Computer Network Security (CNS)

INFT SEMESTER V

VINITA BHANDIWAD

Course Code	Course Name	Teaching Scheme (Contact Hours)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
ITL502	Security Lab	--	02	--	01	01

Course Code	Course Name	Examination Scheme							
		Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test1	Test 2	Avg.					
ITL502	Security Lab	--	--	--	--	--	25	25	50

Syllabus and scheme

Course Code	Course Name	Teaching Scheme (Contact Hours)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
ITC502	Computer Network Security	03	--	03	--	03

Course Code	Course Name	Examination Scheme							
		Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test1	Test2	Avg.					
ITC502	Computer Network Security	20	20	20	80	03	--	--	100

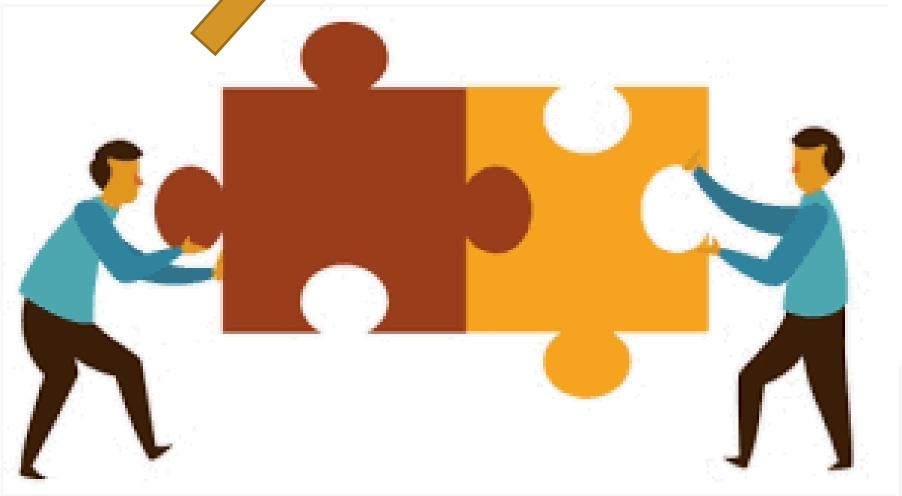
Contents:

1. Introduction to Network Security & cryptography
2. Cryptography: Key management, distribution and user authentication
3. Malicious Software
4. IP Security, Transport level security and Email Security
5. Network Management Security and Network Access Control
6. System Security

Goal of this subject

Confidentiality

Integrity



Availability

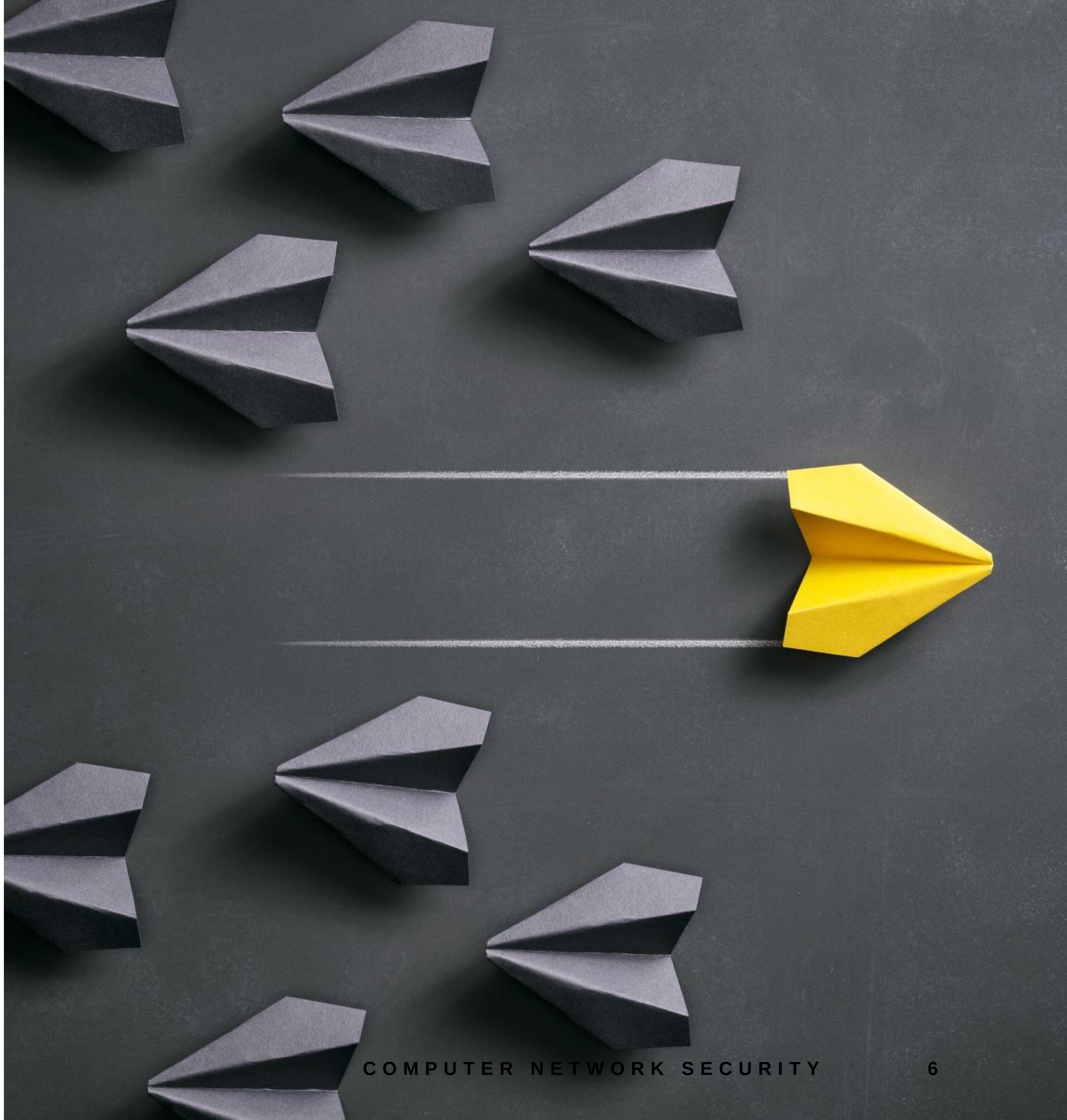
Goal…..

To achieve CIA……

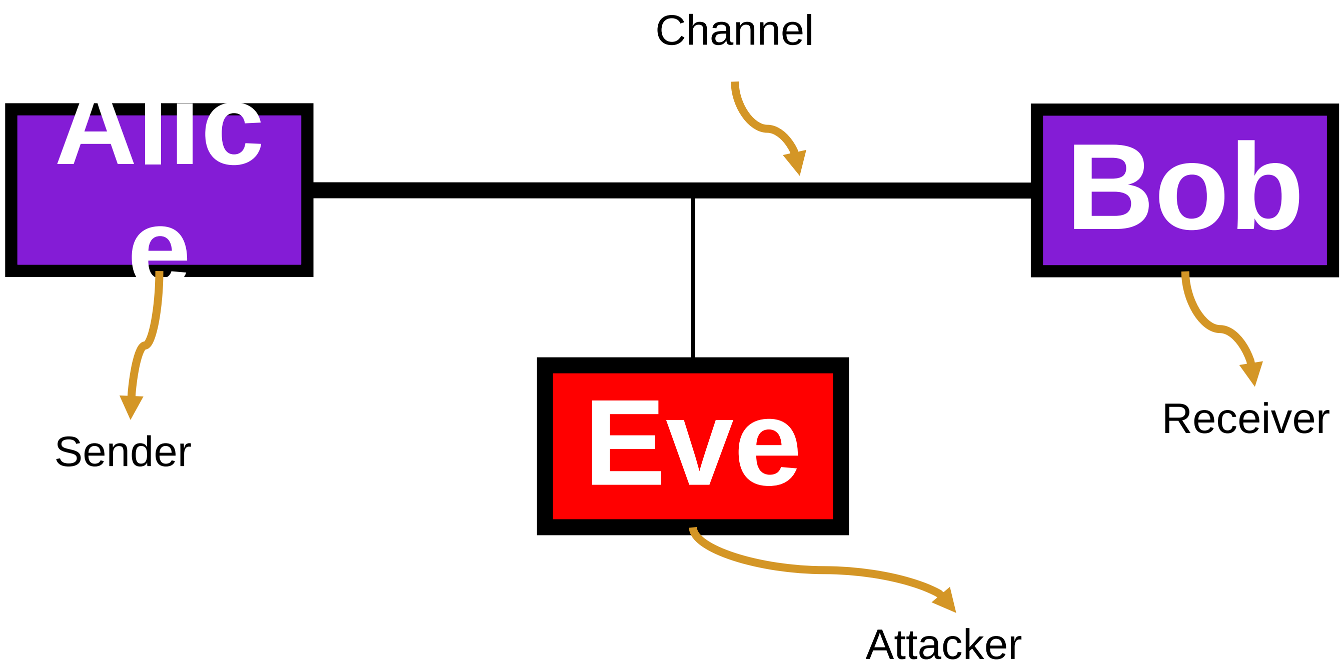
- Confidentiality
- Integrity
- Availability

Let us start with fundamental point??

