Classical Encryption Techniques

Dr. Risala T Khan

Professor

IIT, JU

Symmetric Cipher Model

Overview

A symmetric encryption scheme has five ingredients (Figure 3.1):

- **Plaintext:**
  This is the original intelligible message or data that is fed into the algorithm as input.
- **Encryption algorithm:**
  The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:**
  - The secret key is also input to the encryption algorithm.
  - The key is a value independent of the plaintext and of the algorithm.
  - The algorithm will produce a different output depending on the specific key being used at the time.
  - The exact substitutions and transformations performed by the algorithm depend on the key.

Cont..

- **Ciphertext:**
  - This is the scrambled message produced as output.
  - It depends on the plaintext and the secret key.
  - For a given message, two different keys will produce two different ciphertexts.
- **Decryption algorithm:**
  - This is essentially the encryption algorithm run in reverse.
  - It takes the ciphertext and the secret key and produces the original plaintext.

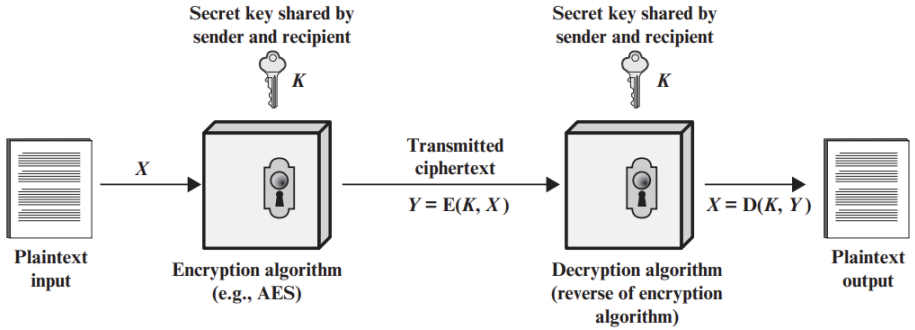


Figure 3.1 Simplified Model of Symmetric Encryption

Requirements

There are two requirements for secure use of conventional encryption:

1. We need a strong encryption algorithm.
   - At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key.
2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure.

Let us take a closer look at the essential elements of a symmetric encryption scheme.

- A source produces a message in plaintext, X = $[X_1, X_2, \ldots\ldots, X_M]$.
  - The M elements of X are letters in some finite alphabet.
- For encryption, a key of the form K = $[K_1, K_2, \ldots\ldots, K_J]$ 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 = $[Y_1, Y_2, \ldots\ldots, Y_N]$.

We can write this as:

$$Y = E(K, X)$$

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:

$$X = D(K, Y)$$

- 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 $\hat{X}$.
- 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 $\hat{K}$.

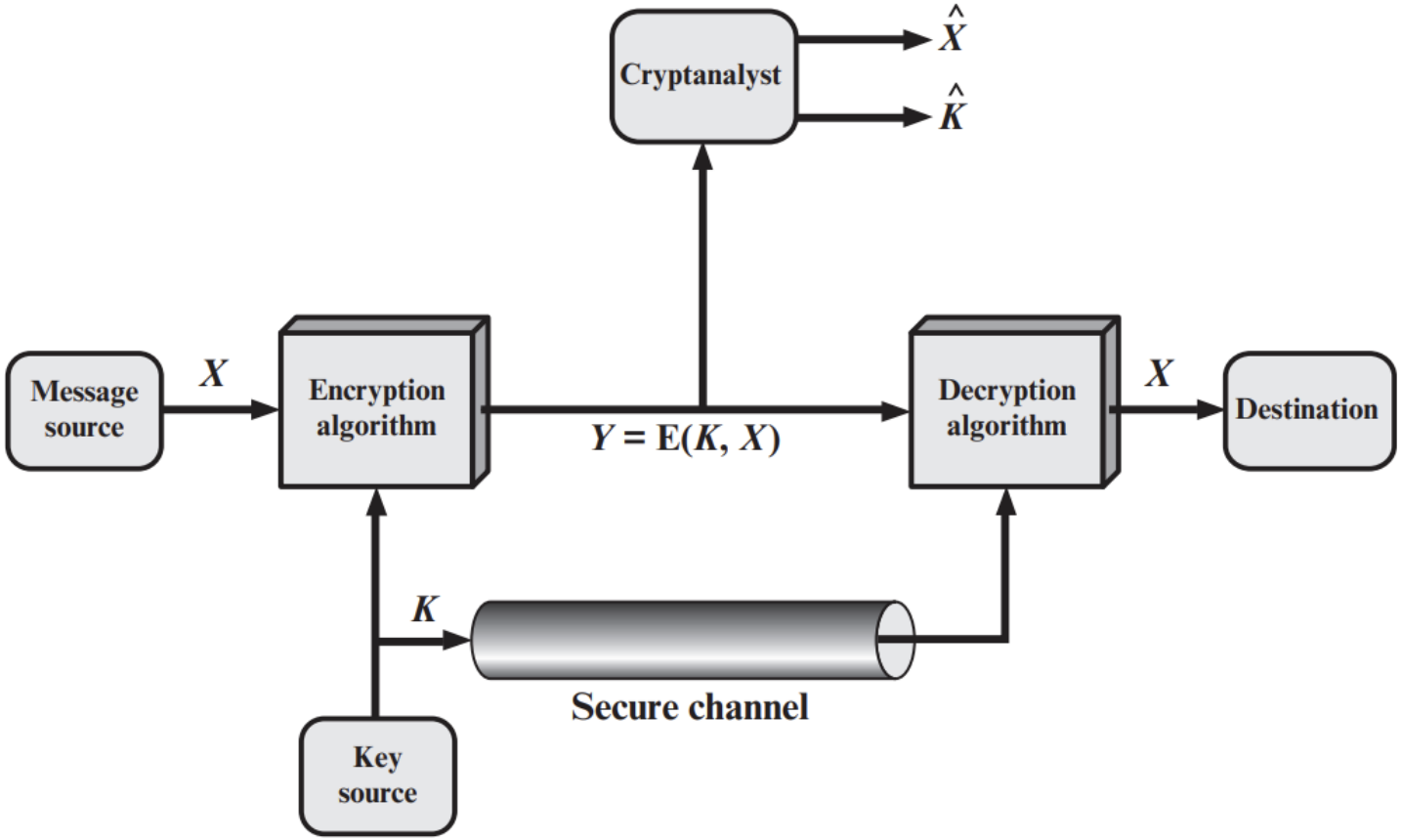


Figure 3.2 Model of Symmetric Cryptosystem

Basic characteristics of a Cryptographic System

Cryptographic systems are characterized along three independent dimensions:

1. **The type of operations used for transforming plaintext to ciphertext**.
   - All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and transposition, in which elements in the plaintext are rearranged.
   - The fundamental requirement is that no information be lost (i.e., that all operations are reversible).

