, with internal wiring that connects each input pin to a unique output pin.

- If we associate each input and output pin with a letter of the alphabet, then a single cylinder defines a monoalphabetic substitution.

- The power of the rotor machine is in the use of multiple cylinders, in which the output pins of one cylinder are connected to the input pins of the next.

- For every complete rotation of the inner cylinder, the middle cylinder rotates one pin position. Finally, for every complete rotation of the middle cylinder, the outer cylinder rotates one pin position.
- This is the same type of operation seen with an odometer.

- The basic principle of the rotor machine is illustrated in Figure (2.4). The result is that there are 26 * 26 * 26 = 17,576 different substitution alphabets used before the system.



A	24	21	26	20	1	8	A
B	25	3	1	1	2	18	B
C	26	15	2	6	3	26	C
D	1	1	3	4	4	17	D
E	2	19	4	15	5	20	E
F	3	10	5	3	6	22	F
G	4	14	6	14	7	10	G
H	5	26	7	12	8	3	H
I	6	20	8	23	9	13	I
J	7	8	9	5	10	11	J
K	8	16	10	16	11	4	K
L	9	7	11	2	12	23	L
M	10	22	12	22	13	5	M
N	11	4	13	19	14	24	N
O	12	11	14	11	15	9	O
P	13	5	15	18	16	12	P
Q	14	17	16	25	17	25	Q
R	15	9	17	24	18	16	R
S	16	12	18	13	19	19	S
T	17	23	19	7	20	6	T
U	18	18	20	10	21	15	U
V	19	2	21	8	22	21	V
W	20	25	22	21	23	2	W
X	21	6	23	9	24	7	X
Y	22	24	24	26	25	1	Y
Z	23	13	25	17	26	14	Z

(a) Initial setting

A	23	13	26	20	1	8	A
B	24	21	1	1	2	18	B
C	25	3	2	6	3	26	C
D	26	15	3	4	4	17	D
E	1	1	4	15	5	20	E
F	2	19	5	3	6	22	F
G	3	10	6	14	7	10	G
H	4	14	7	12	8	3	H
I	5	26	8	23	9	13	I
J	6	20	9	5	10	11	J
K	7	8	10	16	11	4	K
L	8	16	11	2	12	23	L
M	9	7	12	22	13	5	M
N	10	22	13	19	14	24	N
O	11	4	14	11	15	9	O
P	12	11	15	18	16	12	P
Q	13	5	16	25	17	25	Q
R	14	17	17	24	18	16	R
S	15	9	18	13	19	19	S
T	16	12	19	7	20	6	T
U	17	23	20	10	21	15	U
V	18	18	21	8	22	21	V
W	19	2	22	21	23	2	W
X	20	25	23	9	24	7	X
Y	21	6	24	26	25	1	Y
Z	22	24	25	17	26	14	Z

(b) Setting after one keystroke

Figure (2.4): Three-Rotor Machine with Wiring Represented by Numbered Contacts

4- Transposition

The whole message is divided into blocks, each m-character long. Then the characters in each block are permutated according to a certain permutation law as shown below: i.e. the same characters are used but interchanging their positions.

Character no. in Plaintext	1	2	3	4	5	6	7	8	9	.	.	.	.	m
Character no. in Ciphertext	**5**	**3**	**10**	**1**	**8**	**2**	**16**	**4**	**11**					**12**

Discuss the no of available keys:

This will be **m!**, but here m can be very large even larger than the finite no of the alphabet of the language.

i.e., this will give say 100! Or 1000! **No. of available key**

No. of bits to assign a key will be $\log_2 m!$.

i. One Dimension Transposition (1D Transposition)

In transposition, the positions of the plaintext letters in the message rather than the letters of the alphabet are permuted.

Example: Encrypt the plaintext **P='rotational'** using 5-character block transposition cipher with key

$$\mathbf{k} = \begin{bmatrix} \mathbf{P} \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{3} & \mathbf{5} & \mathbf{2} & \mathbf{1} & \mathbf{4} \end{bmatrix}$$

Solution: The plaintext consists of two blocks each 5 character length. According to k, the first letter in C will be the 3rd in P and the 2nd letter in C is the 5th letter in P and so on, hence:

$$\begin{bmatrix} \mathbf{P} \\ \mathbf{C} \end{bmatrix} = \left[\begin{array}{ccccc|ccccc} \mathbf{r} & \mathbf{o} & \mathbf{t} & \mathbf{a} & \mathbf{t} & \mathbf{i} & \mathbf{o} & \mathbf{n} & \mathbf{a} & \mathbf{l} \\ \mathbf{T} & \mathbf{T} & \mathbf{O} & \mathbf{R} & \mathbf{A} & \mathbf{N} & \mathbf{L} & \mathbf{O} & \mathbf{I} & \mathbf{A} \end{array}\right]$$

, Hence **C = 'TTORANLOIA'**

Example: Carry out decryption for **C = 'TTORANLOIA'** of the previous example.

