LEC6

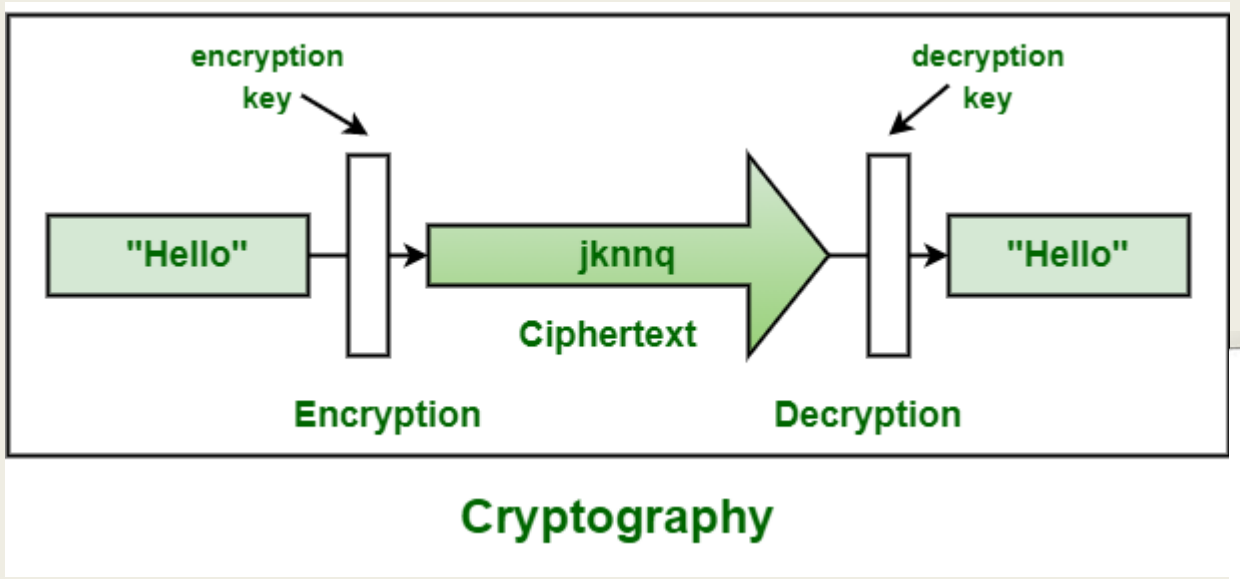
Classical encryption techniques I

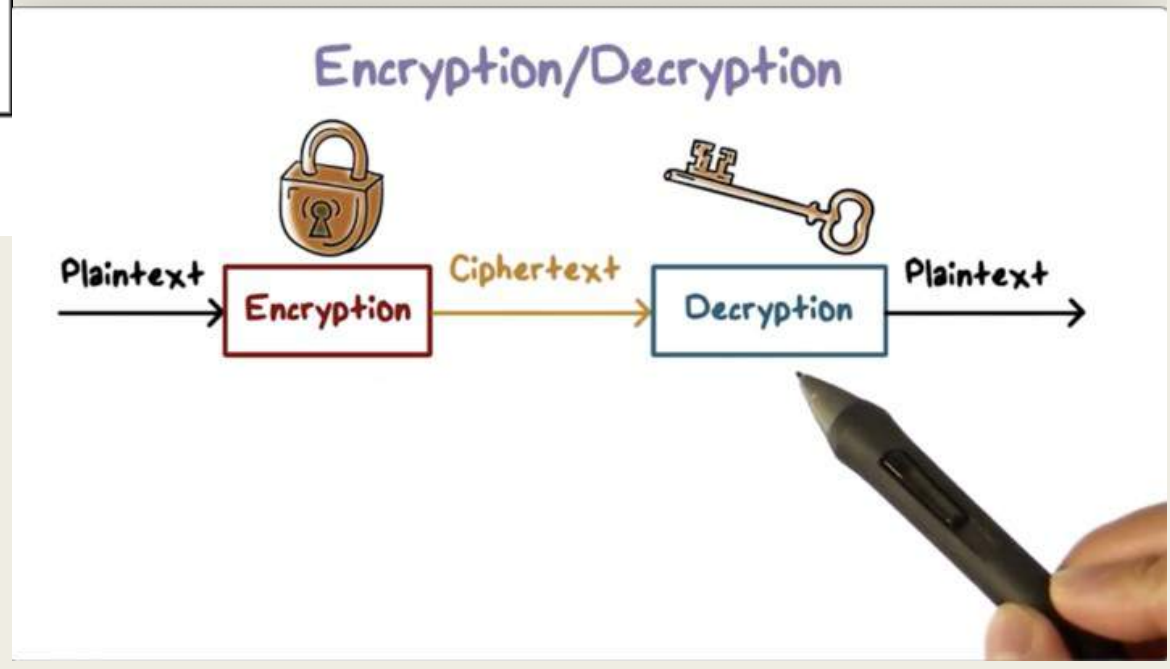
- Encryption Algorithms,
- Symmetric cipher,
- Transposition cipher
- Substitution cipher,
- Product cipher