Cont...

2. The number of keys used.

- If both sender and receiver use the same key, the system is referred to as **symmetric, single-key, secret-key, or conventional encryption.**
- If the sender and receiver use different keys, the system is referred to as **asymmetric, two-key, or public-key encryption.**

3. **The way in which the plaintext is processed.**

- A block cipher processes the input one block of elements at a time, producing an output block for each input block.
- A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.

Cryptanalysis and Brute-Force Attack

- **Cryptanalysis:**
  - Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs.
  - This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.
- **Brute-force attack:**
  - The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.
  - On average, half of all possible keys must be tried to achieve success.
- **If either type of attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.**

Types of Cryptanalytic Attacks on Encrypted Messages

- The table shows the various types of cryptanalytic attacks based on the amount of information known to the cryptanalyst.

Type of Attack	Known to Cryptanalyst
Ciphertext Only	■ Encryption algorithm ■ Ciphertext
Known Plaintext	■ Encryption algorithm ■ Ciphertext ■ One or more plaintext–ciphertext pairs formed with the secret key
Chosen Plaintext	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	■ Encryption algorithm ■ Ciphertext ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Cryptanalytic Attacks(ciphertext-only attack)

- **The ciphertext-only attack** is the hardest to break because the opponent has the least amount of information to work with.
  - In some cases, not even the encryption algorithm is known, but in general, we can assume that the opponent does know the algorithm used for encryption.
  - One possible attack under these circumstances is the brute-force approach of trying all possible keys.
  - If the key space is very large, this becomes impractical.
  - Thus, the opponent must rely on an analysis of the ciphertext itself, generally applying various statistical tests to it.

Cryptanalytic Attacks(known plaintext)

- In many cases, however, the analyst has more information.
  - The analyst may be able to capture one or more plaintext messages as well as their encryptions.
  - Or the analyst may know that certain plaintext patterns will appear in a message.
  - For example, a file that is encoded in the Postscript format always begins with the same pattern, or there may be a standardized header or banner to an electronic funds transfer message, and so on.
  - All these are examples of ***known plaintext***.
  - With this knowledge, the analyst may be able to deduce the key on the basis of the way in which the known plaintext is transformed.

Cryptanalytic Attacks(Cont..)

- Closely related to the *known-plaintext attack* is what might be referred to as a **probable-word attack**.
  - If the opponent is working with the encryption of some general prose message, he or she may have little knowledge of what is in the message.
  - However, if the opponent is after some very specific information, then parts of the message may be known.
  - For example, if an entire accounting file is being transmitted, the opponent may know the placement of certain key words in the header of the file.
  - As another example, the source code for a program developed by Corporation X might include a copyright statement in some standardized position.

Cryptanalytic Attacks(chosen-plaintext attack)

- If the analyst is somehow being able to get into the source system to insert a message chosen by the analyst into the system, then a **chosen-plaintext attack** is possible.
  - The idea is if the analyst is able to choose the plaintext messages to encrypt, submit those messages into the encryption system and after getting the cipher text the attacker tries to find the pattern or correlation between the plain text and cipher text to reveal the secret key.

Brute-force Attack

- A **brute-force attack** involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained.
  - On average, half of all possible keys must be tried to achieve success.
  - That is, if there are X different keys, on average an attacker would discover the actual key after X/2 tries.
  - It is important to note that there is more to a brute-force attack than simply running through all possible keys.
  - Unless known plaintext is provided, the analyst must be able to recognize plaintext as plaintext.
  - If the message is just plain text in English, then the result pops out easily, although the task of recognizing English would have to be automated.
  - If the text message has been compressed before encryption, then recognition is more difficult.
  - And if the message is some more general type of data, such as a numerical file, and this has been compressed, the problem becomes even more difficult to automate.
  - Thus, to supplement the brute-force approach, some degree of knowledge about the expected plaintext is needed, and some means of automatically distinguishing plaintext from garble is also needed.

SUBSTITUTION TECHNIQUE

Overview

- A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.
- If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.
- Traditional symmetric-key ciphers can be classified into two broad categories:

1. Substitution Cipher
2. Transposition Cipher

Monoalphabetic Cipher

- In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one.
- That is, a character or symbol in the plaintext is always changed to the same character or symbol in the ciphertext regardless of its position in the text.
- For example, if letter A in the plaintext is changed to letter D, every letter A is changed to letter D.

Example:

Additive cipher, Caesar cipher, multiplicative cipher, affine cipher etc. are some examples of monoalphabetic ciphers.

Additive Cipher

- **The simplest monoalphabetic cipher is the additive cipher**.
- This cipher is sometimes called a shift cipher and sometimes a Caesar cipher, but the term additive cipher better reveals its mathematical nature.
- The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet.
- For example:

  plain: meet me after the toga party

  cipher: PHHW PH DIWHU WKH WRJD 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 numeric 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 can be expressed as follows.

  For each plaintext letter p, substitute the ciphertext letter C:

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

  A shift may be of any amount, so that the general Caesar algorithm is:

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

  The decryption algorithm is simply

  $p = D(k, C) = (C - k) \bmod 26$

- If it is known that a given ciphertext is a shift cipher, then a brute-force attack is easily performed: simply try all the 25 possible keys.
- Figure shows the results of applying this strategy to the example ciphertext.
- In this case, the plaintext leaps out as occupying the third line.

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

Figure 3.3 Brute-Force Cryptanalysis of Caesar Cipher

NOTE:

1. Each character (uppercase or lowercase) is assigned an integer in Z_{26} . The secret key between Alice and Bob is also an integer in Z_{26} .
2. When the cipher is additive, the plaintext, ciphertext, and key are integers in Z_{26} .
3. The encryption algorithm adds the key to the plaintext character; the decryption algorithm subtracts the key from the ciphertext character. That is, encryption and decryption are inverse of each other.