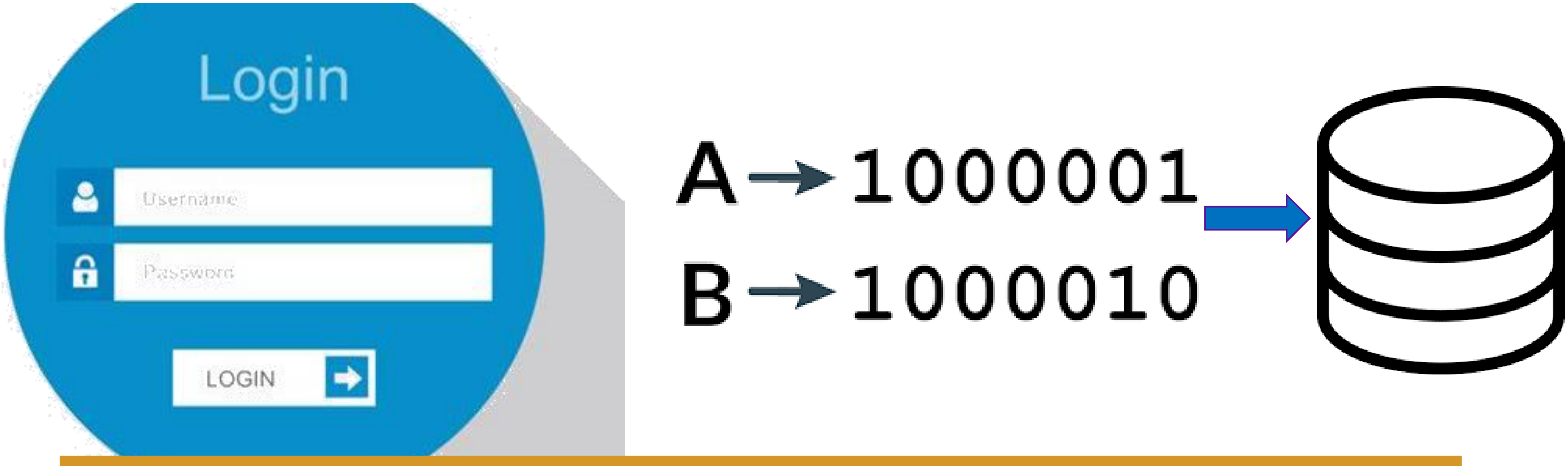
Why is security is needed in first place?



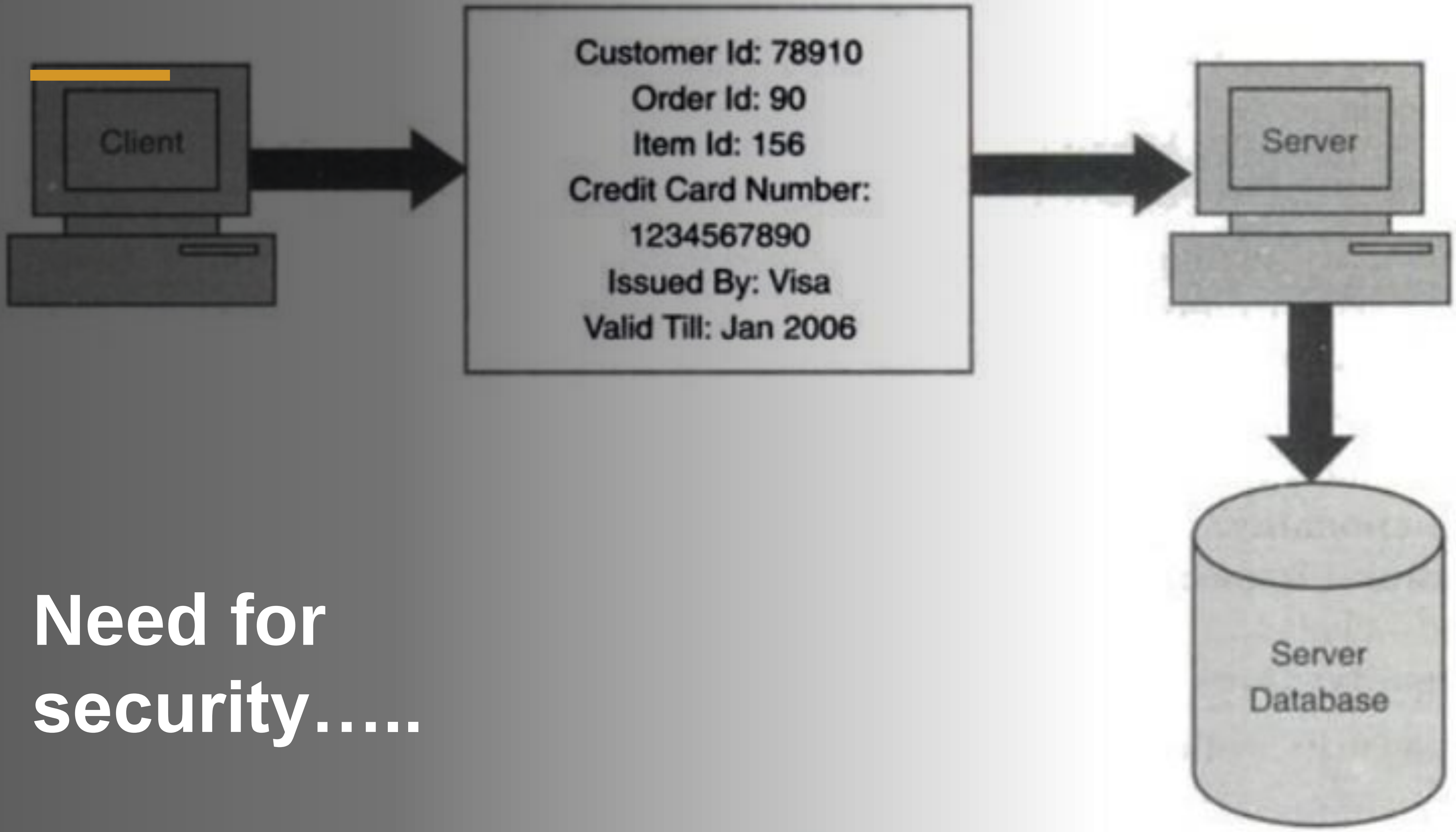
Need of security…..



Need for Network Security

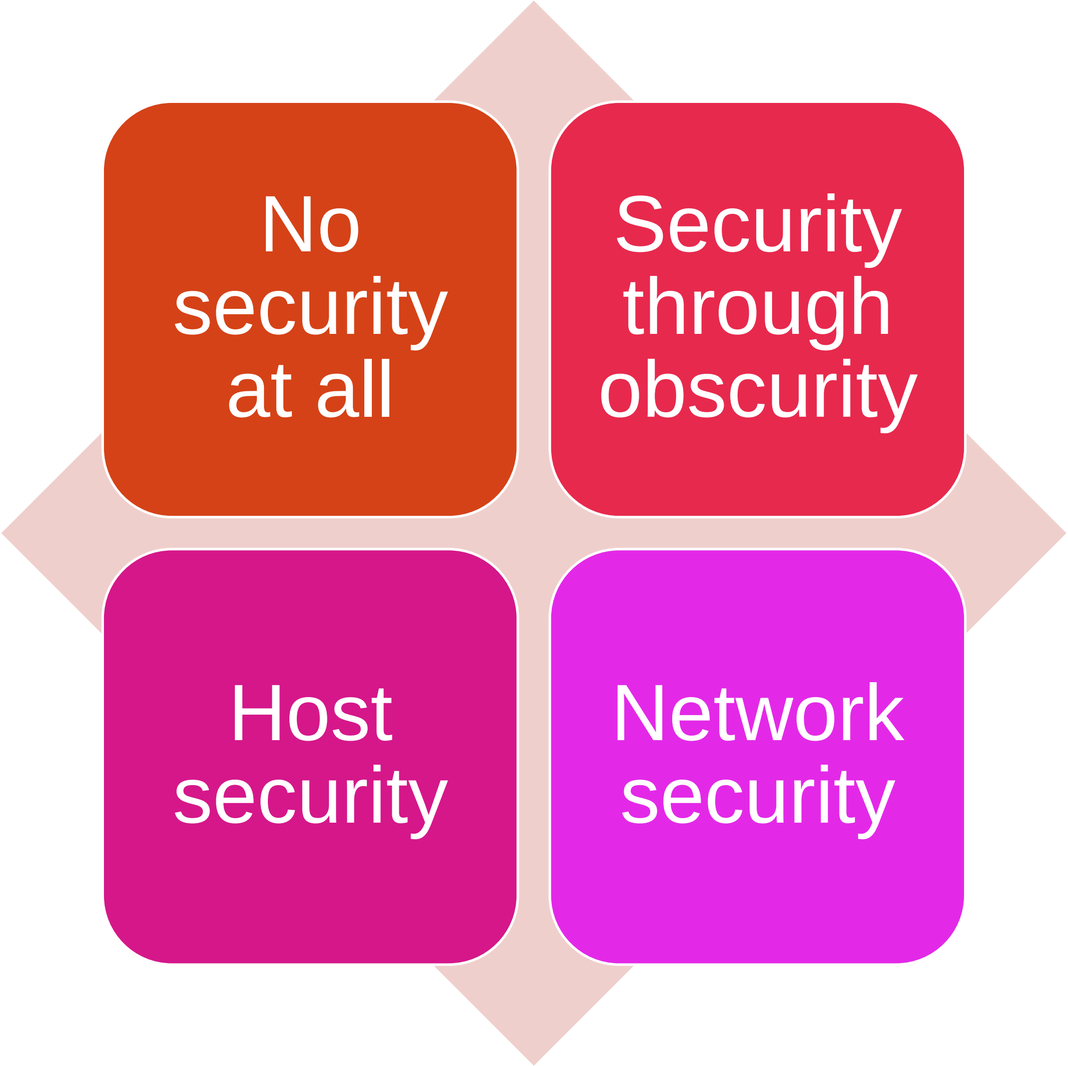


Two approaches/ security mechanism were followed



Need for security…..