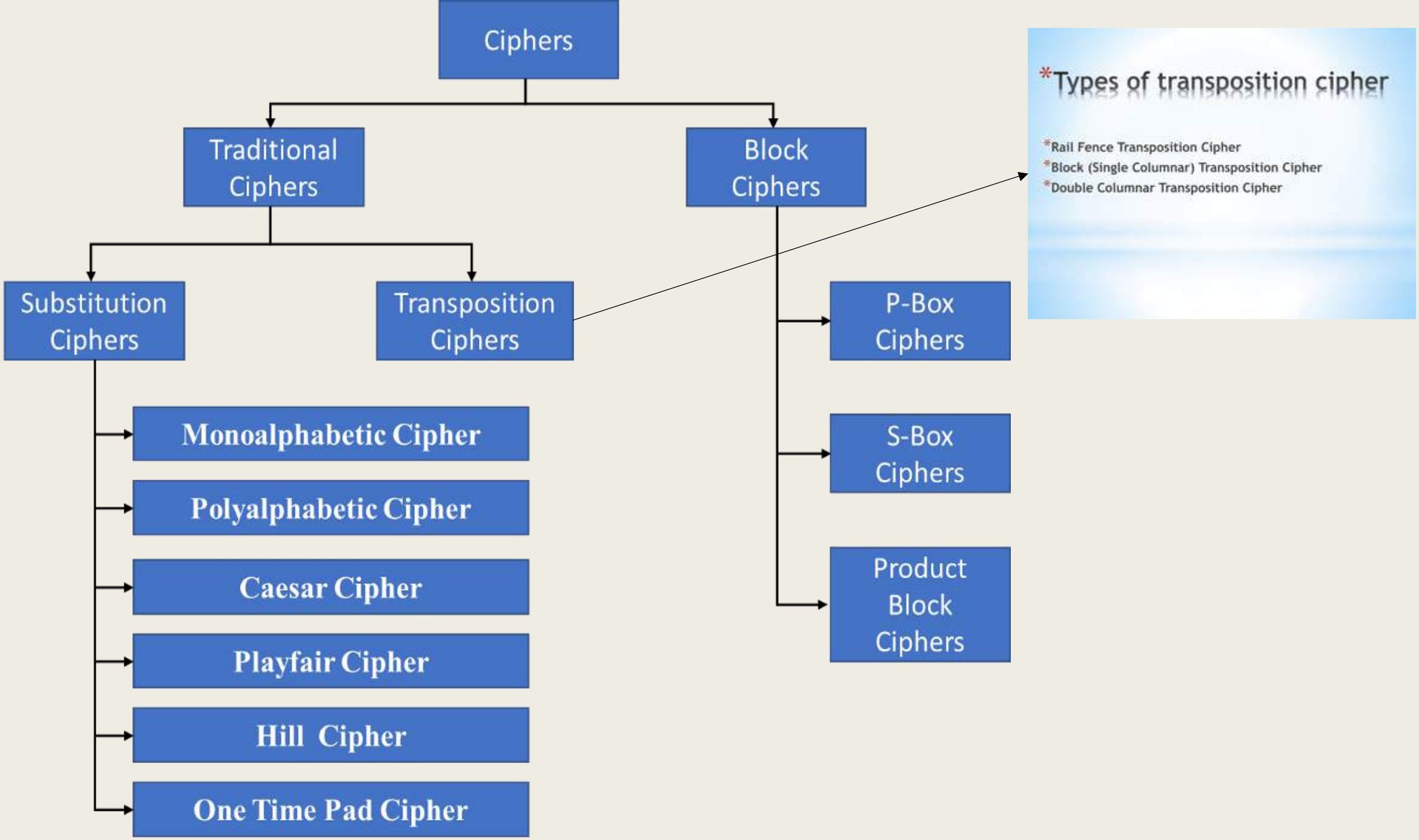
2025-2026
Second semester
2026-3-11

Alaa Y. Taqa

Encryption is the overall process of transforming data into an unreadable format (ciphertext) to ensure security, while a cipher is the specific algorithm or set of rules (e.g., AES, RSA) used to perform that transformation