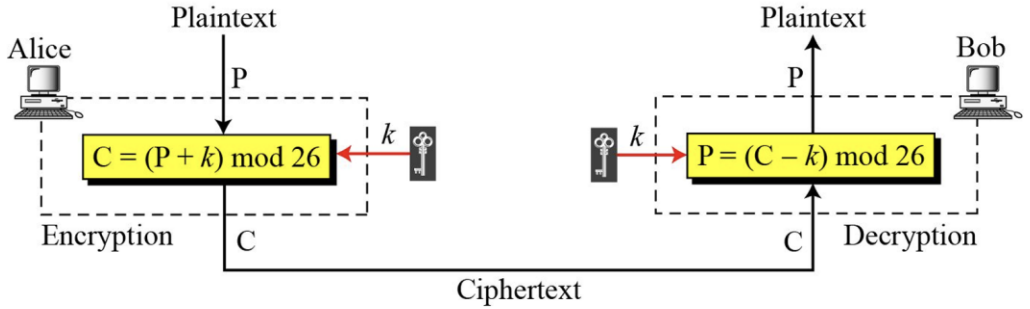


Figure: ***Additive cipher***

Ideal case for Brute Force attack

Three important characteristics of this problem enabled us to use a brute force 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.

Example

Use the additive cipher with key = 15 to encrypt the message "hello".

Solution

We apply the encryption algorithm to the plaintext, character by character:

Plaintext: h → 07	Encryption: (07 + 15) mod 26	Ciphertext: 22 → W
Plaintext: e → 04	Encryption: (04 + 15) mod 26	Ciphertext: 19 → T
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: o → 14	Encryption: (14 + 15) mod 26	Ciphertext: 03 → D

- The result is 'WTAAD'.

Use the additive cipher with key = 15 to decrypt the message "WTAAD".

Solution:

We apply the decryption algorithm to the plaintext character by character:

Ciphertext: W $\rightarrow$ 22	Decryption: (22 – 15) mod 26	Plaintext: 07 $\rightarrow$ h
Ciphertext: T $\rightarrow$ 19	Decryption: (19 – 15) mod 26	Plaintext: 04 $\rightarrow$ e
Ciphertext: A $\rightarrow$ 00	Decryption: (00 – 15) mod 26	Plaintext: 11 $\rightarrow$ l
Ciphertext: A $\rightarrow$ 00	Decryption: (00 – 15) mod 26	Plaintext: 11 $\rightarrow$ l
Ciphertext: D $\rightarrow$ 03	Decryption: (03 – 15) mod 26	Plaintext: 14 $\rightarrow$ o

- The result is 'hello'.
- Note that the operation is in modulo 26, which means that a negative result needs to be mapped to Z_{26}. (for example, -15 becomes 11).

Shift and Caesar Cipher

Shift Cipher:

- Historically, additive ciphers are called shift ciphers.
- Because, the encryption algorithm can be interpreted as “shift key character down” and the decryption algorithm can be interpreted as “shift key character up” .
- For example, if the key=15, the encryption algorithm shifts 15 character down. The decryption algorithm shifts 15 character up.

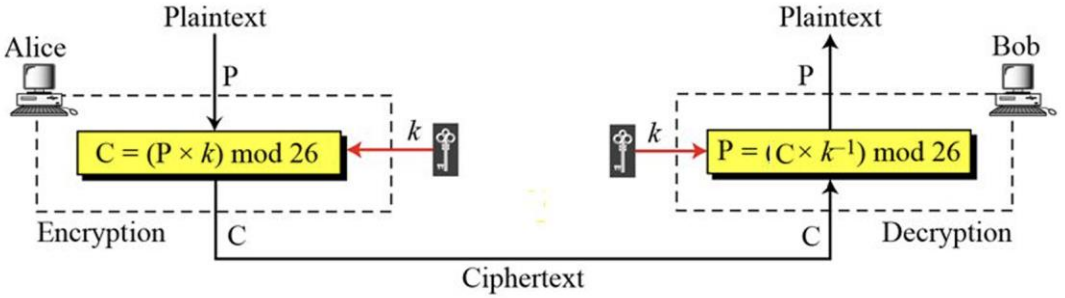
Caesar Cipher:

- Additive ciphers are also called Caesar cipher. Because, Julius Caesar used this cipher to communicate with his officers.
- Caesar used a key of 3 for his communications.
- That is, the cipher involves replacing each letter of the plaintext with the letter standing three places further down the alphabet.
- For example:

Plaintext	**:**	**Meet**	**me**	**after**	**the**	**lunch**
Ciphertext	:	PHHW	PH	DIWHU	WKH	OXQFK

Multiplicative Cipher

- In a multiplicative cipher The encryption algorithm specifies multiplication of the plaintext by the key.
- The decryption algorithm specifies division of the ciphertext by the key.
- In other words, decryption algorithm means multiplication of the ciphertext by the multiplicative inverse of the key.
- The plaintext and ciphertext are integers in Z_{26} , but the key is an integer in Z_{26} *.
- Encryption and decryption are inverse of each other. Figure shows the process of multiplicative cipher.



EXAMPLE

Encrypt the message "hello" with a key of 7 using multiplicative cipher.

Solution:

We apply the following encryption algorithm to the plaintext character by character: C=(Pxk)mod 26

Plaintext: h $\rightarrow$ 07	Encryption: $(07 \times 07) \bmod 26$	ciphertext: 23 $\rightarrow$ X
Plaintext: e $\rightarrow$ 04	Encryption: $(04 \times 07) \bmod 26$	ciphertext: 02 $\rightarrow$ C
Plaintext: l $\rightarrow$ 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 $\rightarrow$ Z
Plaintext: l $\rightarrow$ 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 $\rightarrow$ Z
Plaintext: o $\rightarrow$ 14	Encryption: $(14 \times 07) \bmod 26$	ciphertext: 20 $\rightarrow$ U

The ciphertext is "XCZZU".

Example:

Decrypt the message "XCZZU" with a key of 7 using multiplicative cipher.

Solution:

We apply the following decryption algorithm to the ciphertext character by character: $P=(C \times k^{-1}) \bmod 26$, where k^{-1} is the multiplicative inverse of k. Here the multiplicative inverse of 7 is 15 in z_{26}.