Security approaches used….



OSI Security architecture

It mainly focuses on:

- Security attack
- Security mechanism
- **Security service**

Security Service:

- Authentication
- Access control
- Data integrity
- Non repudiation
- Data confidentiality

Security mechanism and attacks

- Specific security mechanism
- Pervasive security mechanism

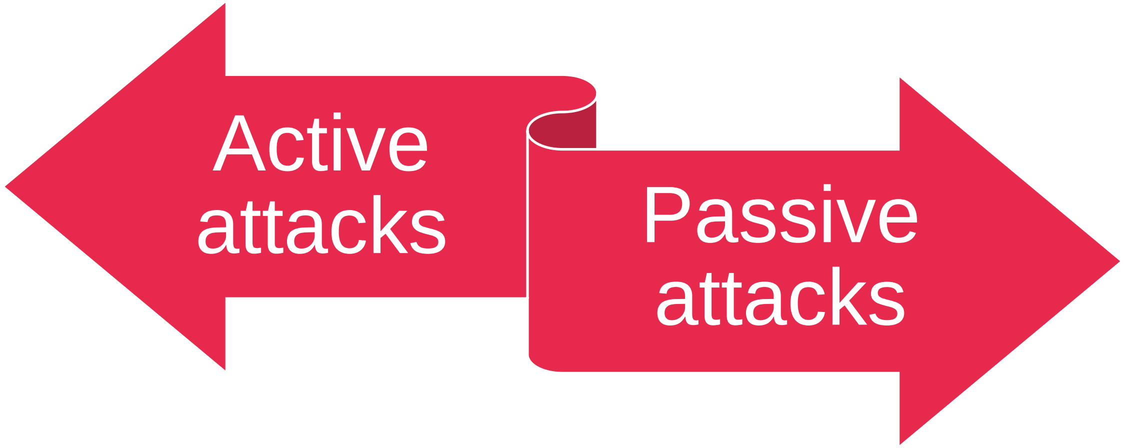
Security attacks:

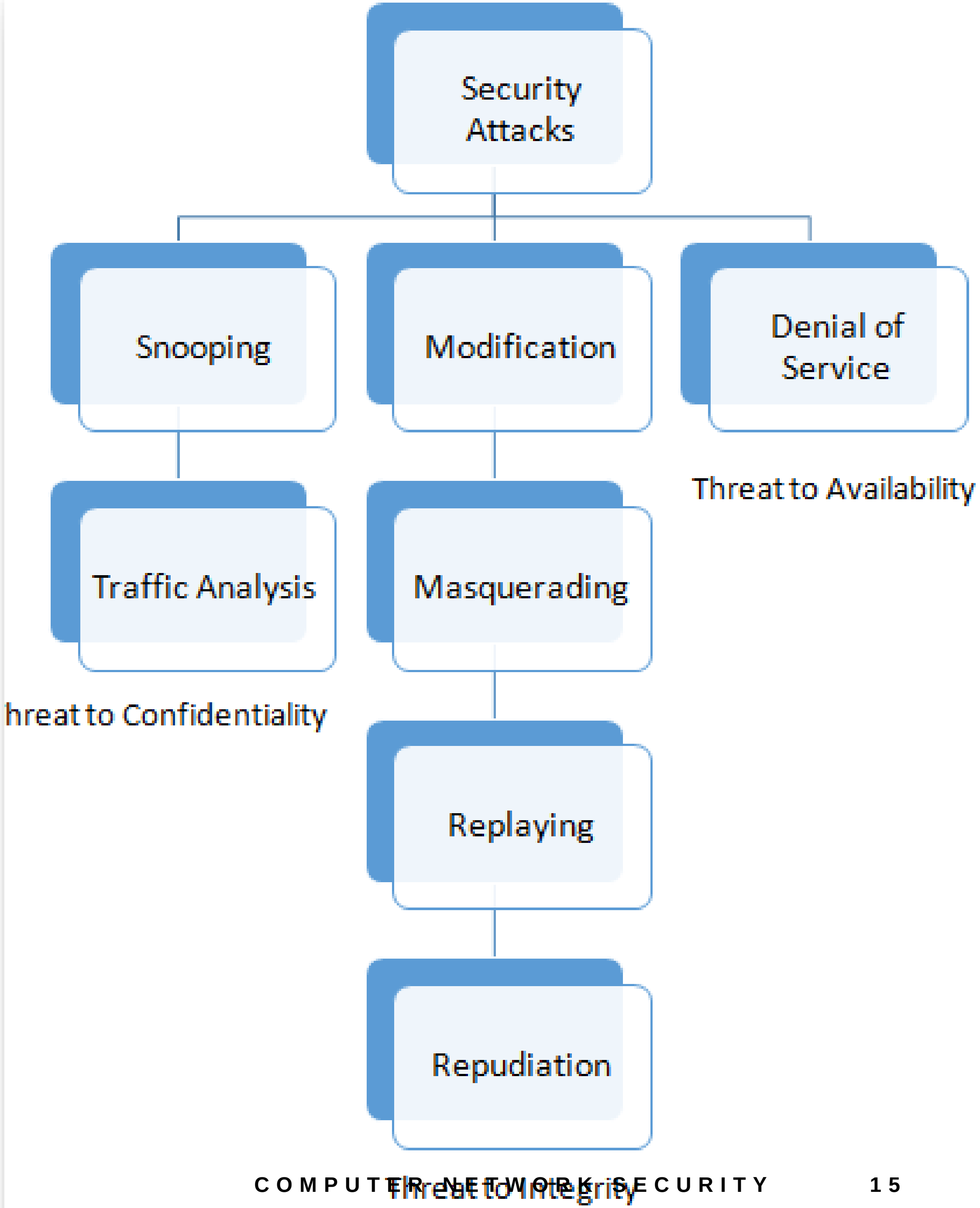
Lets summarize





Types of attack





Passive attack

Active attacks

Techniques used to achieve the security goals are:

CRYPTOGRAPHY
(GENERALIZED)

STEGANOGRAPHY
(SPECIFIC)

What happens exactly?

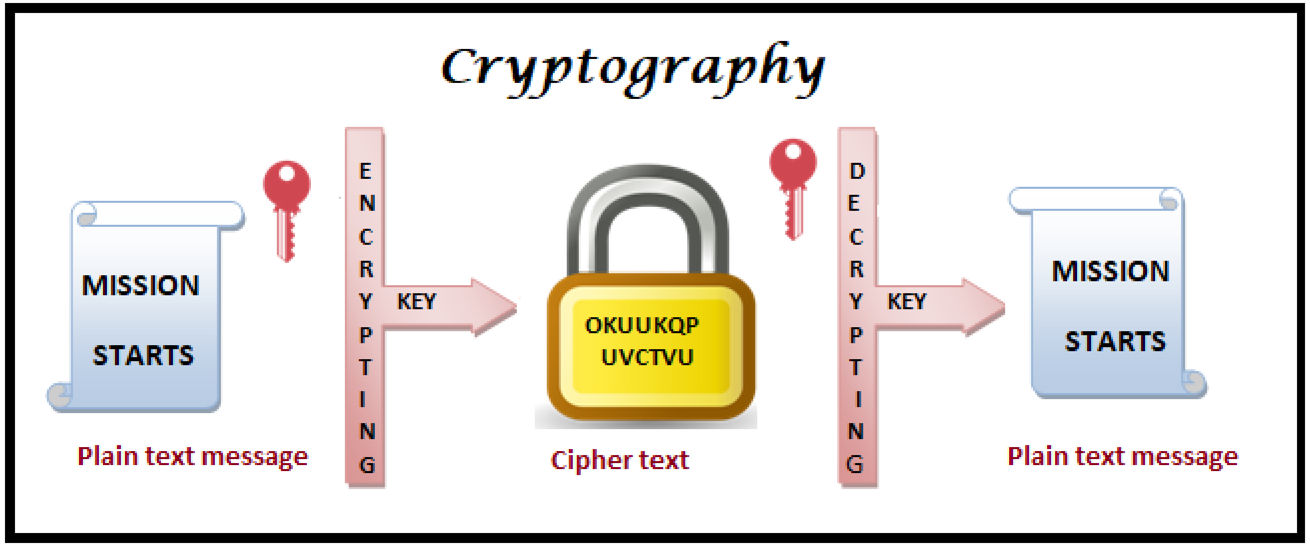
Plain text----- Hello everyone (easy to read)

Plain text----- Hello everyone

Transferred text---ciunhaphd (difficult to decrypt)

