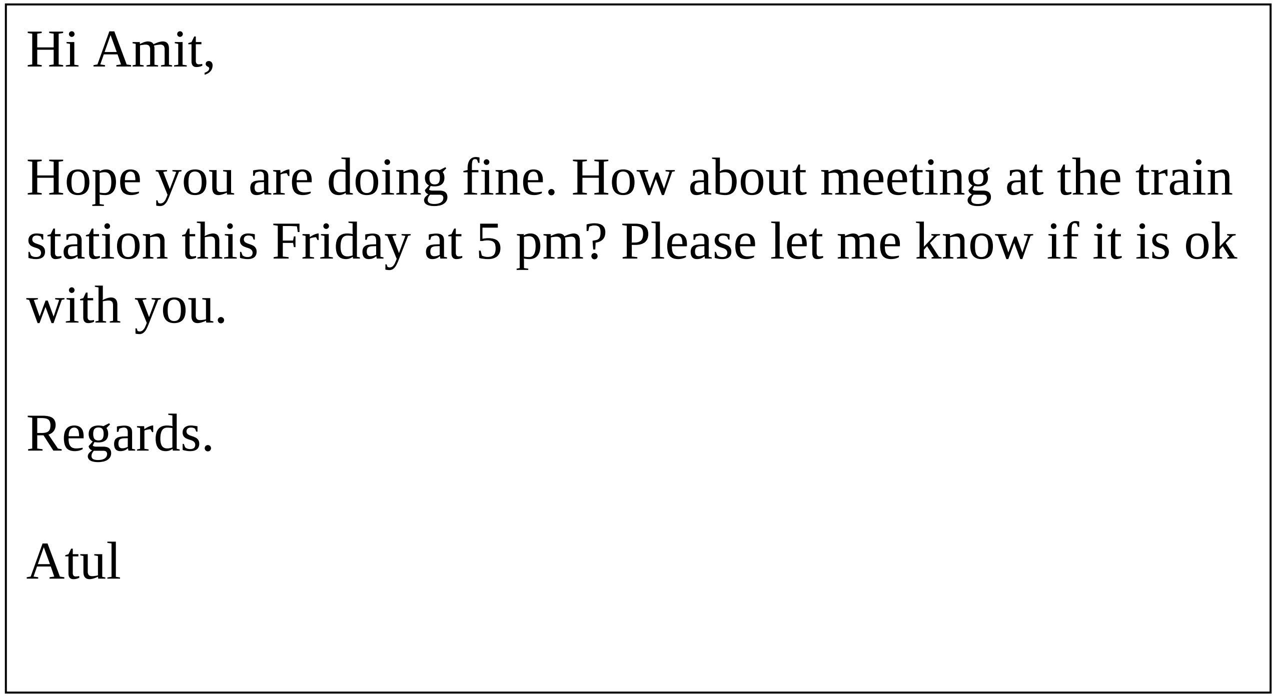
Plain Text

- Also called as *clear text*
- Language that we normally use
- Easily understood by everybody

Example of Plain Text Message



Hi Amit,

Hope you are doing fine. How about meeting at the train station this Friday at 5 pm? Please let me know if it is ok with you.

Regards.

Atul

Fig 2.1

Caesar Cipher

- Mechanism to make a message non-understandable
- Replaces each alphabet with the one three places down
- Example: Replace each A with D, B with E, etc.

Caesar Cipher

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
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

Fig 2.2

Plain Text and Cipher Text

- Plain Text: Language that can be easily understood
- Cipher Text: Language that cannot be understood
- To achieve security, plain text is transformed into cipher text

Plain Text and Cipher Text

Hi Amit, Hope you are doing fine. How about meeting at the train station this Friday at 5 pm? Please let me know if it is ok with you. Regards. Atul	**Kl Dplw,** **Krsh brx duh grlqj ilqh. Krz derxw phhwlqj dw wkh wudlq vwdwlrq wklv Iulgdb dw 5 sp? Sohdvh ohw ph nqrz li lw lv rn zlwk brx.** **Uhjdugv.** **Dwxo**
Plain text message	**Corresponding cipher text message**