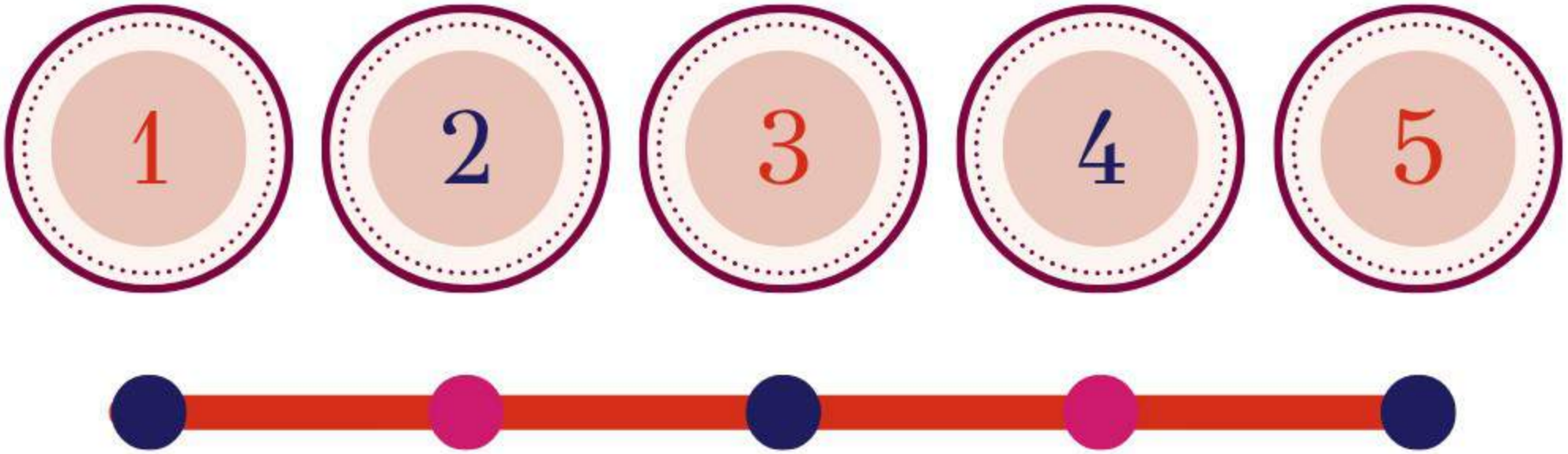

Ciphers
Traditional Ciphers
Block Ciphers
Substitution Ciphers
Transposition Ciphers
P-Box Ciphers
S-Box Ciphers
Product Block Ciphers
Monoalphabetic Cipher
Polyalphabetic Cipher
Caesar Cipher
Playfair Cipher
Hill Cipher
One Time Pad Cipher
*Types of transposition cipher
*Rail Fence Transposition Cipher
*Block (Single Columnar) Transposition Cipher
*Double Columnar Transposition Cipher

Basic Steps in Classical Encryption Process

The process generally follows these steps:

1. Sender writes a **plaintext message**
2. An **encryption algorithm** is applied
3. A **secret key** controls the transformation
4. Message becomes **ciphertext**
5. Receiver uses **decryption + key**
6. Original **plaintext is recovered**