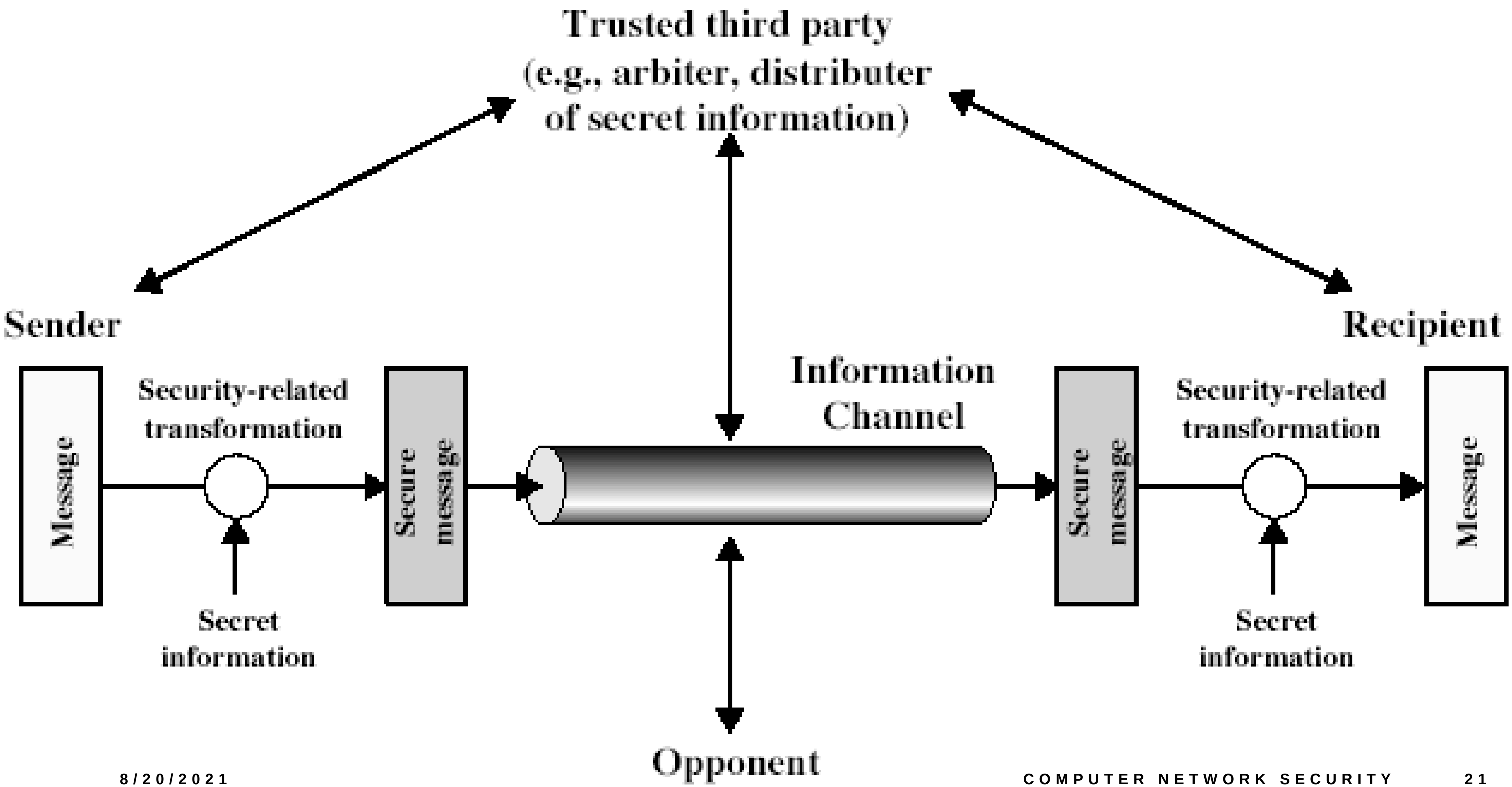
Decrypted text---- Hello everyone

So how to achieve this……..



Cryptography

- Some security mechanisms listed in the previous section can be implemented using cryptography. Cryptography, a word with Greek origin, means “secret writing”. However, we use the term to refer to the science and art of transforming messages to make them secure and immune to attacks. Although in the past cryptography referred only to the encryption and decryption of messages using secret keys, today it is defined as involving three distinct mechanisms: symmetric-key encipherment, asymmetric-key encipherment, and hashing.
- It is a method of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it. The prefix "crypt-" means "hidden" or "vault" -- and the suffix "-graphy" stands for "writing."
- Parts:
  - ❖ **Encryption**
  - ❖ **Decryption**


Trusted third party
(e.g., arbiter, distributer
of secret information)
Sender
Recipient
Message
Security-related
transformation
Secure
message
Information
Channel
Secure
message
Security-related
transformation
Message
Secret
information
Secret
information
Opponent

Techniques

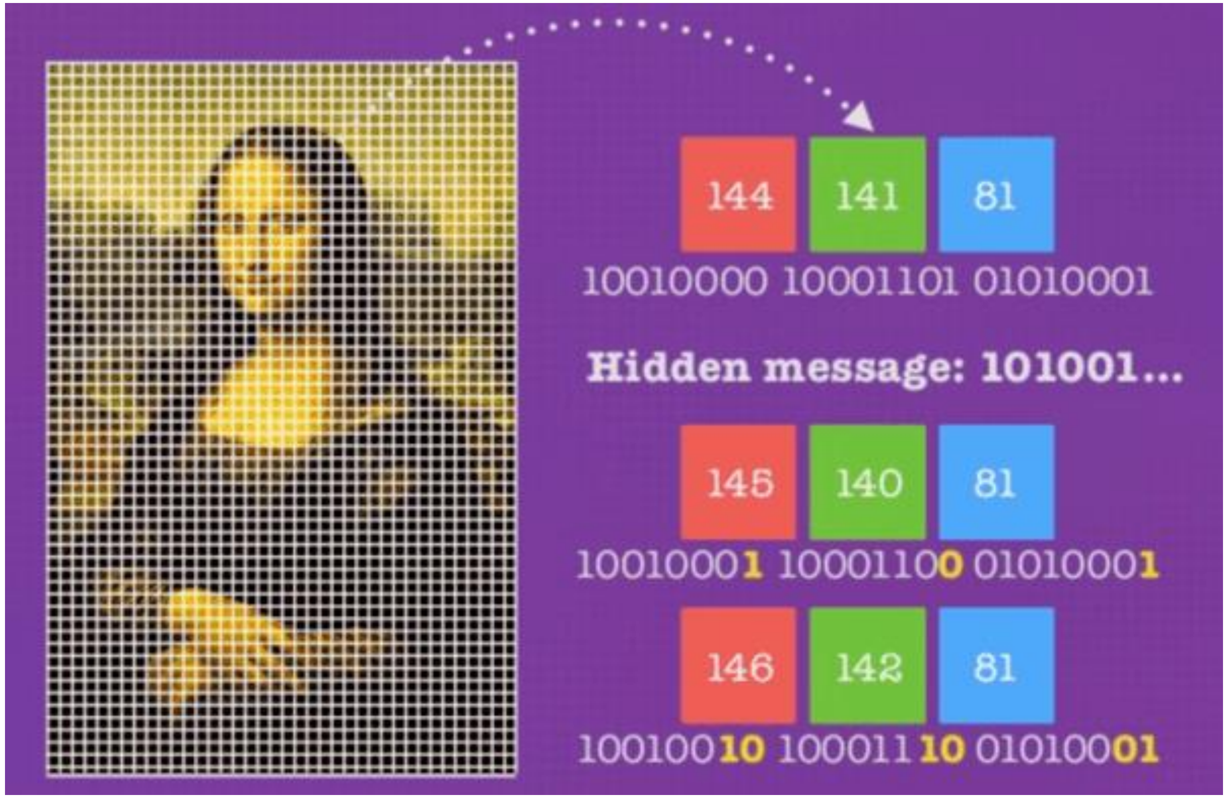
- In symmetric encipherment, an entity, say Alice, can send a message to other entity, say Bob, over an insecure channel with the assumption that an adversary, say Eve, cannot understand the contents of the message by simply eavesdropping over the channel. Alice encrypts the message using an encryption algorithm. Bob decrypts the message using a decryption algorithm. Symmetric-key encipherment uses a single secret key for both encryption and decryption. Encryption/decryption can be thought of as electronic locking system. In symmetric-key enciphering, Alice puts the message in a box and locks the box using the shared secret key; Bob unlocks the box with the same key and takes out the messages.

In asymmetric encipherment, we have the same situation as the symmetric-key encipherment, with a few exceptions. First, there are two keys instead of one; one public key and one private key. To send a secure message to Bob, Alice firsts encrypts the message using Bob's public key. To decrypt the message, Bob uses his own private key.

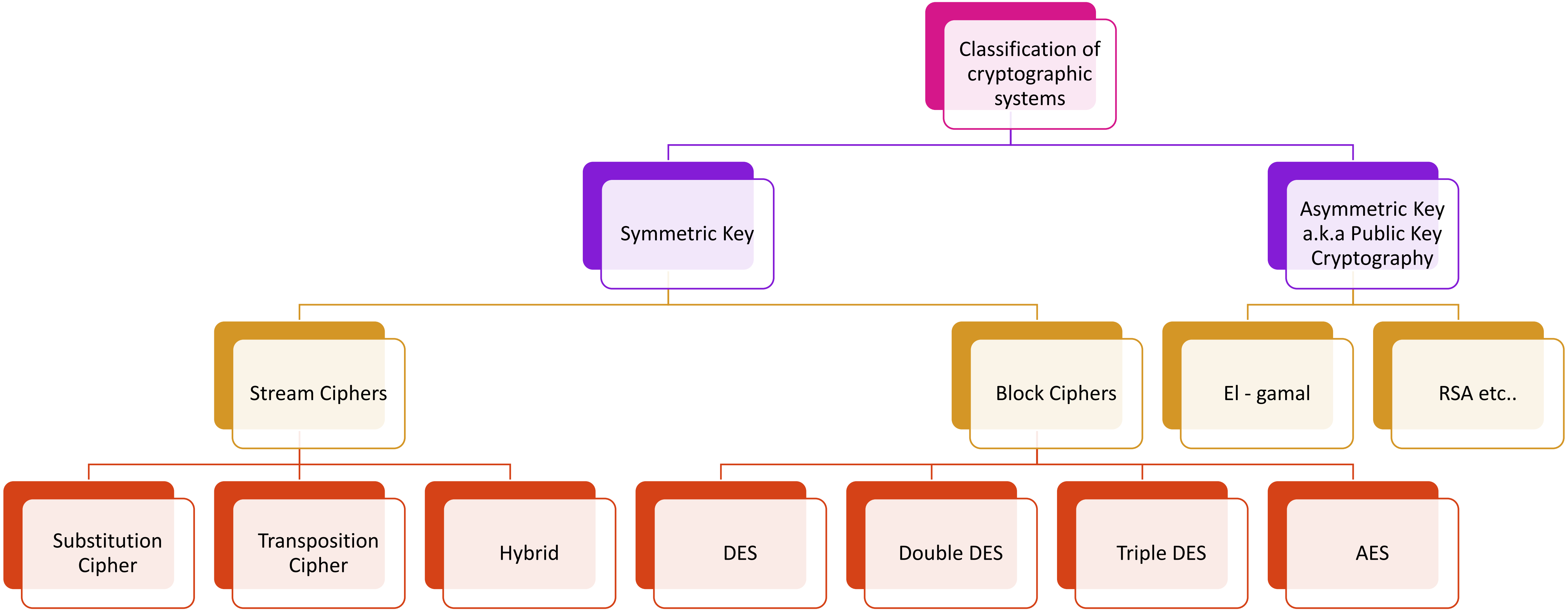
In hashing, a fixed-length message digest is created out of a variable-length message. The digest is normally much smaller than the message. To be useful, both the message and the digest must be sent to Bob. Hashing is used to provide check values