Fig 2.4

Techniques for Plain Text to Cipher Text Conversion

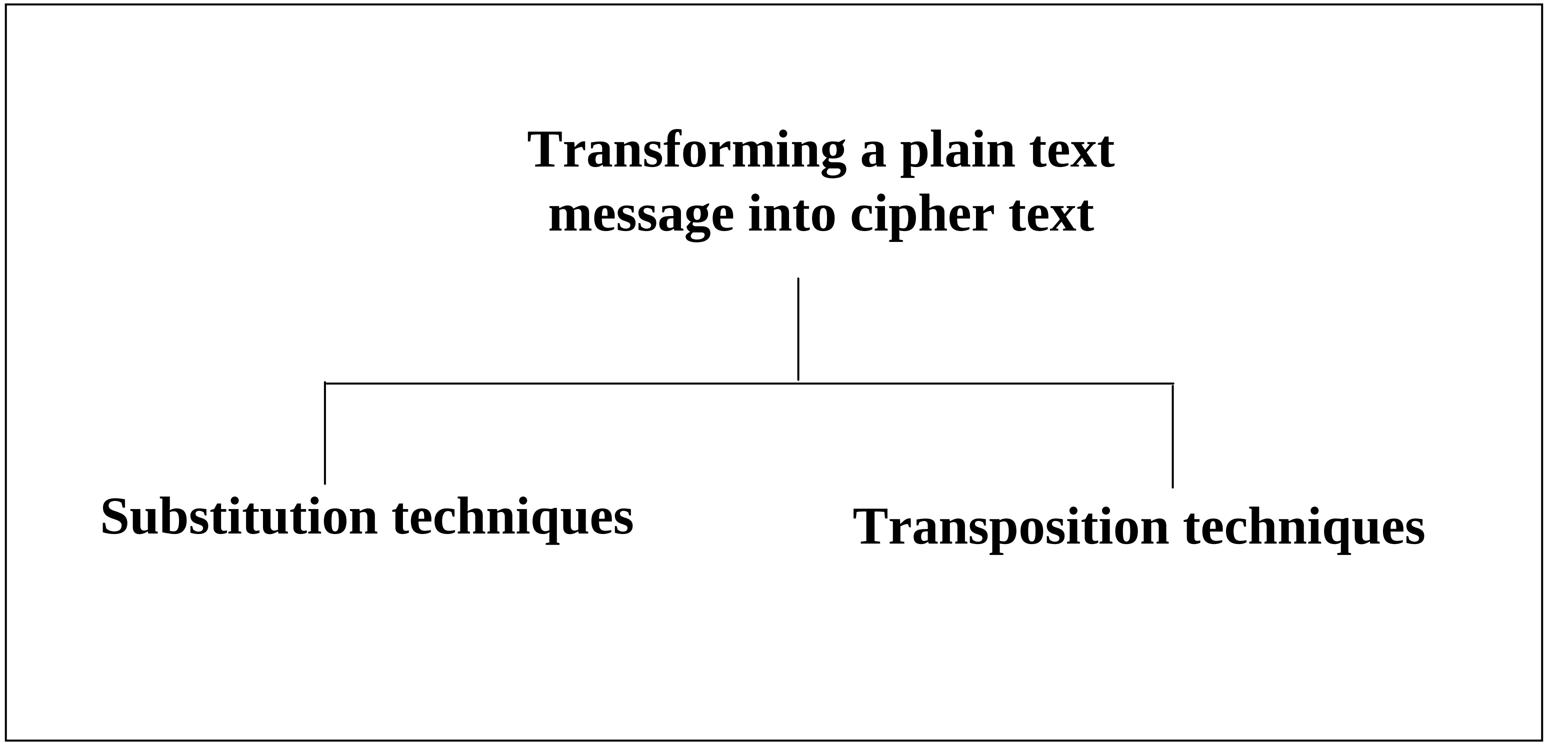


Fig 2.5

Algorithm to break Caesar Cipher

1. Read each alphabet in the cipher text message, and search for it in the second row of the Figure 2.2.

2. When a match is found, replace that alphabet in the cipher text message with the corresponding alphabet in the same column but the first row of the table (e.g. if the alphabet in cipher text is J, replace it with G).