Ciphertext: X□23	Decryption: (23x15) mod 26	Plaintext: 05□h
Ciphertext: C□02	Decryption: (02x15) mod 26	Plaintext: 04□e
Ciphertext: Z□25	Decryption: (25x15) mod 26	Plaintext: 11□l
Ciphertext: Z□25	Decryption: (25x15) mod 26	Plaintext: 11□l
Ciphertext: U□20	Decryption: (20x15) mod 26	Plaintext: 14□o

The result is 'hello'.

Affine Cipher

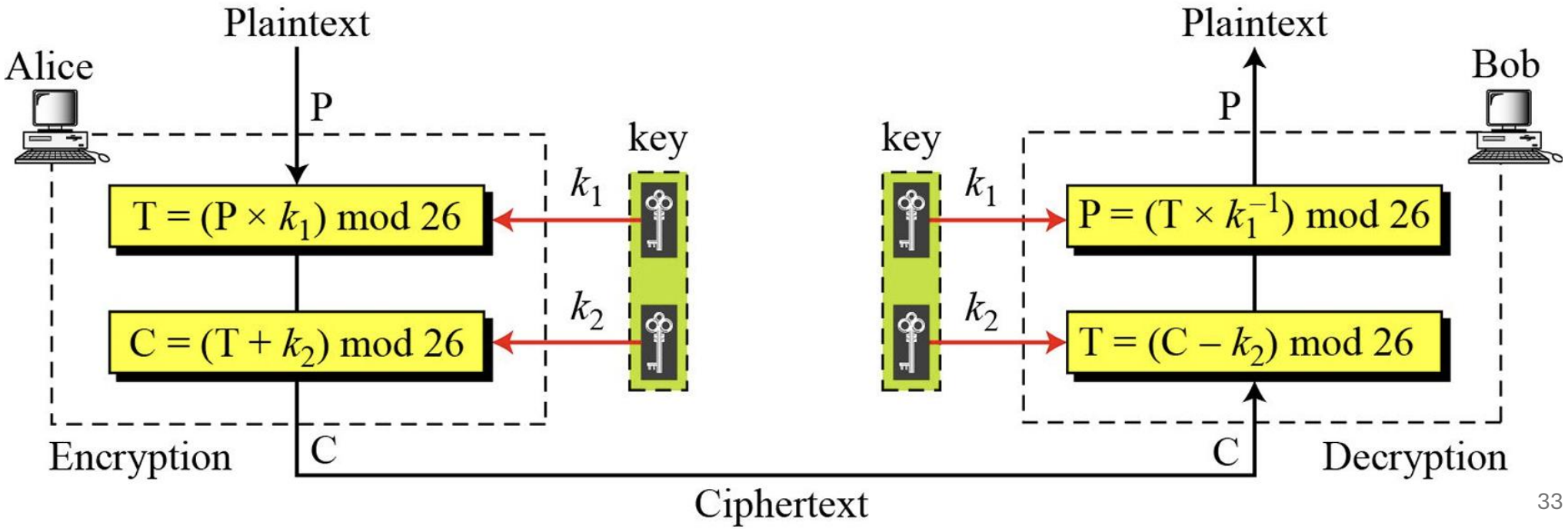
- It is the combination of additive and multiplicative ciphers with a pair of keys.
- The first key is used with the multiplicative cipher which comes from Z_{26} *.
  - This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.
- The second key is used with the additive cipher which comes from Z_{26} .
  - This set has only 26 members: 0, 1, 2, 3, 4, 5, ……, 25.
  - Therefore, the size of the key domain for any Affine cipher is

$$26 \times 12 = 312$$

- Figure below shows that Affine cipher is actually two ciphers, applied one after another.
- In Affine cipher, the encryption and decryption algorithms are based on the following two formulas:

$$C = (P \times k_1 + k_2) \bmod 26 \qquad P = ((C - k_2) \times k_1^{-1}) \bmod 26$$

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2



Affine Cipher

Example:

Use an affine cipher to encrypt the message "hello" with the key pair (7, 2) in modulo 26.

Solution:

We apply the following encryption algorithm to the plaintext character by character: $C=(P \times k_1 + k_2) \bmod 26$

P: h $\rightarrow$ 07	Encryption: $(07 \times 7 + 2) \bmod 26$	C: 25 $\rightarrow$ Z
P: e $\rightarrow$ 04	Encryption: $(04 \times 7 + 2) \bmod 26$	C: 04 $\rightarrow$ E
P: l $\rightarrow$ 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 $\rightarrow$ B
P: l $\rightarrow$ 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 $\rightarrow$ B
P: o $\rightarrow$ 14	Encryption: $(14 \times 7 + 2) \bmod 26$	C: 22 $\rightarrow$ W

The ciphertext is "ZEBBW".

Affine Cipher

Example:

Use the affine cipher to decrypt the message "ZEBBW" with the key pair (7, 2) in modulus 26.

Solution

We apply the following decryption algorithm to the ciphertext character by character: $C=((P - k_2) \times k^{-1}) \bmod 26$, where k^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2. Here additive inverse of 2 is 24 and multiplicative inverse of 7 is 15.

C: Z $\rightarrow$ 25	Decryption: $((25 - 2) \times 7^{-1}) \bmod 26$	P:07 $\rightarrow$ h
C: E $\rightarrow$ 04	Decryption: $((04 - 2) \times 7^{-1}) \bmod 26$	P:04 $\rightarrow$ e
C: B $\rightarrow$ 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P:11 $\rightarrow$ l
C: B $\rightarrow$ 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P:11 $\rightarrow$ l
C: W $\rightarrow$ 22	Decryption: $((22 - 2) \times 7^{-1}) \bmod 26$	P:14 $\rightarrow$ o

The plaintext is "hello".

Polyalphabetic Ciphers

- In polyalphabetic substitution, each occurrence of a character may have a different substitute.
- The relationship between a character in the plaintext to a character in the ciphertext is one-to-many.
- For example, if letter “a” could be enciphered as “D” in the beginning of the text, but as “N” at the middle.
- Polyalphabetic ciphers have the advantage of hiding the letter frequency of the undelying language.
- Eve cannot use the single-letter frequency statistics to break the ciphertext.
- Autokey cipher, playfair cipher, vigenere cipher, Hill cipher etc. are some examples of polyalphabetic ciphers.