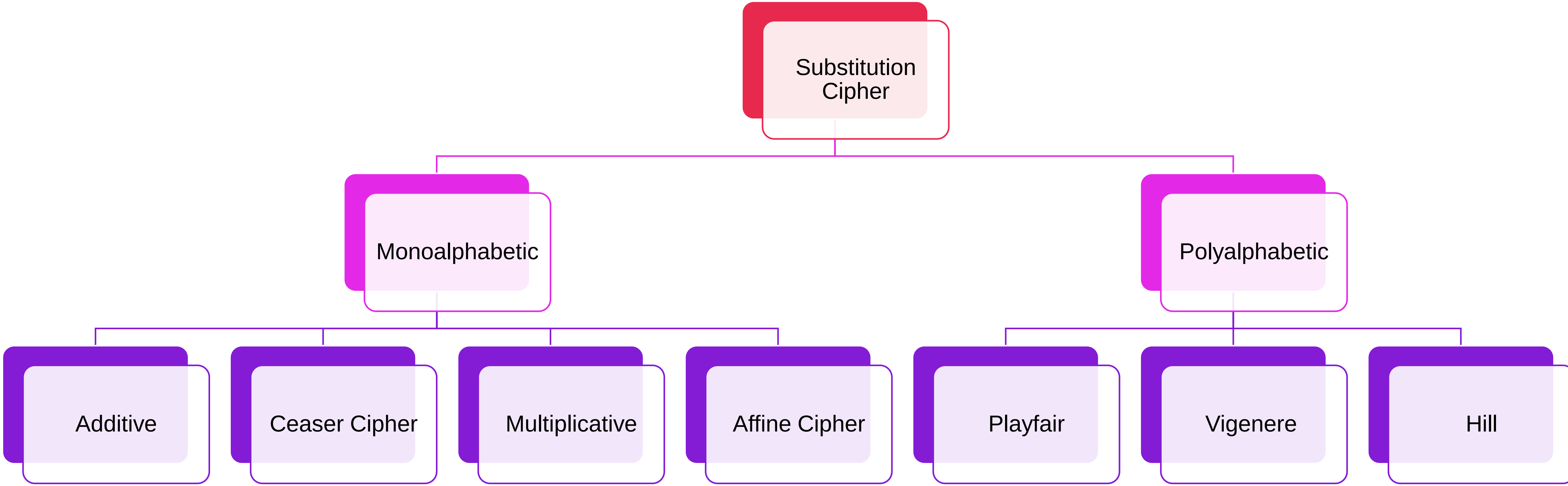
Steganography

This is the art of hiding messages in another form. Message is not altered as in encryption. A text can hide a message. For example “red umbrella needed” may mean the message “run”. The first letter of each word in the text becomes the message. An image can also be used for hiding messages. Digital images are after all binary information. Suppose the image is grey image. The least significant bit of consecutive eight pixels may be altered to be a specific bit pattern of a character. We will discuss this technique of steganography in detail in the unit to come.



Classification of cryptographic system




Substitution Cipher
Monoalphabetic
Polyalphabetic
Additive
Ceaser Cipher
Multiplicative
Affine Cipher
Playfair
Vigenere
Hill

<u>Let us understand each method in brief……</u>

SYMMETRIC CIPHER MODEL

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.

Decryption algorithm

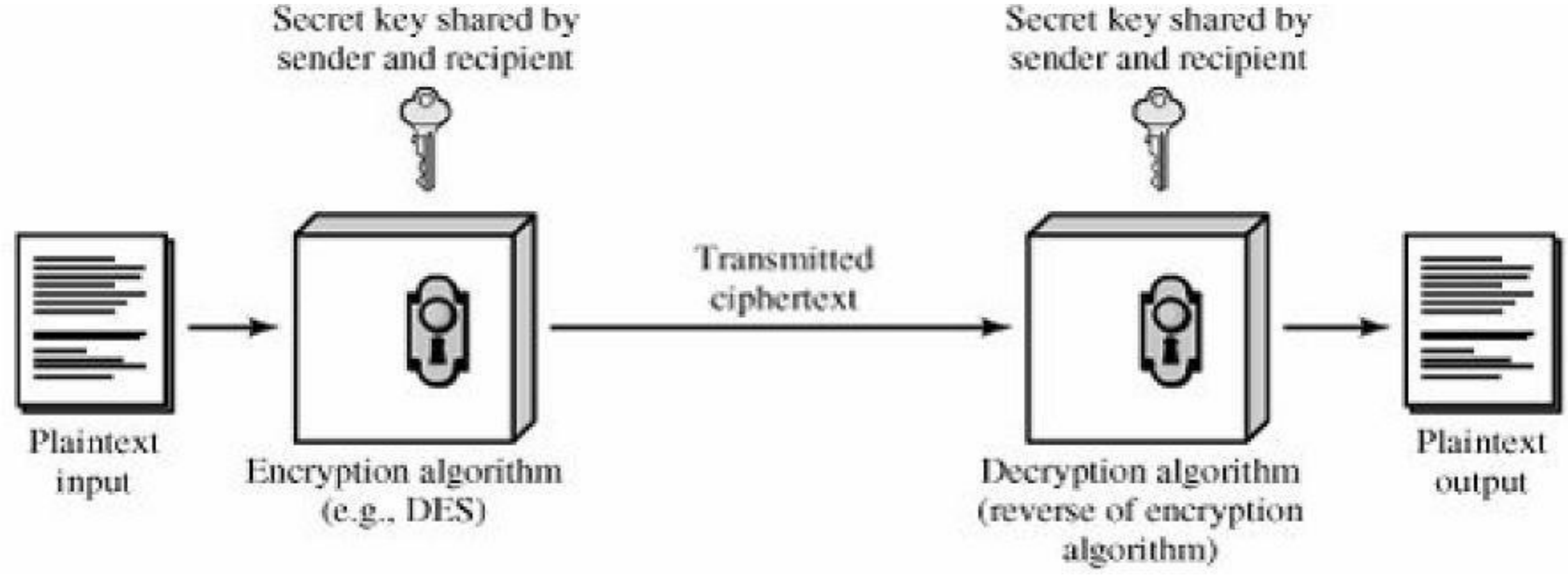
- This is essentially the encryption algorithm run in reverse. It takes the cipher text and the secret key and produces the original plaintext

Cipher text

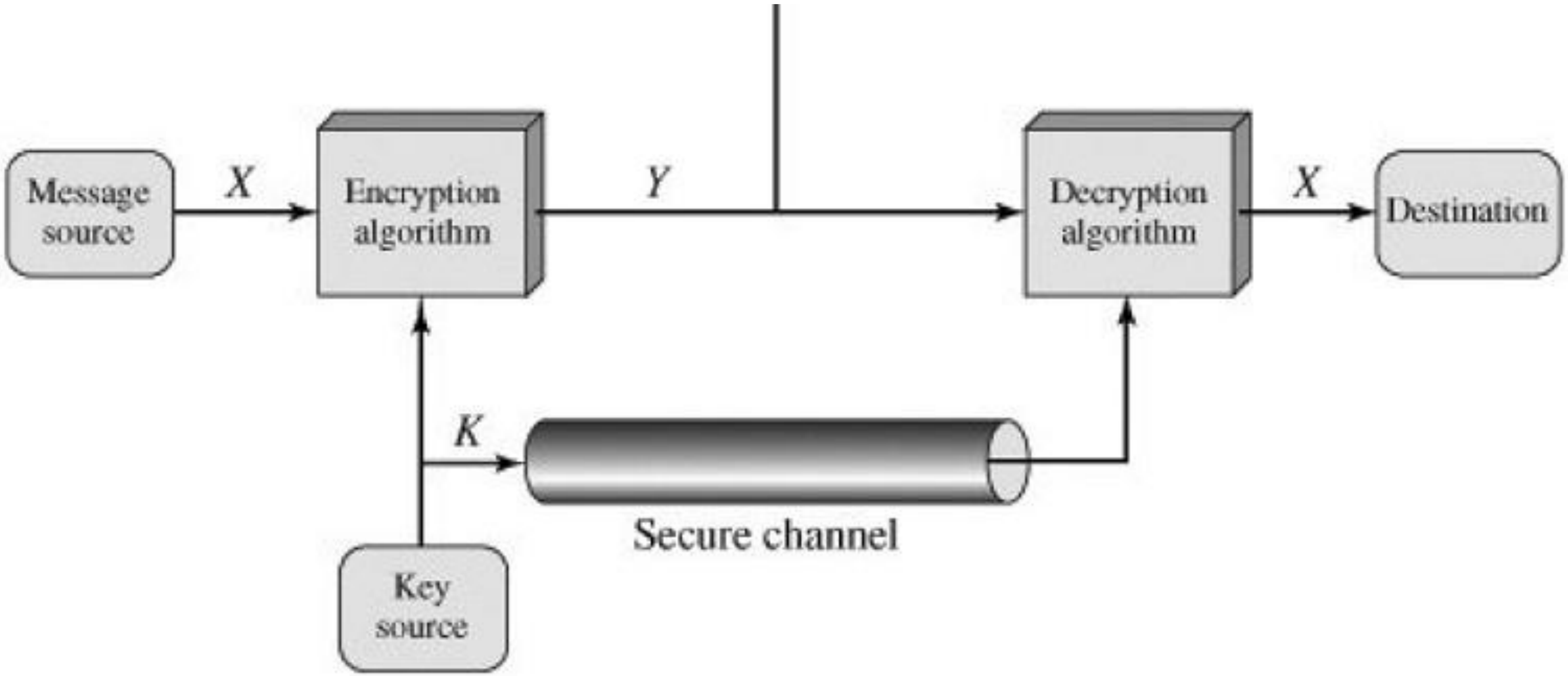
- This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different cipher texts. The cipher text is an apparently random stream of data and, as it stands, is unintelligible.