Cryptography - Basic Process

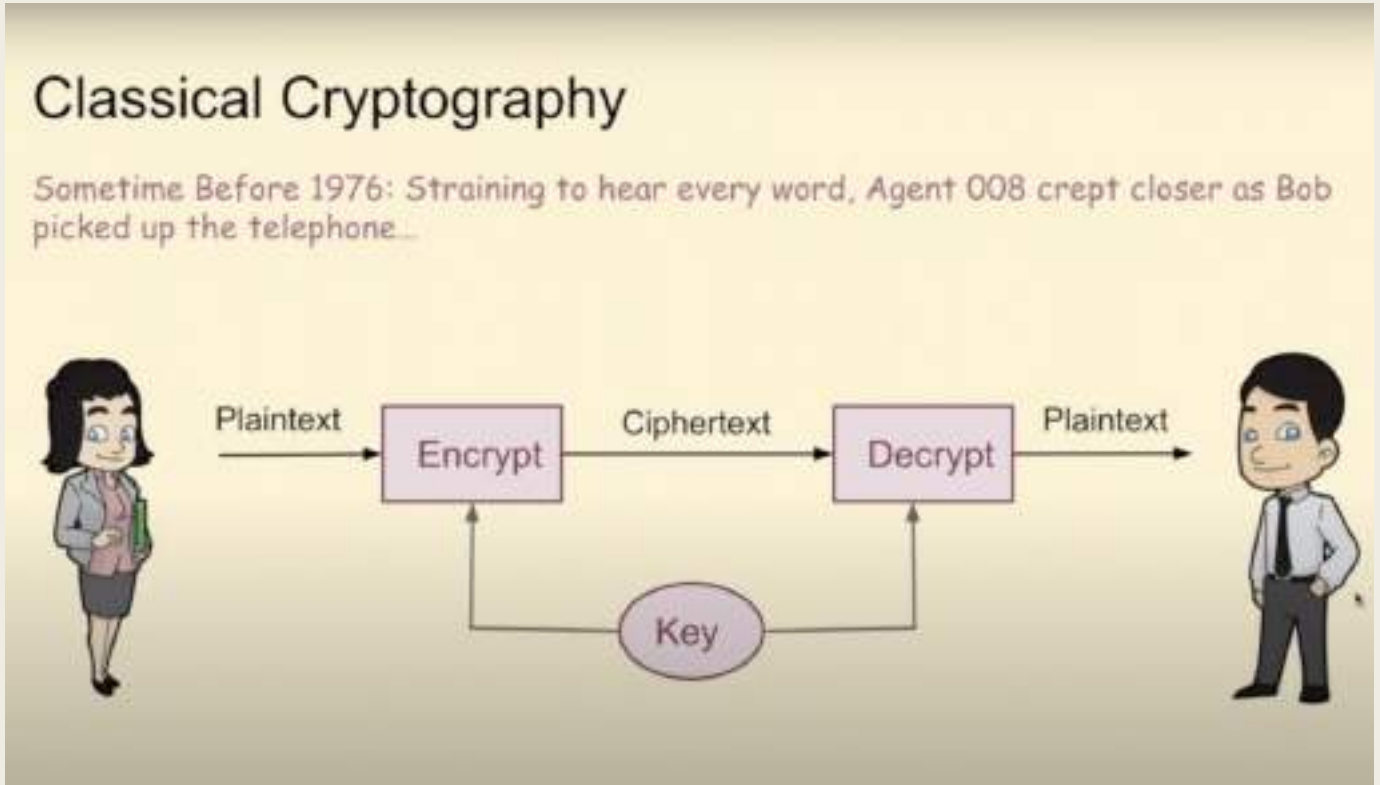


Definition of Classical Encryption Techniques

Classical encryption techniques are traditional methods used to protect information by converting readable data (**plaintext**) into an unreadable form (**ciphertext**) using mathematical rules and a **secret key**.

These techniques were mainly used **before modern computer-based cryptography** and are the foundation of the field of Cryptography.

The goal is to ensure that **only authorized people can read the message** by decrypting the ciphertext using the correct key.



Requirements of Classical Encryption

For classical encryption to work properly, several requirements are needed:

1. **Plaintext**
    1. The original readable message (e.g., text, letters).

2. **Encryption Algorithm**
    1. The mathematical procedure used to convert plaintext into ciphertext.

3. **Secret Key**
    1. A value used by the algorithm to control the encryption process.

4. **Ciphertext**
    1. The encrypted message that looks meaningless to attackers.

5. **Decryption Algorithm**
    1. The process that converts ciphertext back into plaintext.

6. **Key Distribution**
    1. Sender and receiver must **securely share the secret key**.

Substitution Techniques in Cryptography

Substitution techniques are encryption methods where **each element of the plaintext (letter, number, or symbol) is replaced with another element** according to a fixed rule or key.
They are one of the two main classical encryption methods in Cryptography (the other being transposition techniques).