Autokey Cipher

- In autokey cipher, the key is a stream of subkeys, in which each subkey is used to encrypt the corresponding plaintext character.
- The first subkey is a predetermined value secretly agreed upon by Alice and Bob.
- The second subkey is the value of the first plaintext character (between 0 to 25).
- The third subkey is the value of the second plaintext character. And so on.
- The name of this cipher as ‘autokey’ implies that the subkeys are automatically created from the plaintext cipher characters during the encryption process.
- Encryption and decryption is done using the following formulas.

$$P = P_1P_2P_3 \ldots \qquad C = C_1C_2C_3 \ldots \qquad k = (k_1, P_1, P_2, \ldots)$$

$$\text{Encryption: } C_i = (P_i + k_i) \bmod 26 \qquad \text{Decryption: } P_i = (C_i - k_i) \bmod 26$$

Autokey Cipher

Example for Encryption:

Assume that Alice and Bob agreed to use an autokey cipher with initial key value k_1 = 12. Now Alice wants to send Bob the message "Attack is today".

Solution:
Enciphering is done character by character.

1. Replace each plaintext character by its integer value (e.g *a* with 00, *b* with 01 etc.)
2. Write the 1st subkey (k_1=12) underneath the 1st plaintext character, 2nd subkey (k_2=00, which is the 1st plaintext character) underneath the 2nd plaintext character. And so on.
3. Now encrypt each plaintext character using the formula:

$$C_i=(P_i+k_i) \bmod 26$$

For example, for 3rd plaintext character ***t***, its corresponding ciphertext is-

$C_3=(P_3+k_3) \bmod 26$
$t=(19+19) \bmod 26$
$t=12=M$

Autokey Cipher

Example for Encryption (continue):

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P's Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	*12*	*00*	*19*	*19*	*00*	*02*	*10*	*08*	*18*	*19*	*14*	*03*	*00*
C's Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	**M**	**T**	**M**	**T**	**C**	**M**	**S**	**A**	**L**	**H**	**R**	**D**	**Y**

The ciphertext is "MTMTCMSALHRDY".

Note:

- We see that autokey cipher is a polyalphabetc cipher because the three occurrence of "a" in the plaintext are encrypted differently. The 1st 'a' is encrypted as M, the 2nd as T, and the 3rd as D.

Autokey Cipher

Example for Decryption:

With initial key value k_1 = 12, use the autokey cipher to decrypt the message sent by Alice to Bob: 'MTMTCMSALHRDY'.

Solution:

Deciphering is done character by character in the reverse direction.

1. Replace each ciphertext character by its integer value (e.g *M* with 12, *T* with 19 etc).
2. Write the 1st subkey (k_1=12) underneath the 1st ciphertext character and then find the first letter of the plaintext using the formula:

$$P_i=(C_i-k_i) \bmod 26.$$

For example, for 1st ciphertext character ***M***, its corresponding plaintext is-

$P_1=(C_1-k_1) \bmod 26$
$M=(12-12) \bmod 26$
$M=00=a$

3. Write the integer value of the first plaintext character as the 2nd subkey underneath the 2nd ciphertext character and find the plaintext character using above formula. And so on.

Autokey Cipher

Example for Decryption (continue):

Ciphertext	:	M	T	M	T	C	M	S	A	L	H	R	D	Y
C's Values	:	12	19	12	19	02	12	18	00	11	07	17	03	24
Key Stream	:	12	00	19	19	00	02	10	08	18	19	14	03	00
P's Values	:	00	19	19	00	02	10	08	18	19	14	03	00	24
Plaintext	:	a	t	t	a	c	k	i	s	t	o	d	a	y

The plaintext is "attack is today".

Playfair Cipher

- The best-known poly-alphabetic cipher is Playfair cipher.
- This cipher is invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair. It was used by the British Army during World War I.
- The secret key in this cipher is made of 25 alphabet letters arranged in a 5x5 matrix (letters I and J are considered the same when encrypting).
- Different arrangements of the letters in the matrix can create many different secret keys. One of the possible arrangement is shown in the figure here.
- Before encryption, the plaintext characters are grouped as two-character pairs.
- If the two letters in a pair are the same, a bogus letter is inserted to separate them.
- After inserting the bogus letters (if any), if the number of characters in the plaintext is odd, then one extra bogus character is added at the end to make the number of characters even.

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

Figure : ***Secret key in Playfair Cipher***

Playfair Cipher

Encryption rule for Playfair Cipher:

The playfair cipher uses three rules for encryption:

1. If the two letters in a pair are located in the same row of the secret key matrix, the corresponding encrypted character for each letter is the next letter to the right in the same row (with wrapping to the beginning of the row if the plaintext letter is the last character in the row).
2. If the two letters in a pair are located in the same column of the secret key matrix, the corresponding encrypted character for each letter is the letter beneath it in the same column (with wrapping to the beginning of the column if the plaintext letter is the last character in the column).
3. If the two letters in a pair are not located in the same row or column of the secret key matrix, the corresponding encrypted character for each letter is a letter that is in its own row but in the same column as the other letter.

Playfair Cipher

Example for Encryption:

Encrypt the plaintext "hello" using the secret key matrix shown in the figure below.

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

Secret Key

Solution:

- We group the plaintext as two-character pairs: "he // o"
- Here, in the second pair, the two letters are the same. So, we insert ***x*** as a bogus letter between the two *l's*. Now we have:

  "***he lx lo***".
- Now encrypt the message using the encryption rules for playfair cipher.

he → EC lx → QZ lo → BX

Plaintext: hello Ciphertext: ECQZBX

We see that playfair cipher is actually a polyalphabetc cipher because the two occurrence of "l" in the plaintext are encrypted differently, such as "Q" and "B".

Playfair Cipher

Decryption rule for Playfair Cipher:

The playfair cipher uses three rules for decryption:

1. If the two ciphertext letters in a pair are located in the same row of the secret key matrix, the corresponding decrypted character for each letter is the previous letter to the left in the same row (with wrapping to the end of the row if the ciphertext letter is the first character in the row).

2. If the two ciphertext letters in a pair are located in the same column of the secret key matrix, the corresponding decrypted character for each letter is the letter above it in the same column (with wrapping to the end of the column if the ciphertext letter is the first character in the column).

3. If the two ciphertext letters in a pair are not located in the same row or column of the secret key matrix, the corresponding decrypted character for each letter is a letter that is in its own row but in the same column as the other letter.

Playfair Cipher

Example for Decryption:

Decrypt the message "ECQZBX" using the secret key matrix shown in the figure below.

Solution:

- We group the ciphertext as two-character pairs: "EC *QZ* BX"
- Now decrypt the message using the decryption rules for playfair cipher.