3. Repeat the process for all alphabets in the cipher text message.

Fig 2.6

Polygram Substitution Cipher

- Block of plain text transformed into block of cipher text
- Similar text patterns can yield completely different cipher text patterns
- Block-by-block replacement

Polygram Substitution Cipher

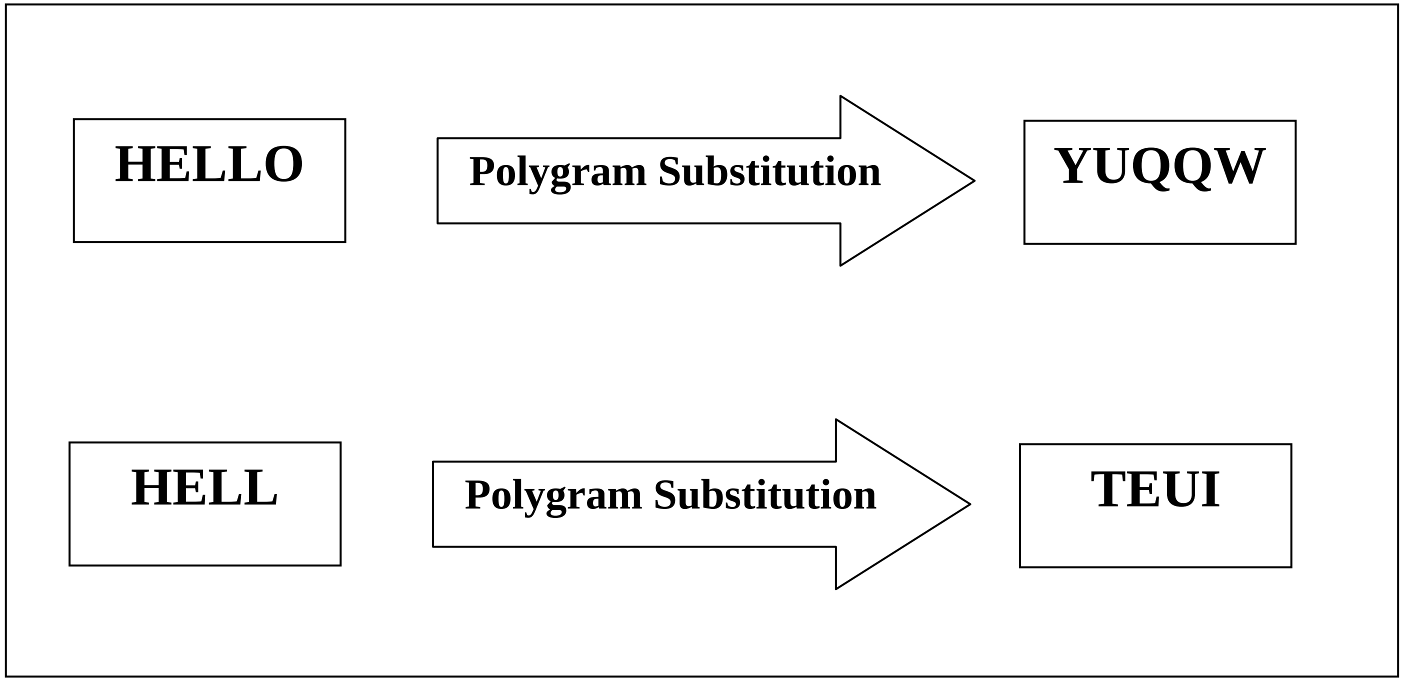


Fig 2.10

Rail Fence Technique

- Example of transposition cipher
- Write plain text as sequence of diagonals
- Read text as sequence of columns

Rail Fence Technique

1. Write down the plain text message as a sequence of diagonals.

2. Read the plain text written in *step 1* as a sequence of rows.

Fig 2.11

Simple Columnar Transposition

- Write text as rows
- Read the text so written as columns
- Result is cipher text

Simple Columnar Transposition Technique

1. Write the plain text message row-by-row in a rectangle of a pre-defined size.

2. Read the message column-by-column. However, it need not be in the order of columns 1, 2, 3 etc. It can be any random order such as 2, 3, 1, etc.

3. The message thus obtained is the cipher text message.

Fig 2.13

Simple Columnar Transposition Technique with Multiple Rounds

1. Write the plain text message row-by-row in a rectangle of a pre-defined size.

2. Read the message column-by-column. However, it need not be in the order of columns 1, 2, 3 etc. It can be any random order such as 2, 3, 1, etc.