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


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

- Let us take a closer look at the essential elements of a symmetric encryption scheme, using Figure 3.2. 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. Nowadays, the binary alphabet {0, 1} is typically used. 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 cipher text **Y= [Y1, Y2, ..., YN].**
- We 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 using decryption algorithm and the secret key.
- We write this as **X= D (K, Y)**


Message source
X
Encryption algorithm
Y
Decryption algorithm
X
Destination
K
Secure channel
Key source

Additive Cipher

ENCRYPTION

- Plain Text (P)
- Key (K)
- Cipher (C) = (P+K) mod 26

DECRYPTION

- Cipher
- Key (K)
- Plain Text (P) = (P-K) mod 26

E.g., P = Hello Everyone
K = 7

P = Hello Everyone

K = T

Char	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
Int	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

Step 1: Convert P and K to int

h	e	l	l	o	e	v	e	r	y	o	n	e
7	4	11	11	14	4	21	4	17	24	14	13	4

K = 19

Step 2: (p+k) MOD 26

How to calculate the cipher values is by performing modular function

Text == p

Key == k

Total characters are 26

So ciphered text will be denoted as X

X == (p+k) mod 26

P+k = a

26 = b

So X = a mod b

= a – (int (a/b) * b)

For ex: p = 7 , k = 19, calculate X

X = (7+19) mod 26 == 26 mod 26 == 26 – (int (26/26) * 26) == 26 – (1 * 26) == 26 – 26 == 0

X == 0

h	e	l	l	o	e	v	e	r	y	o	n	e
7	4	11	11	14	4	21	4	17	24	14	13	4
0	23	4	4	7	23	14	23	10	17	7	6	23

Step 3 : Convert the int back to char

Char	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
Int	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

0	23	4	4	7	23	14	23	10	17	7	6	23
a	x	e	e	h	x	o	x	k	r	h	g	x

Cipher :axeehxoxkrhgx

Cipher :axeehxoxkrhgx   K = 19

Decryption

Char	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
Int	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

a	x	e	e	h	x	o	x	k	r	h	g	X
0	23	4	4	7	23	14	23	10	17	7	6	23

P = (C-K) mod 26

- **-19 mod 26 = 7**

a:- (0-19)mod26

-19mod26

-19+26 = 7 : - **h**

We have: a (dividend) = -19, b (divisor) = 26

a mod b = a – (Floor [a / b] × b) , where **Floor** is the round down integer function

-19 mod 26 = -19 – (Floor (-19 / 26) × 26)

-19 mod 26 = -19 – (-1 × 26)

-19 mod 26 = -19 – (-26) = 7

-19 mod 26 = 7

Lets code:

```
import java.io.*;

public class Adder {

static int a[] = new int[26];
static char c[] = new char[]{'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'};

//static char c[] = new char[26];
    static String str1 = "", s = "";

    public static void main(String[] args) throws IOException {
        for (int i = 0; i < 26; i++) {
            a[i] = i;
        }
    InputStreamReader isr = new InputStreamReader(System.in);
            BufferedReader br = new BufferedReader(isr);
            System.out.println("ENTER PLAINTEXT: ");
            s = br.readLine();
            System.out.println("ENTER KEY VALUE: ");
            int k = Integer.parseInt(br.readLine());
```

```java
for (int i = 0; i < s.length(); i++) {
            int j;
            if (s.charAt(i) == ' ') {
                i++;
                str1 = str1.concat(" ");
            }
 for (j = 0; j < 26; j++) {
                if (s.charAt(i) == c[j]) {
                    break;
                }
            }
            j = (a[j] + k) % 26;
            str1 = str1.concat(c[j] + "");
        }
System.out.println("AFTER ENCRYPTION: ");
        System.out.println(str1);
        s = "";
        for (int i = 0; i < str1.length(); i++) {
            int j;
            if (str1.charAt(i) == ' ') {
                i++;
                s = s.concat(" ");
            }
for (j = 0; j < 26; j++) {
                if (str1.charAt(i) == c[j]) {
                    break;
                }
            }
            j = (a[j] - k) % 26;
            if (j < 0) {
                j = 26 + j;
            }
            s = s.concat(c[j] + "");
        }
        System.out.println("AFTER DECRYPTION: ");
        System.out.println(s);

    }

}
```

```
ENTER PLAINTEXT:
hey lets meet at ccd
ENTER KEY VALUE:
8
AFTER ENCRYPTION:
pmg tmba ummb ib kkl
AFTER DECRYPTION:
hey lets meet at ccd
```

Sample output

Perform additive ciphering on the text mentioned below:

Text to be sent is P == welcome to engineering

Hidden key K= u

(perform using Java/Python)

Submit on :

https://tinyurl.com/Additivecipher

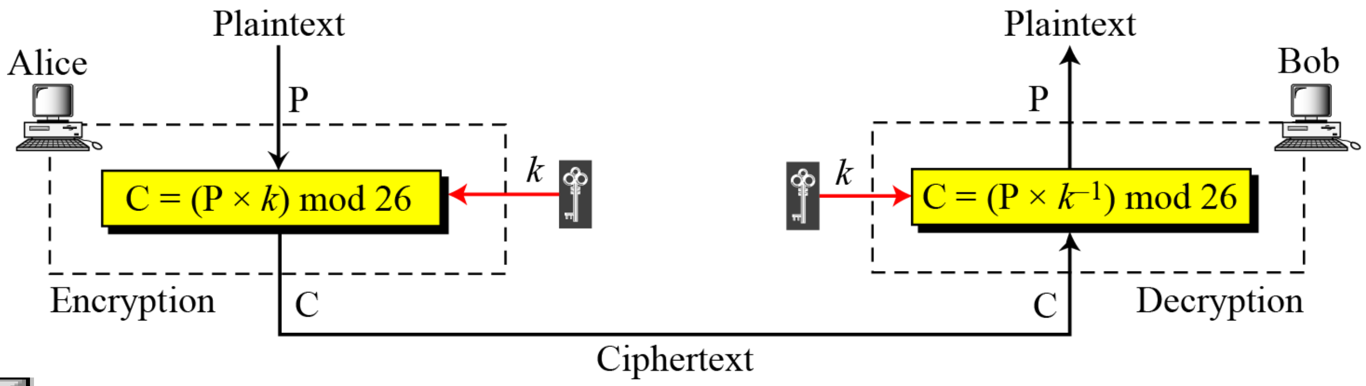
Next lecture

- Multiplicative cipher
- Affine cipher

Multiplicative cipher

Multiplicative cipher

- Similar to additive ciphering
- Instead of additive, multiplication is performed.



Note

In a multiplicative cipher, the plaintext and ciphertext are integers in Z_{26}; the key is an integer in Z_{26}^*.

Cont.….

FOR ENCRYPTION:

P == PLAIN TEXT

K == KEY

C == CIPHER TEXT

C == (P*K) MOD 26

FOR DECRYPTION:

C == CIPHER TEXT

P == PLAIN TEXT

k^{-1} == INVERSE KEY

P == (C * k^{-1}) MODE 26

EXAMPLE:

We use a multiplicative cipher to encrypt the message “hello” with a key of 7. The ciphertext is “XCZZU”.