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

Secret Key

EC → he QZ → lx BX → lo

Ciphertext: ECQZBX Plaintext: hello

Vigenere Cipher

- This cipher was designed by French mathematician Blaise de Vigenere.
- In this cipher, the secret key stream is created by repeating the initial secret key stream as many times as needed.
- The initial secret key stream of length m (where 1<=m<=26) is previously agreed upon by Alice and Bob.
- The cipher can be described as follows:

$P=P_1P_2P_3...C=C_1C_2C_3...$ $K=[(k_1,k_2,...,k_m), (k_1,k_2,...,k_m),...]$

Encryption: $C_i=(P_i+k_i) \bmod 26$

Decryption: $P_i=(C_i-k_i) \bmod 26$

Here, (k_1, k_2, k_3,….., k_m) is the initial secret key stream

Vigenere Cipher

Example for Encryption:

Encrypt the message "*She is listening*" using Vigenere cipher with the 6-character keyword "PASCAL".

Solution:

1. The initial key stream is "PASCAL" (*15, 0, 18, 2, 0, 11*). The key stream is the repetition of this initial key stream (as many times as needed).
2. Now encrypt each plaintext character using the formula $C_i=(P_i+k_i) \bmod 26$

Plaintext:	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P's values:	18	07	04	08	18	11	08	18	19	04	13	08	13	06
Key stream:	***15***	***00***	***18***	***02***	***00***	***11***	***15***	***00***	***18***	***02***	***00***	***11***	***15***	***00***
C's values:	**07**	**07**	**22**	**10**	**18**	**22**	**23**	**18**	**11**	**6**	**13**	**19**	**02**	**06**
Ciphertext:	**H**	**H**	**W**	**K**	**S**	**W**	**X**	**S**	**L**	**G**	**N**	**T**	**C**	**G**

The ciphertext is "HHWKSWXSLGNTCG".

Vigenere Cipher

Example for Decryption:

Decrypt the message "HHWKSWXSLGNTCG" using Vigenere cipher with the 6-character keyword "PASCAL".

Solution:

1. The initial key stream is "PASCAL" (*15, 0, 18, 2, 0, 11*). The key stream is the repetition of this initial key stream (as many times as needed).
2. Now decrypt each plaintext character using the formula $P_i=(C_i-k_i) \bmod 26$

Ciphertext	:	**H**	**H**	**W**	**K**	**S**	**W**	**S**	**X**	**L**	**G**	**N**	**T**	**C**	**G**
C's values	:	07	07	22	10	18	22	23	18	11	06	13	19	02	06
Key stream	:	15	00	18	02	00	11	15	00	18	02	00	11	15	00
P's values	:	18	07	04	08	18	11	08	18	19	04	13	08	13	06
Plaintext	:	s	h	e	i	s	/	i	s	t	e	n	i	n	g

The plaintext is "she is listening".

TRANSPOSITION CIPHERS

Transposition Cipher

- A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols.
- A symbol in the first position of the plaintext may appear in the ninth position of the ciphertext.
- A symbol in the eighth position of the plaintext may appear in the first position of the ciphertext.

Types of Transposition Cipher:

- There are three types of transposition cipher:
  - -Keyless Transposition Ciphers
  - -Keyed Transposition Ciphers
  - -Keyed Columnar Transposition Ciphers or Columnar Transposition Ciphers

Keyless Transposition Cipher

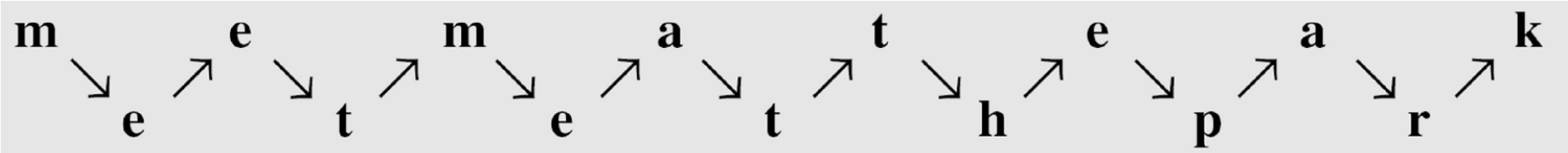
- These are simple transposition ciphers and were used in the past.
- There are two methods for permutation of characters:
  - In the first method, the text is written into a table column by column and then transmitted row by row.
  - In the second method, the text is written into a table row by row and then transmitted column by column.

Keyless Transposition Ciphers

Example:

1st Method: Written column by column and transmitted row by row

- A good example of a keyless cipher using the first method is the rail fence cipher.
- In this cipher, the plaintext is arranged in two lines as a zigzag pattern (which means column by column).
- The ciphertext is created reading the pattern row by row. For example, to send the message "Meet me at the park" to Bob, Alice writes-



By sending the first row followed by the second row, Alice then creates the ciphertext "MEMATEAKETETHPR".

Bob receives the ciphertext and divides in half (in this example, the second half has one less character). The first half forms the first row; the second half forms the second row. She reads the result in zigzag.

Because there is no key and the number of rows is fixed (2 here), the cryptanalysis of the ciphertext would be very easy for Eve.

Keyless Transposition Ciphers

Example:

2nd Method: Written row by row and transmitted column by column

- Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.

m	e	e	t
m	e	a	t
t	h	e	p
a	r	k	

By transmitting the characters column by column, Alice then creates the ciphertext "MMTAEEHREAEKTTP".

Bob receives the ciphertext and follows the reverse process. He writes the received message column by column and reads it row by row as the plaintext.

Eve can easily decipher the message if she knows the number of columns.

Example:

The cipher in the previous example is actually a transposition cipher. The following shows the permutation of each character in the plaintext into the ciphertext based on the positions.

Plaintext	:	**m**	**e**	**e**	**t**	**m**	**e**	**a**	**t**	**t**	**h**	**e**	**p**	**a**	**r**	**K**
Source Position	:	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
		□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Destination Position	:	01	05	09	13	02	06	10	14	03	07	11	15	04	08	12
Ciphertext	:	M	M	T	A	E	E	H	R	E	A	E	K	T	T	P

- The first character in the plaintext has not changed its position. The second character in the plaintext has moved to the fifth position in the ciphertext; the third character has moved to the ninth position; and so on.
- Although the characters are permuted, there is a pattern in the permutation: (01, 05, 09, 13), (02, 06, 10, 14), (03, 07, 11, 15), and (04, 08, 12). In each section, the difference between the two adjacent numbers is 4.