3. The message thus obtained is the cipher text message of round 1.

4. Repeat steps 1 to 3 as many times as desired.

Fig 2.15

Vernam Cipher

- Also called as *One Time Pad*
- Use a random set of non-repeating input characters as cipher text
- Never used again (hence the name *one time*)

Vernam Cipher

1. Treat each plain text alphabet as a number in an increasing sequence, i.e. A = 0, B = 1, … Z = 25.

2. Do the same for each character of the input cipher text.

3. *Add* each number corresponding to the plain text alphabet to the corresponding input cipher text alphabet number.

4. If the sum thus produced is greater than 26, subtract 26 from it.

5. Translate each number of the sum back to the corresponding alphabet. This gives the output cipher text.

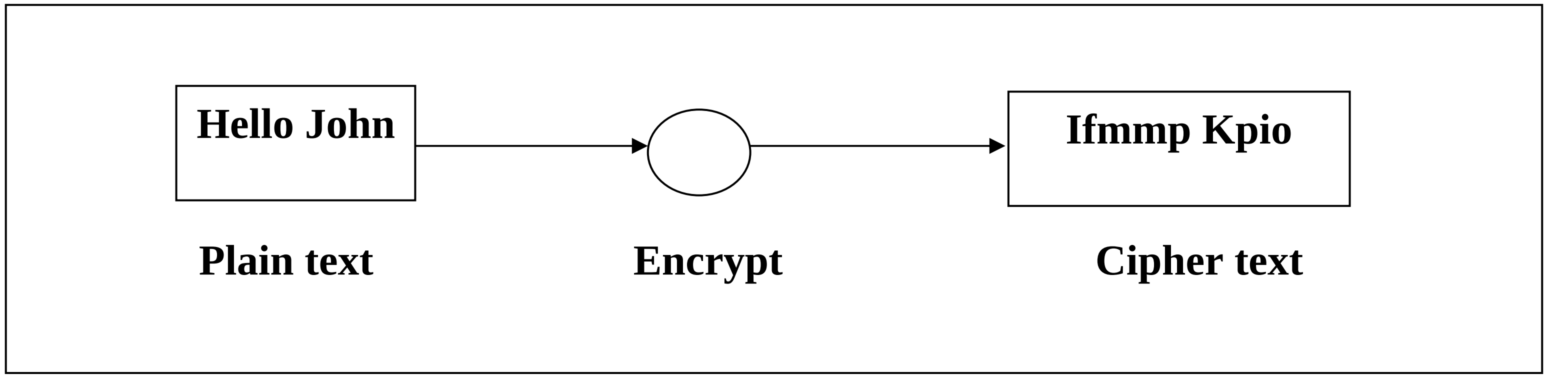
Fig 2.17

Encryption and Decryption

- Encryption
  - Conversion of Plain Text to Cipher Text

- Decryption
  - Conversion of Cipher Text to Plain Text

Encryption and Decryption



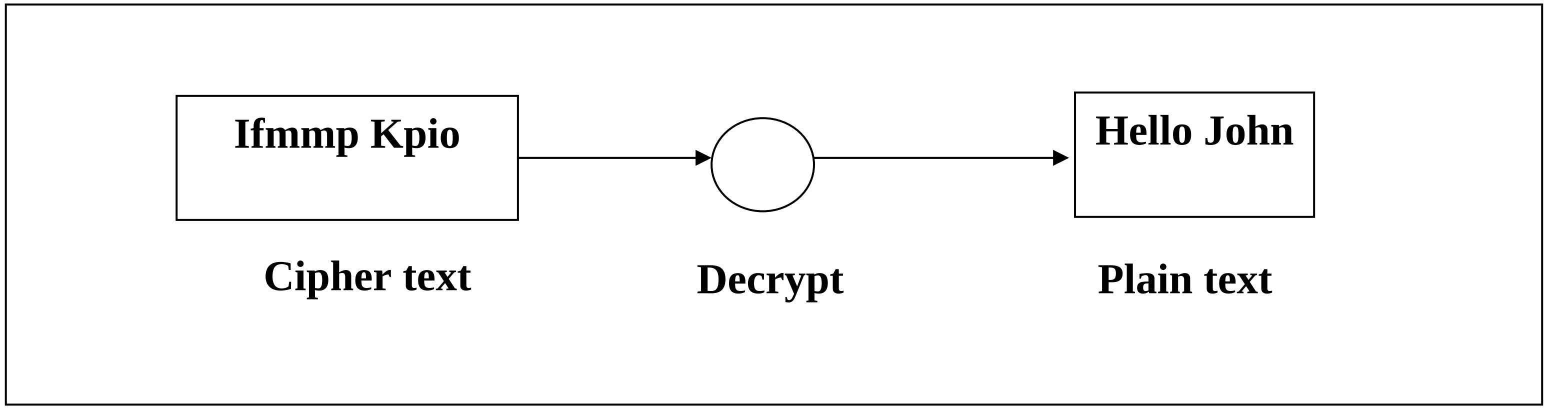


Fig 2.19, 2.20

Encryption and Decryption

- Two aspects related to this process:
  - Algorithm
  - Key

Aspects of Encryption and Decryption

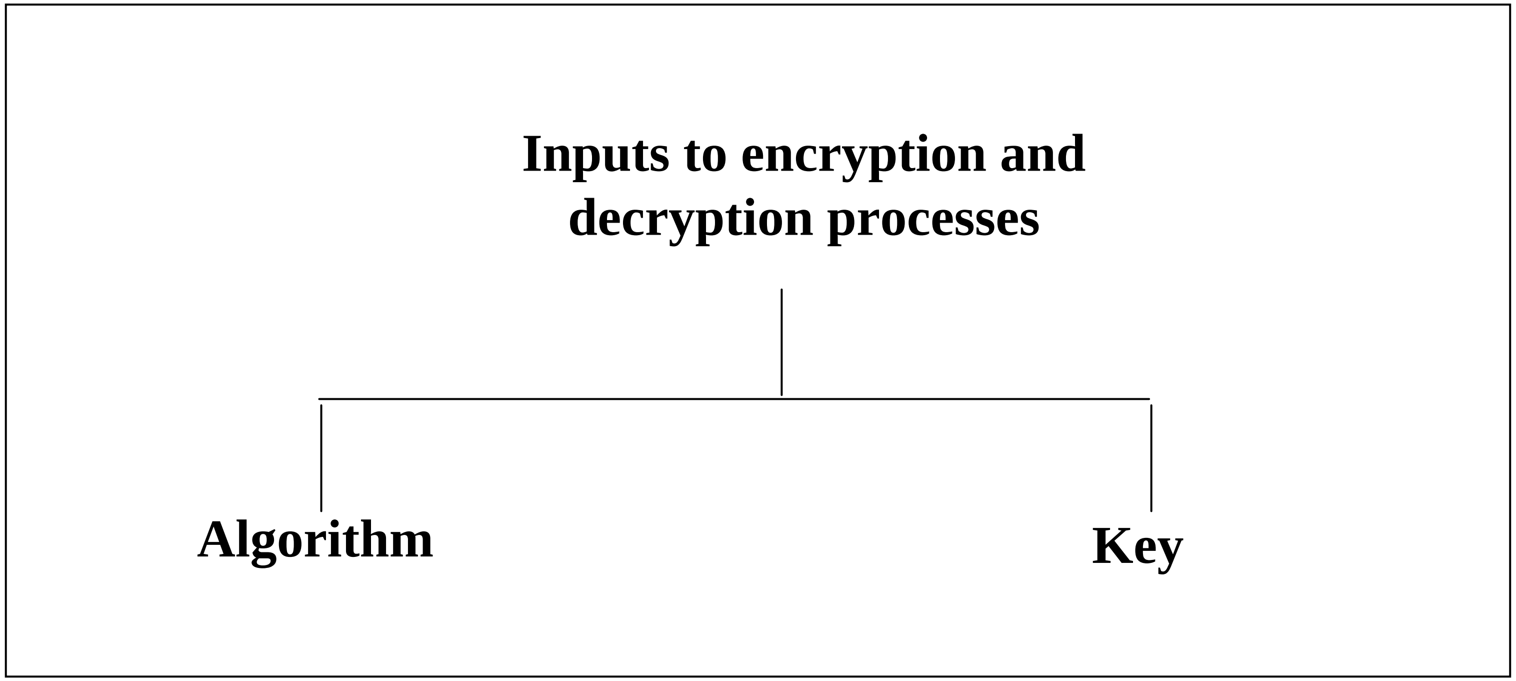


Fig 2.22

Types of Cryptography

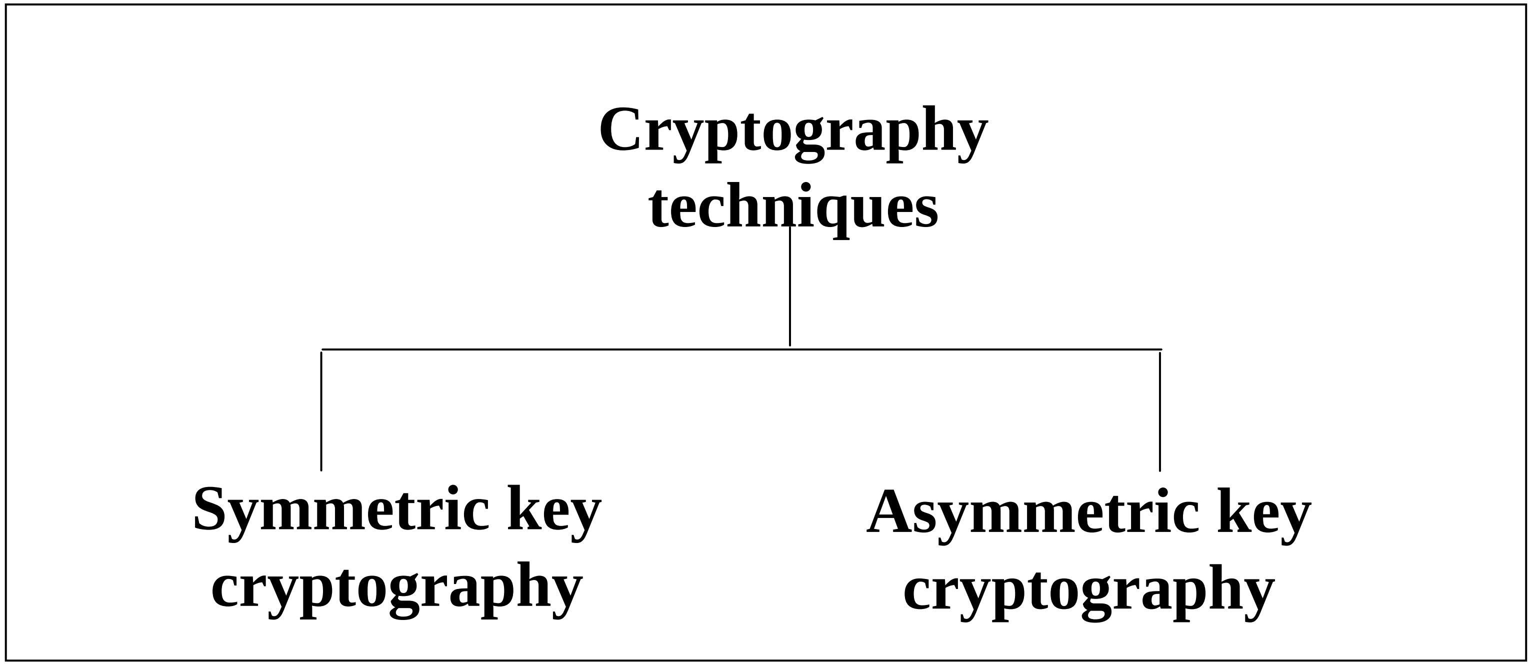


Fig 2.23

Diffie-Hellman Key Exchange

- Solves the problem of *Key Exchange*
- Alice and Bob can decide upon a key without meeting
- No secrets are exchanged, and yet a secret key can be agreed upon

Diffie-Hellman Key Exchange

1. Firstly, Alice and Bob agree on two large prime numbers, n and g. These two integers need not be kept secret. Alice and Bob can use an insecure channel to agree on them.

2. Alice chooses another large random number x, and calculates A such that:
$A = g^x \bmod n$

3. Alice sends the number A to Bob.

4. Bob independently chooses another large random integer y and calculates B such that:
$B = g^y \bmod n$

5. Bob sends the number B to Alice.

6. A now computes the secret key K1 as follows:
$K1 = B^x \bmod n$

7. B now computes the secret key K2 as follows:
$K2 = A^y \bmod n$

Fig 2.27

Brute Force Attack

- Attacker tries all possible keys one by one
- Can be successful if key length is small
- Start with Key = 0, then Key = 1, etc.