Solution: Here, the inverse key for decryption will be

$$\mathbf{k} = \begin{bmatrix} \mathbf{C} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{4} & \mathbf{3} & \mathbf{1} & \mathbf{5} & \mathbf{2} \end{bmatrix}$$

, i.e. the 1st letter in P will the 4th letter in C and the 2nd letter in P will be the 3rd letter in C and so on

C	**T**	**T**	**O**	**R**	**A**	**N**	**L**	**O**	**I**	**A**
P	**r**	**o**	**t**	**a**	**t**	**i**	**o**	**n**	**a**	**l**

Note: If the plaintext includes spaces, then the space is considered as a character. This will strengthen the system since plaintext words will be broken into other words.

Example: Encrypt the plaintext P=‘**the ball is big**’ using 5-character block transposition cipher with key

$$\mathbf{k} = \begin{bmatrix} \mathbf{P} \\ \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{3} & \mathbf{5} & \mathbf{2} & \mathbf{1} & \mathbf{4} \end{bmatrix}$$

Solution:

Plaintext	**t**	**h**	**e**		**b**	**a**	**l**	**l**		**i**	**s**		**b**	**i**	**g**
	1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Ciphertext	**E**	**B**	**H**	**T**		**L**	**I**	**L**	**A**		**B**	**G**		**S**	**I**

Hence C=‘EBHT LILA BG SI’

Notice that the structure of the plaintext sentence is completely changed in C

Example: Decrypt the Ciphertext "**ETQ HEU NESHG AEI NVRBHTIO A TE LAHYTS H IOPXDUB NXOXXY.** ", using a 1D Transposition method. if the above message is broken into five-character blocks (including spaces) and the letters in each block are rearranged according to the **inverse key** $\begin{bmatrix} \mathbf{C} \\ \mathbf{P} \end{bmatrix} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \mathbf{2} & \mathbf{5} & \mathbf{1} & \mathbf{4} & \mathbf{3} \end{pmatrix}$

(i.e., the 1st character in P is the 2nd character in C and the 2nd character in P is the 5th character in P and so on). The plaintext will be read from the following tables as shown:

“ETQ HEU NESHG AEI NVRBHTIO A TE LAHYTS H IOPXDUB NXOXXY. ”.

$$\begin{bmatrix}C\\P\end{bmatrix} = \begin{pmatrix}1 & 2 & 3 & 4 & 5\\2 & 5 & 1 & 4 & 3\end{pmatrix}$$

E	T	Q		H	E	U		N	E	S	H	G		A	E	I		N	V	R	B	H	T	I	O		A		T	E		L	A	H	Y	T	S		H		I	O	P	X	D	U	B		N	X	O	X	X	Y
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5

t	h	e		q	u	e	e	n		h	a	s		g	i	v	e	n		b	i	r	t	h		t	o		a		h	e	a	l	t	h	y		s	i	x		p	o	u	n	d		b	o	y	x	x	x
2	5	1	4	3	2	5	1	4	3	2	5	1	4	3	2	5	1	4	3	2	5	1	4	3	2	5	1	4	3	2	5	1	4	3	2	5	1	4	3	2	5	1	4	3	2	5	1	4	3	2	5	1	4	3

The plaintext is: “the queen has given birth to a healthy six pound boy”

Hint: The **original key** from P to C was $\begin{bmatrix}P\\C\end{bmatrix} = \begin{pmatrix}1 & 2 & 3 & 4 & 5\\3 & 1 & 5 & 4 & 2\end{pmatrix}$

Summary of the number of available key and no. of bit required for assign key

	No. of available key	No. of bit required for assign key
Substitution	$\mathbf{n!}$	$\log_2 \mathbf{n!}$
Transposition	$\mathbf{n!}$	$\log_2 \mathbf{n!}$
Polyalphabetic	$C_b^{n!} = \frac{(n!)!}{b!\,(n! - b)!}$ $\approx \frac{(n!)^b}{b!}$	$\log_2(C_b^{n!})$ $= \log_2(\frac{(n!)^b}{b!})$
Running Key	∞	$\log_2 \infty$