Keyed Transposition Cipher

- The **Keyed Transposition Cipher** is a type of transposition cipher where the letters of the plaintext are rearranged according to a specific key.
- The **Keyed Transposition Cipher** uses a keyword to determine the order in which to rearrange the columns of the text.

How Keyed Transposition Cipher Works

1. **Key**: A keyword is used to determine the number of columns and the order in which to read the columns for encryption. The letters in the key are assigned a numerical value based on their alphabetical order.
2. **Plaintext**: The plaintext is written into rows of a matrix/grid, with each row being as long as the number of letters in the key.
3. **Rearrange the Columns**: The columns are rearranged based on the alphabetical order of the letters in the keyword.
4. **Ciphertext**: The ciphertext is generated by reading the rearranged columns from top to bottom

Steps of the Keyed Transposition Cipher

- **Choose a Keyword**: The keyword defines the number of columns and the order in which to read the ciphertext. For example, the keyword “**cipher**”
- **Assign Numerical Values to the Keyword**: Assign numbers to the letters of the keyword based on their alphabetical order. For example:
  - Keyword : C I P H E R
  - Numerical Order : C(1) E(2) H(3) I(4) P(5) R(6)
  - Keyword: 145326
- **Write the Plaintext into Rows and Columns**: Write the plaintext into a matrix with the number of columns equal to the number of letters in the key. If the plaintext doesn’t fill the grid completely, you can pad it with extra characters (e.g., "X").
- **Rearrange the Columns Based on the Key**: Rearrange the columns of the grid according to the numerical order of the letters in the key.
- **Read the Columns to Form the Ciphertext**: Finally, the ciphertext is generated by reading the columns from top to bottom in the rearranged order.

Example

- PLAINTEXT: WE ARE DISCOVERED
- PLAINTEXT(after removing spaces): WEAREDISCOVERED
- KEYWORD: CIPHER
- STEP 1:
  - Assign numerical value to the keyword
  - Keyword : C I P H E R
  - Alphabetic Order: C E H I P R
  - Numerical Order : C(1) E(2) H(3) I(4) P(5) R(6)
  - Keyword: 145326

- STEP 2: Write the plaintext into grid :
  - Write the plaintext into grid with as many columns as there are letters in the key:

C(1)	I(4)	P(5)	H (3)	E(2)	R(6)
W	E	A	R	E	D
I	S	C	O	V	E
R	E	D	X	X	X

- Step 3: Rearrange the Columns Based on the Key
  - Rearrange the columns based on the key's alphabetical order. So, the columns will be rearranged according to:
  - KEY ORDER: C I P H E R--→ COLUMN ORDER-→1 4 5 3 2 6
  - The columns will be rearranged like this:

C(1)	E(2)	H(3)	I(4)	P(5)	R(6)
W	E	R	E	A	D
I	V	O	S	C	E
R	X	X	E	D	X

- Step 4: Generate the Ciphertext
  - Now, read the rearranged columns from top to bottom:
  - Column 1: W I R
  - Column 2: E V X
  - Column 3: R O X
  - Column 4: E S E
  - Column 5: A C D
  - Column 6: DEX
  - The final cipher text is: WIREVXROXESEACDDEX

- Step 5: Decryption Process
  - Create a grid with the same number of columns as the key
  - Write the ciphertext into the columns based on the key's alphabetical order
  - Rearrange the columns back to their original order.
  - Read the rows to retrieve the plaintext.

1	2	3	4	5	6
W	E	R	E	A	D
I	V	O	S	C	E
R	X	X	E	D	X

C(1)	I(4)	P(5)	H(3)	E(2)	R(6)
W	E	A	R	E	D
I	S	C	O	V	E
R	E	D	X	X	X

Columnar Transposition Ciphers

- This type of transposition cipher combines the keyless and keyed transposition ciphers to achieve better scrambling.
- Encryption or decryption is done in three steps:
    1. The text is written into a table row by row.
    2. The permutation is done by reordering the columns.
    3. The new table is read column by column.
- Here, the 1st and 3rd steps provide a keyless global reordering and the 2nd step provides a blockwise keyed reordering.

Columnar Transposition Ciphers

Example:

Encrypt the message "enemy attacks tonight" using Columnar transposition cipher.

Solution: The encryption and decryption is illustrated in the figure below.

- The 1st table in the figure is created by Alice writing the plaintext row by row.
- The columns are permuted using the key mentioned.
- The ciphertext is created by reading the 2nd table column by column.
- Bob does the same three steps in the reverse order.