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

DECRYPTION:

CALCULATE k^{-1}

Euclidean distance algorithm for calculating Inverse

Lets use Simple method: t = t1 – t2*q……………….

q	r1	r2	r	t1	t2	t
3	26	7	5	0	1	-3
1	7	5	2	1	-3	4
2	5	2	1	-3	4	-11
2	2	1	0	4	-11	26
	1	0		-11	26	

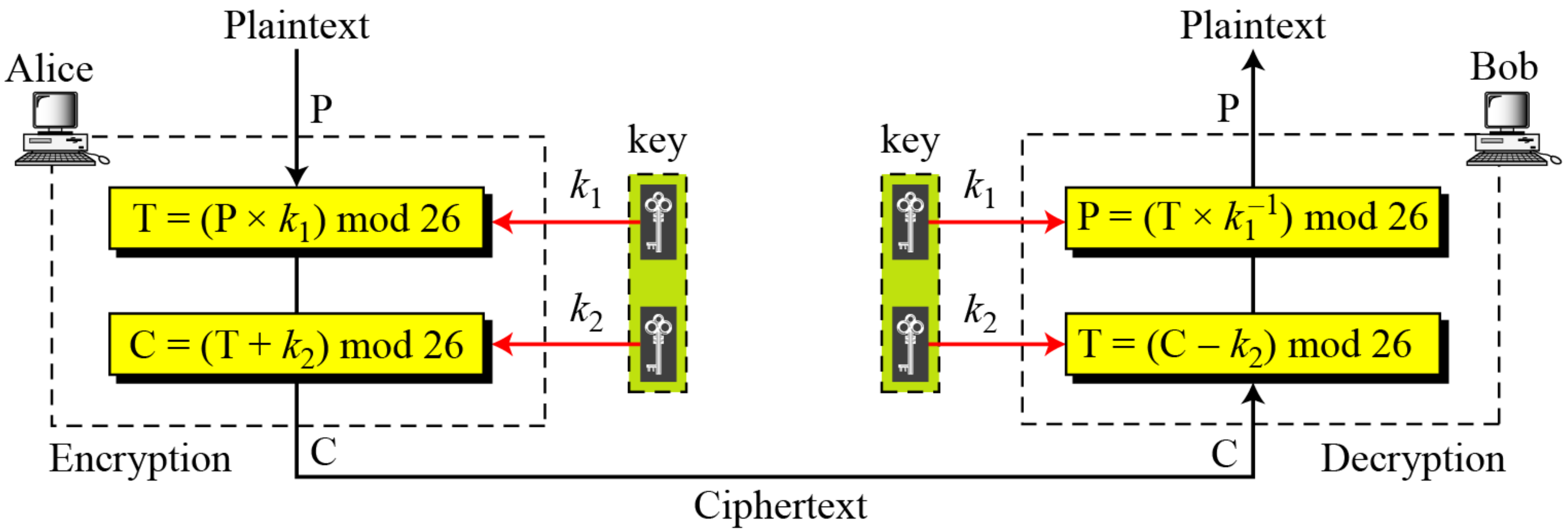
$\boxed{k^{-1} = -11 \text{ or } 26 + (-11) = 15}$

Cont.….

- So now k^{-1} is 15….
- Start decrypting each letter now
- Cipher text is “XCZZU”

Perform encryption and decryption using Multiplicative ciphering on Data: welcome, use key as 11.

Affine cipher



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

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

Affine Cipher (Additive + Multiplicative)

- Encryption
- Plain Text (P)
- Key (a,b)
- Cipher (C) = (P*a + b) mod 26

E.g. P = Hello Everyone
K = (11,3)

Example:

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

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

Decryption:

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

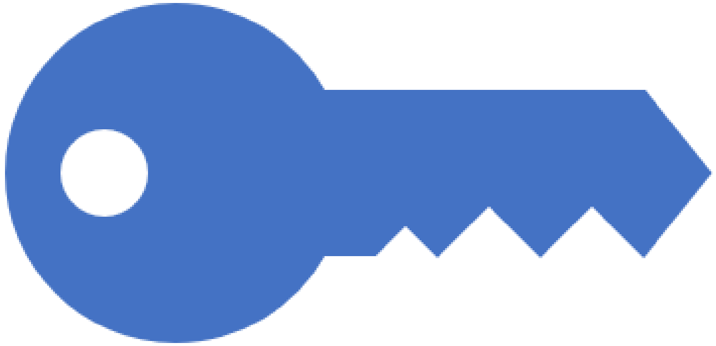
Solution

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

Solve:

Encrypt and decrypt the data word NAME using keys (3, 7) use Affine Cipher technique

Ceaser Cipher



- Same as Additive Cipher where K is fixed to 3

- YASH SHAH

Caesar Cipher Technique

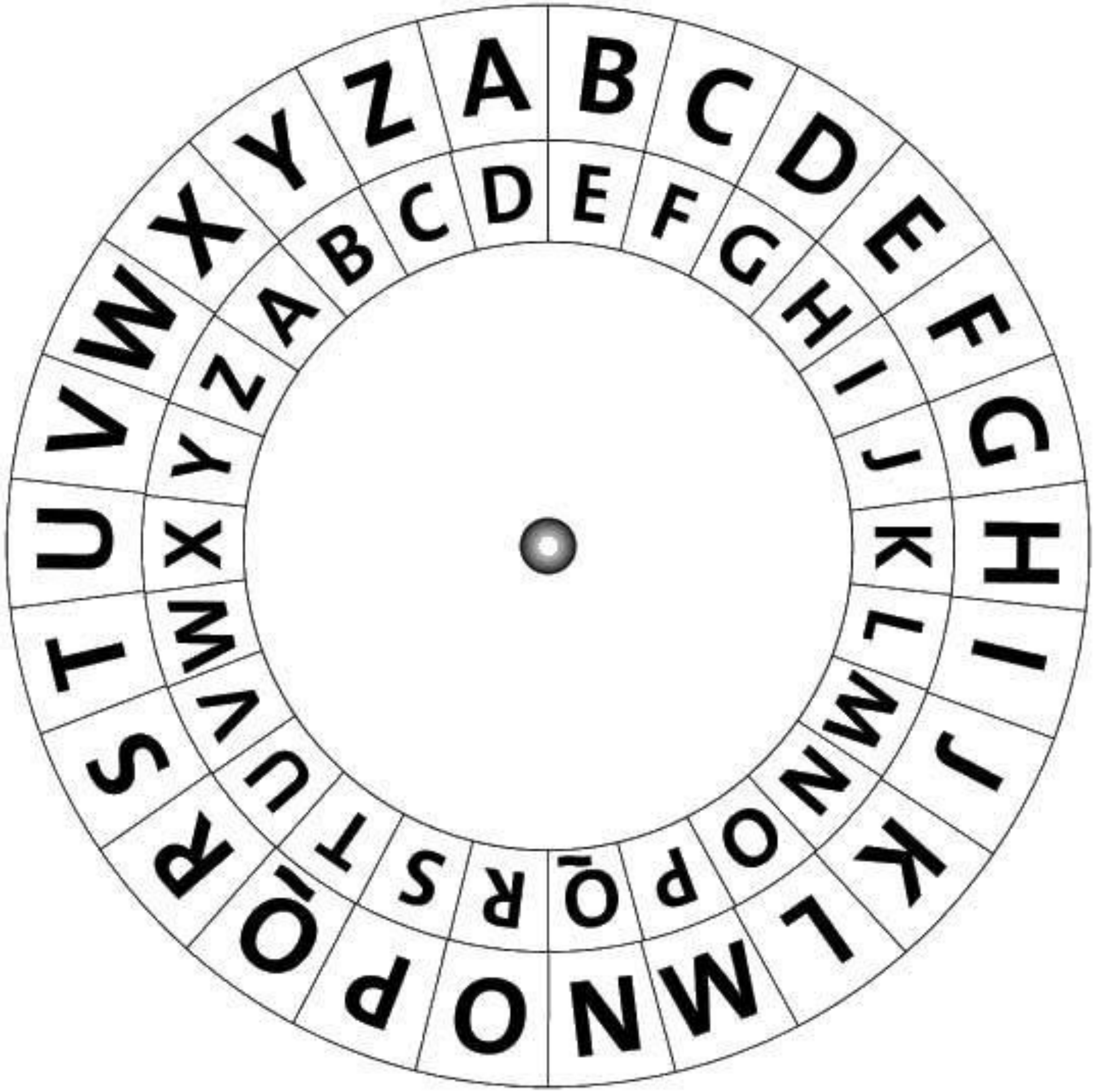
- The Caesar cipher is the simplest and oldest method of cryptography. The Caesar cipher method is based on a mono-alphabetic cipher and is also called a shift cipher or additive cipher. Julius Caesar used the shift cipher (additive cipher) technique to communicate with his officers. For this reason, the shift cipher technique is called the Caesar cipher. The Caesar cipher is a kind of replacement (substitution) cipher, where all letter of plain text is replaced by another letter.
- Let's take an example to understand the Caesar cipher, suppose we are shifting with 1, then A will be replaced by B, B will be replaced by C, C will be replaced by D, D will be replaced by C, and this process continues until the entire plain text is finished.
- Caesar ciphers is a weak method of cryptography. It can be easily hacked. It means the message encrypted by this method can be easily decrypted.
- **Plaintext:** It is a simple message written by the user.

Ceasar cipher

- Also called as shift cipher
- Each letter in plaintext is replaced by corresponding letter to the number of shifts in the alphabet
- Simple ceasar cipher uses key as k+3

			A	B	C	D			
A	B	C	D	E	F	G	H	I	J

- I Love my college-----→



CONT….

- Can use left shift or right shift
- Can represent as:

E(m) = (m+k) mod 26

Where m is message and k Is key

D(m) = (m-k) mod 26

Where m is message and k Is key

If any case (Dn) value becomes negative (-ve), in this case, we will add 26 in the negative value.

Example:

Encrypt data **Engineering** using k set to **3**

Advantages of Caesar cipher

1. It is very easy to implement.
2. This method is the simplest method of cryptography.
3. Only one short key is used in its entire process.
4. If a system does not use complex coding techniques, it is the best method for it.
5. It requires only a few computing resources.