Examples:

- Caesar Cipher
- Monoalphabetic Cipher
- Playfair Cipher
- Vigenère Cipher
- Hill Cipher
- time pad cipher

Definition

A **substitution cipher** replaces characters of **plaintext** with other characters to produce **ciphertext** using a **key**.
Example idea:
Plaintext ⟶ Ciphertext
A ⟶ D
B ⟶ E
C ⟶ F

Steps of Substitution Encryption

Step 1: Choose the plaintext

The original readable message.

Example

HELLO

Step2: Select the substitution rule or key

Example rule (shift 3 as in Caesar Cipher):

A → D

B → E

C → F

Step 3: Replace each letter

Apply the substitution rule to every letter.

H → K

E → H

L → O

L → O

O → R

Step 4: Generate ciphertext

Final encrypted message:

KHOOR

Caesar Cipher

The **Julius Caesar Cipher** is one of the simplest **substitution encryption techniques**. It is a type of **symmetric key cryptography** where each letter in the plaintext is shifted by a fixed number of positions in the alphabet.

For example, with a **shift of 3**:

- A → D
- B → E
- C → F

This method was historically used by **Julius Caesar** to send secret messages to his generals.

1. Caesar Cipher Algorithm

Encryption Algorithm

1.Start with a plaintext message.
2.Choose a **shift value (key)** k.
3.For each letter in the plaintext:
    1. Find its position in the alphabet.
    2. Shift it forward by k positions.
    3. Wrap around if it passes **Z**.
4.The result is the **ciphertext**.

Decryption Algorithm

1.Take the ciphertext.
2.Use the same key k.
3.Shift each letter **backward by k** positions.
4.Recover the **original plaintext**.

2. Mathematical Formula

For letters represented as numbers (A=0, B=1, ... Z=25):

Encryption

$$C = (P + k) \bmod 26$$

Decryption

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

Where:

- **P** = plaintext letter number
- **C** = ciphertext letter number
- **k** = key (shift value)

3. Example

Plaintext:

```
HELLO
```

Key:

```
k = 3
```

Encryption:

Plain	Shift	Cipher
H	+3	K
E	+3	H
L	+3	O
L	+3	O
O	+3	R

Ciphertext:

```
KHOOR
```

Decryption:

```
KHOOR → HELLO
```

4. MATLAB Code

```
% Caesar Cipher in MATLAB
clc;
clear;

text = input('Enter text: ','s');
k = input('Enter shift key: ');

text = upper(text);
cipher = '';

for i = 1:length(text)
    ch = text(i);
        if ch >= 'A' && ch <= 'Z'
        cipher_char = char(mod(double(ch)-65 + k, 26) + 65);
        cipher = [cipher cipher_char];
    else
        cipher = [cipher ch];
    end
end
disp(['Encrypted Text: ', cipher]);
```

```
% Decryption
plain = '';

for i = 1:length(cipher)
    ch = cipher(i);

    if ch >= 'A' && ch <= 'Z'
        plain_char = char(mod(double(ch)-65 - k, 26) + 65);
        plain = [plain plain_char];
    else
        plain = [plain ch];
    end
end
disp(['Decrypted Text: ', plain]);
```

Example MATLAB Output

Input:

```
Enter text: HELLO
Enter shift key: 3
```

Output:

```
Encrypted Text: KHOOR
Decrypted Text: HELLO
```

Monoalphabetic Cipher

A **Monoalphabetic Cipher** is a **substitution cipher** where each letter of the plaintext is replaced with **another fixed letter of the alphabet.**

The substitution remains **constant throughout the entire message.**

- Each plaintext letter → mapped to **one unique ciphertext letter**
- The mapping is defined by a **substitution key (permutation of alphabet).**

Mathematical Model

Let:

- $P = \{A, B, C, . , .. Z\}$ →plaintext alphabet
- $C = \{A, B, C, . , .. Z\}$ →ciphertext alphabet
- f → substitution function (a permutation)

Encryption Function

$$C = f(P)$$

where:

- P= plaintext letter
- C= ciphertext letter
- f= substitution mapping

Decryption Function

$$P = f^{-1}(C)$$

where:

- $f^{-1}$is the **inverse mapping** of the substitution.

Mapping Table

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

Encryption Algorithm

1. Define the **plaintext alphabet**.
2. Choose a **substitution key** (random permutation of alphabet).
3. For each plaintext letter:
    1. Find its position in the plaintext alphabet.
    2. Replace it with the corresponding letter from the cipher alphabet.
4. The result is the **ciphertext**.