Brute Force Attack

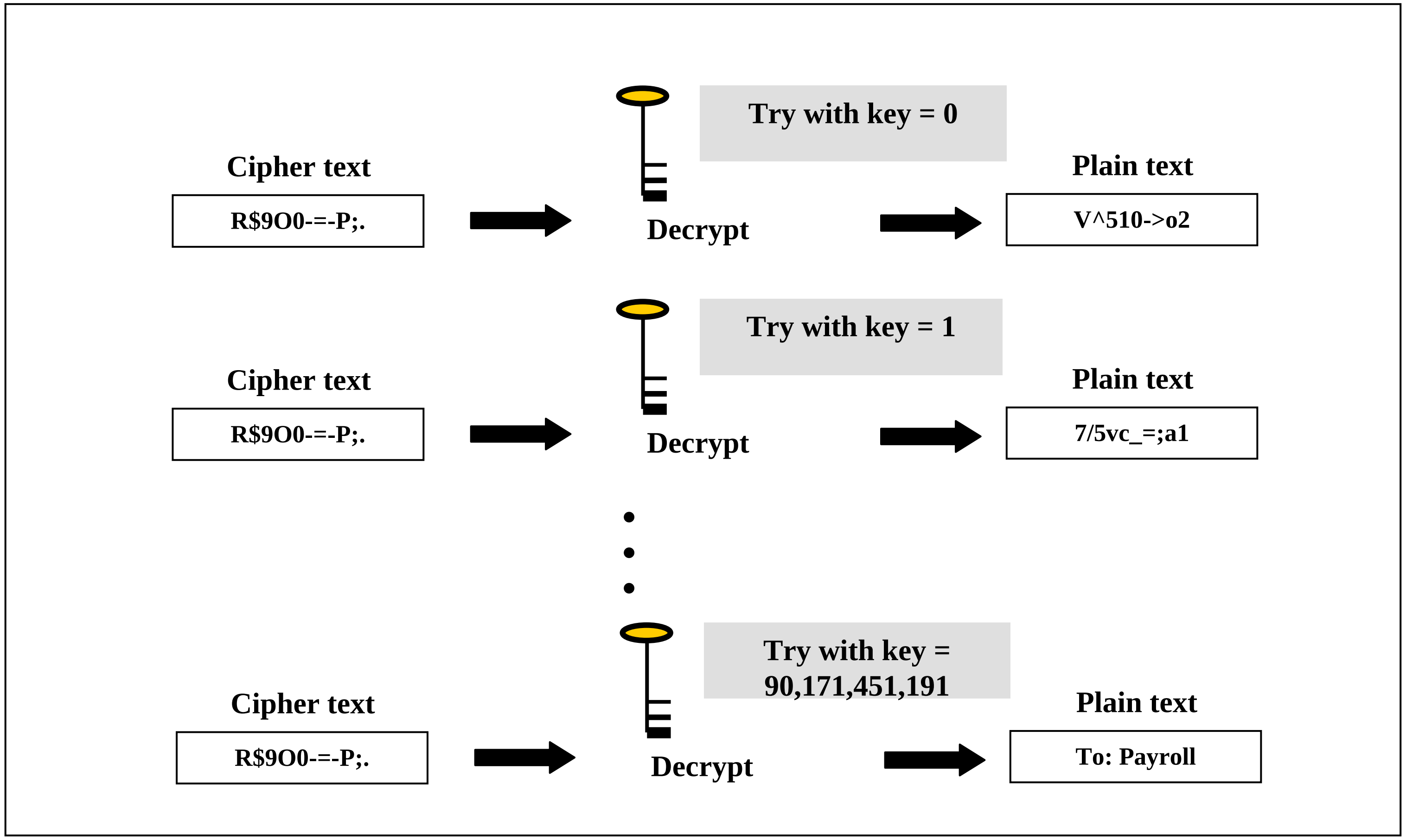


Fig 2.37

Key Range

- Specifies the number of possible keys
- Bigger the key range, more difficult is the attack
- In practice, at least 64, 128, 256 bit keys are used

Key Range

A 2-bit binary number has four possible states:
00
01
10
11

If we have one more bit to make it a 3-bit binary number, the number of possible states also doubles to eight, as follows:
000
001
010
011
100
101
110
111

In general, if an *n bit* binary number has *k* possible states, an *n+1 bit* binary number will have *2k* possible states.

Fig 2.38

Key Sizes and Range

Key size = 40 bits

00 00 00 00 00
00 00 00 00 01
…
FF FF FF FF FF

Key size = 64 bits

00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 01
…
FF FF FF FF FF FF FF FF

Key size = 128 bits

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01
…
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Fig 2.40