Disadvantages of Caesar cipher

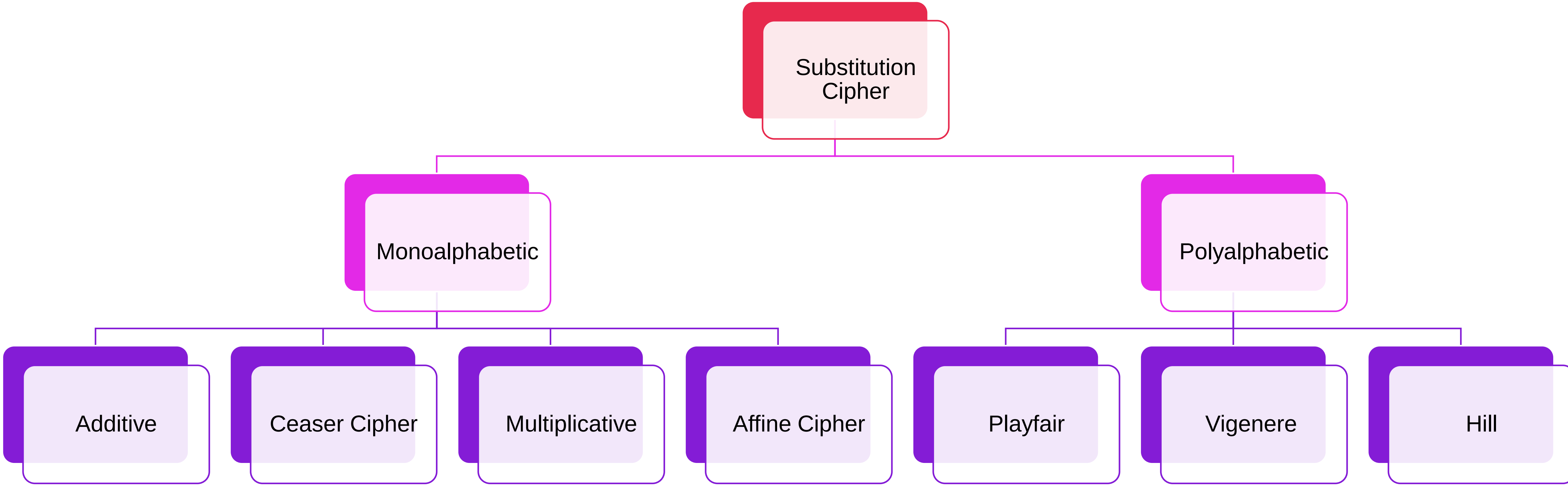
1. It can be easily hacked. It means the message encrypted by this method can be easily decrypted.
2. It provides very little security.
3. By looking at the pattern of letters in it, the entire message can be decrypted.

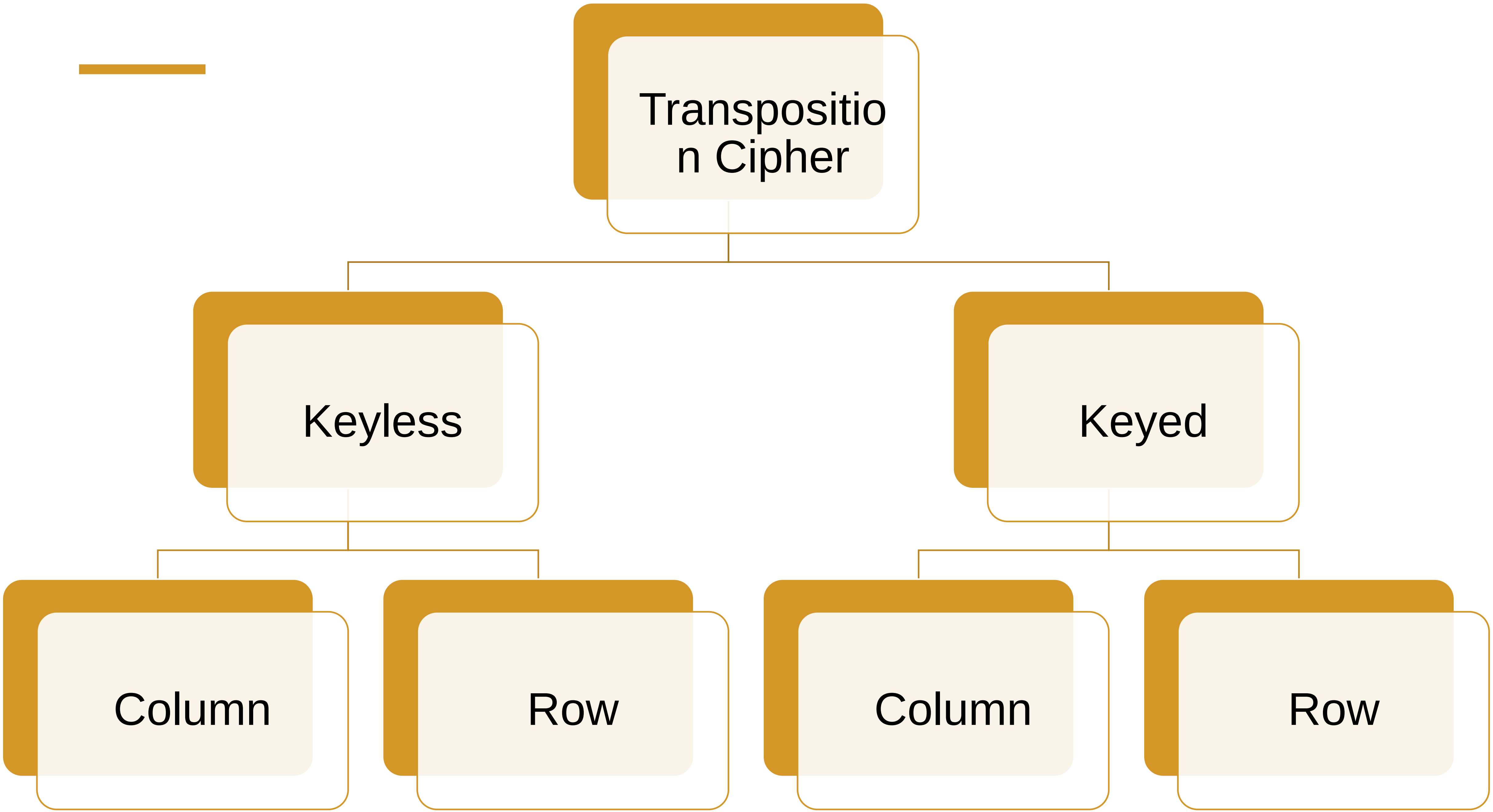
Example

Encrypt and Decrypt data using ceasar cipher method:

Plain text = do not disclose data

K == 3


Substitution Cipher
Monoalphabetic
Polyalphabetic
Additive
Ceaser Cipher
Multiplicative
Affine Cipher
Playfair
Vigenere
Hill


Transposition Cipher
Keyless
Keyed
Column
Row
Column
Row

Encryption

Keyless Transposition(Column)

Decryption

P: Hello Everyone

N = 13 = 4 * 4 matrix

h	e	l	l
o	e	v	e
r	y	o	n
e	x	x	x

hore eey
x lovx lenx

horeeeyxlovxlen
x

C : horeeeyxlovxlenx

h	o	r	e
e	e	y	x
l	v	o	x
l	e	n	x

hell oev
e ryo
n exxx

Helloeveryonexx
x

Keyed Transposition

- P: Hello Everyone
- K: HACK = 3124

N = 4 (no of Columns)

3	1	2	4
h	e	l	l
o	e	v	e
r	y	o	n
e	x	x	x

eey x lvox hore lenx

eeyxlovxhorelen x

Decryption

K: HACK = 3124

3	1	2	4
E	L	H	L
E	V	O	E
Y	o	R	N
X	x	E	X

H	E	L	L
O	E	O	E
R	Y	V	N
E	X	x	X

- Keyless Transposition

  C: horeeeyxlovxlenx

  P:??

- Keyed Transposition

  C: eeyxlovxhorelenx

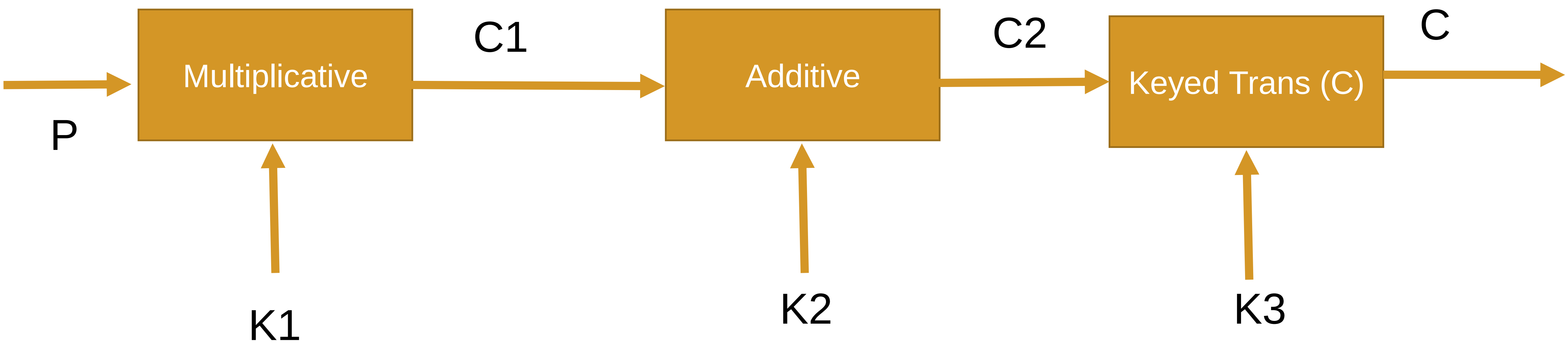
  K: HACK

Encrypt the following

$C = k3(k2(k1(P)))$

- P: Hello Everyone ; K1: 17; k2: 11; K3: VINI
- **C:??**

pqffeqtqdsenq **abqqpbebodpyb** **bbdxqbyxqepxapob**



End of Module 1