Decryption Algorithm

1. Use the same substitution key.
2. Create the **inverse substitution table**.
3. For each ciphertext letter:
    1. Find its position in the cipher alphabet.
    2. Replace it with the corresponding plaintext letter.
4. Recover the **original message**.

Example

Plaintext

```
HELLO
```

Substitution Key

```
Plain  : ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher : QWERTYUIOPASDFGHJKLZXCVBNM
```

Encryption

Plain	Cipher
H	I
E	T
L	S
L	S
O	G

Ciphertext

```
ITSSG
```

Decryption

```
ITSSG → HELLO
```

```
clc;
clear;
plain = input('Enter plaintext: ','s');

alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
cipherKey = 'QWERTYUIOPASDFGHJKLZXCVBNM';
% substitution key

plain = upper(plain);
cipher = '';
% Encryption
for i = 1:length(plain)
    ch = plain(i);
        idx = find(alphabet == ch);
        if ~isempty(idx)
        cipher = [cipher cipherKey(idx)];
    else
        cipher = [cipher ch];
    end
end

disp(['Encrypted Text: ', cipher]);
```

```
% Decryption
decrypted = '';

for i = 1:length(cipher)
    ch = cipher(i);

    idx = find(cipherKey == ch);

    if ~isempty(idx)
        decrypted = [decrypted alphabet(idx)];
    else
        decrypted = [decrypted ch];
    end
end
disp(['Decrypted Text: ', decrypted]);
```

Example MATLAB Output

Input

```
Enter plaintext: HELLO
```

Output

```
Encrypted Text: ITSSG
Decrypted Text: HELLO
```

Hill Cipher

The **Lester S. Hill Cipher** is a **polygraphic substitution cipher** that encrypts **blocks of letters** using **linear algebra and matrix multiplication**.
It was introduced in **1929** by Lester S. Hill and is based on operations in **Linear Algebra and **Cryptography**.
Unlike simple substitution ciphers, the Hill cipher **encrypts multiple letters at once**, making it harder to break using frequency analysis.

A **Hill Cipher** is a **block cipher** in which a group of plaintext letters is represented as a **vector**, and encryption is performed by multiplying this vector by a **key matrix**, followed by **modulo 26 arithmetic**.

Mathematical Model

Let $P = (p_1, p_2, .., p_n)$
be the **plaintext vector**.
Let K be the $n \times n$key matrix.

Encryption Formula

$$C = KP \pmod{26}$$

Where
- P= plaintext vector
- K= key matrix
- C= ciphertext vector
- operations are done **mod 26**.

Decryption Formula

$$P = K^{-1}C \pmod{26}$$

Where
- $K^{-1}$is the **inverse matrix of** K**modulo 26**.

Alphabet Number Representation

Letter	Number
A	0
B	1
C	2
…	…
Z	25

Encryption Algorithm

1. Choose a **key matrix** K of size $n \times n$.
2. Convert plaintext letters into numbers.
3. Divide plaintext into **blocks of size** n.
4. Convert each block into a **vector** P.
5. Compute

$$C = KP \mod 26$$

6. Convert the resulting numbers back to letters.
7. Combine all blocks to obtain the **ciphertext**.

Decryption Algorithm

1. Compute the **inverse matrix** K^{-1} **modulo 26**.
2. Convert ciphertext letters to numbers.
3. Divide ciphertext into blocks.
4. For each block compute

$$P = K^{-1}C \mod 26$$

5. Convert numbers back to letters to obtain the **original plaintext**.

Key Matrix

$$K = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$$

Plaintext

```
HI
```

Convert to numbers

```
H = 7
I = 8
```

Vector

$$P = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

Encryption

$$C = KP \mod 26$$

$$= \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

$$= \begin{bmatrix} 45 \\ 54 \end{bmatrix}$$

Now mod 26

```
45 mod 26 = 19
54 mod 26 = 2
```

Numbers

```
19 = T
2 = C
```

Ciphertext

```
TC
```

```
clc;
clear;
alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';

% Key Matrix
K = [3 3; 2 5];

plaintext = input('Enter plaintext (2 letters): ','s');
plaintext = upper(plaintext);
% Convert letters to numbers
P = double(plaintext) - 65;
P = P';

% Encryption
C = mod(K * P,26);
cipher = char(C' + 65);
disp(['Ciphertext: ', cipher])

% Decryption
K_inv = mod(round(inv(K)*det(K)),26);
P2 = mod(K_inv * C,26);
plain = char(P2' + 65);

disp(['Decrypted text: ', plain])
```