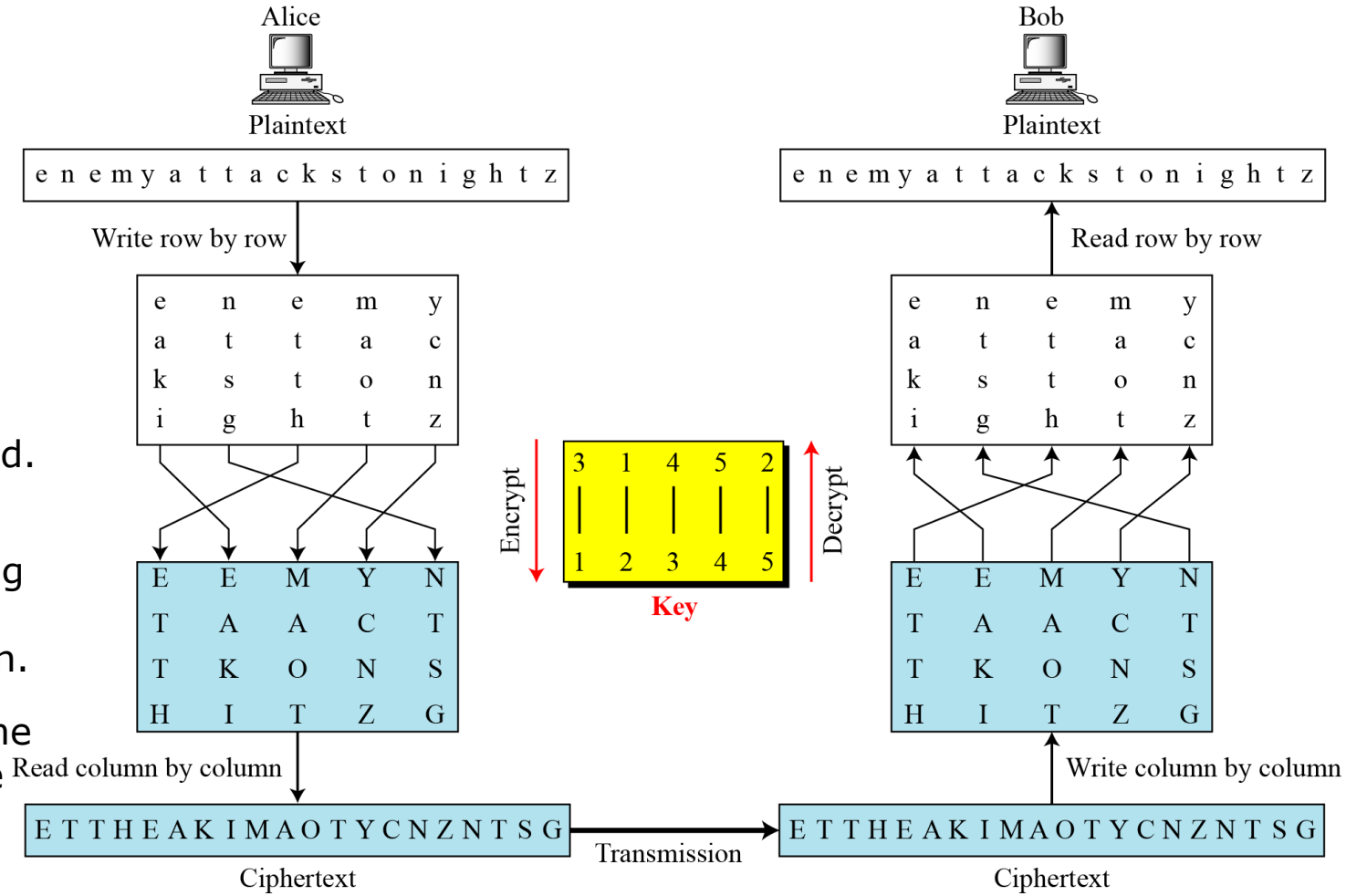


Figure: ***Combining Two Approaches***

Columnar Transposition Ciphers

Keys

- In the previous example, a single key was used in two directions for the column exchange:
  - downward for encryption
  - upward for decryption.
- It is customary to create two keys.

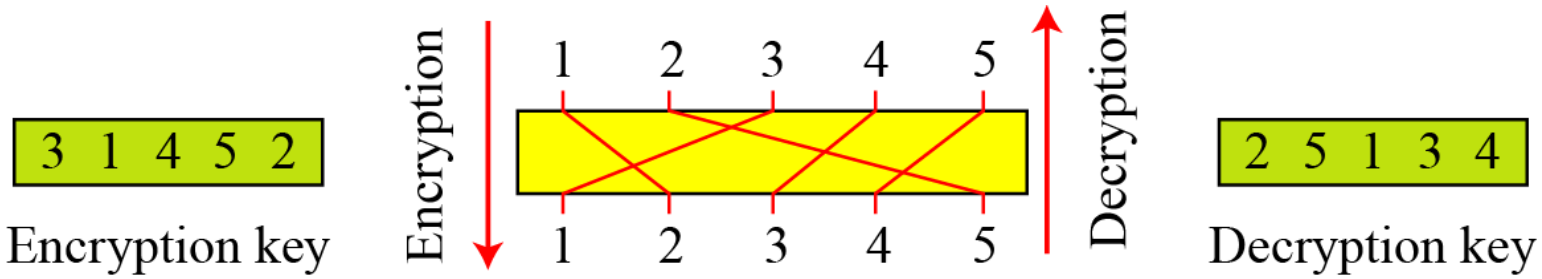


Figure: Encryption/decryption keys in transpositional ciphers

Key inversion in a transposition cipher

- How can the inverse of a key be created if the initial or original key is given, or vice versa?
- The process can be done manually in a few steps.
- Figure below shows how to invert an encryption key, i.e. how to find the decryption key.

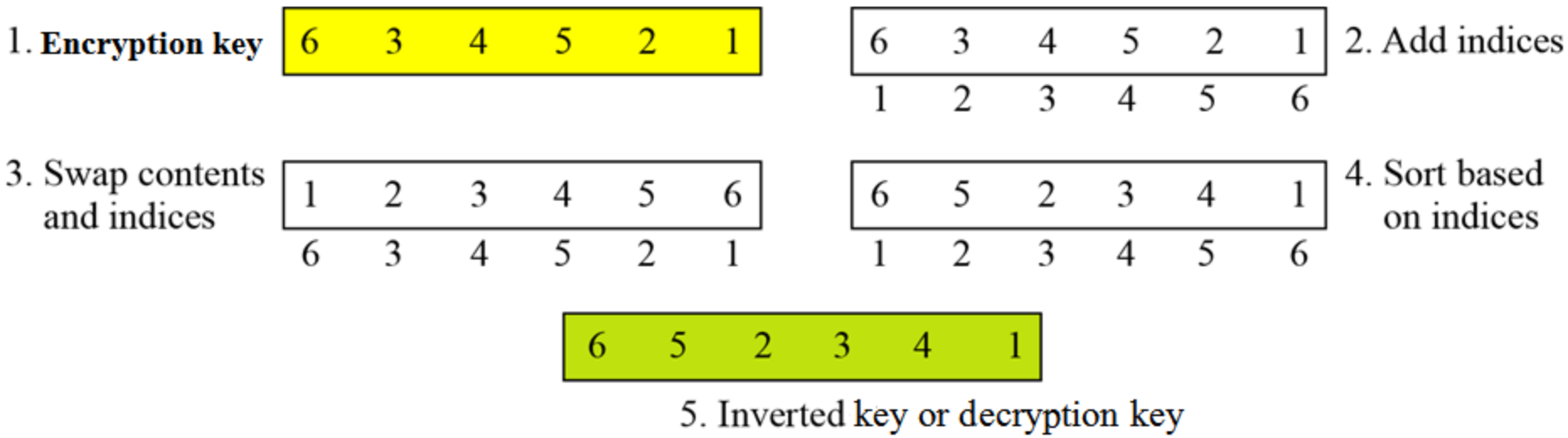


Figure: Inverting a permutation table