Example MATLAB Output

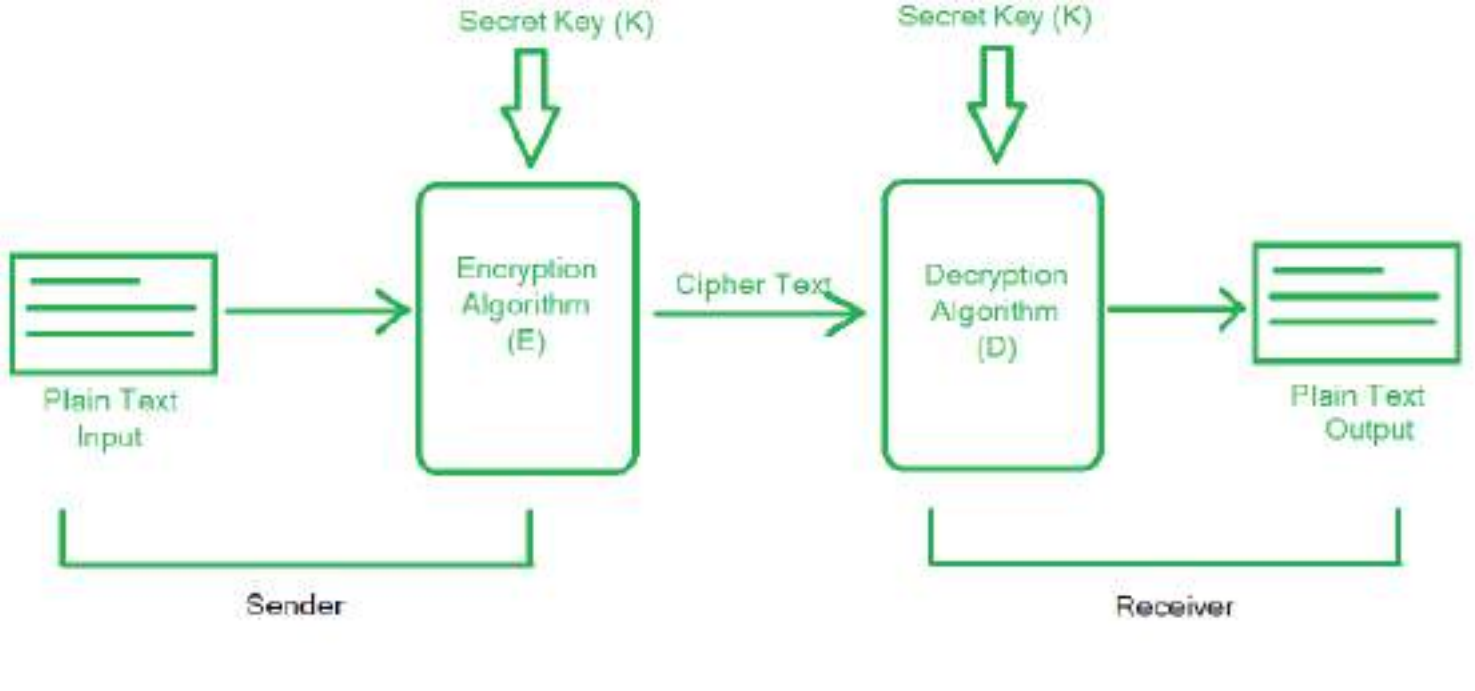
Input

```
Enter plaintext: HI
```

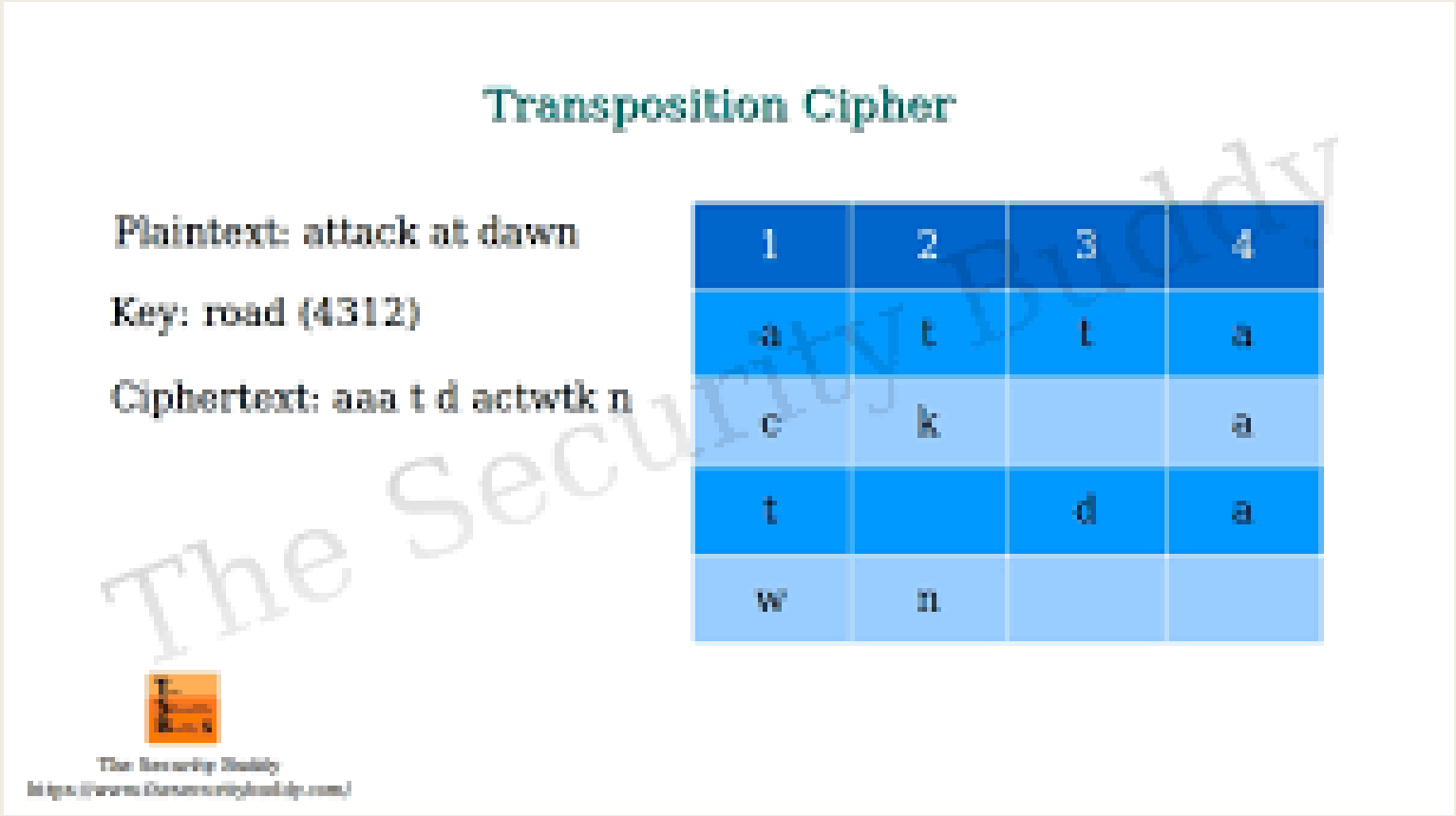
Output

```
Ciphertext: TC
Decrypted text: HI
```

A symmetric cipher is a cryptographic method that uses the same single secret key for both encryption and decryption of data. It is highly efficient and faster than asymmetric encryption, making it ideal for large data volumes and real-time communication.



A transposition cipher is a cryptographic method that secures data by systematically rearranging the order of plaintext characters based on a specific algorithm or key, rather than substituting them. Unlike substitution ciphers, the original letters remain unchanged but are shuffled like pieces of a



Example (Columnar Transposition)

Plaintext

```
HELLOWORLD
```

Key = **4 columns**

Write plaintext row-wise

H	E	L	L
O	W	O	R
L	D	X	X

(X added as padding)

Read column-wise

Column1 → H O L
Column2 → E W D
Column3 → L O X
Column4 → L R X

Ciphertext

```
HOLEWDLOXLRX
```

Product Cipher A **Product Cipher** is an encryption technique in **Cryptography** that combines **two or more simple ciphers** (usually **substitution and transposition**) to produce a **stronger encryption system**.
The idea is that **one cipher alone may be weak**, but combining multiple ciphers creates a much more secure system.


Questions


THANK
YOU