Computer Security Technology and Principles

Cryptographic Tools

Cryptographic Tools

- ❑ Introduction
  - ❖ Definitions
  - ❖ Cryptography applications
  - ❖ Classification of Cryptography
- ❑ Symmetric Encryption
  - ❖ Introduction to Symmetric Encryption
  - ❖ Symmetric Block Encryption Algorithms
  - ❖ Symmetric Stream Ciphers
  - ❖ Symmetric Encryption Algorithms
- ❑ Public-Key Encryption
  - ❖ Public-Key Encryption Structure
  - ❖ Applications for Public-Key Cryptosystems
  - ❖ Requirements for Public-Key Cryptography
  - ❖ Asymmetric Encryption Algorithms

Cryptographic Tools

- ❑ Hybrid Encryption
- ❑ Authentication and Hash Functions
  - ❖ Authentication Using Symmetric Encryption
  - ❖ Message Authentication without Message Encryption
  - ❖ Secure Hash Functions
  - ❖ Other Applications of Hash Functions
- ❑ Digital Signatures and Key Management
  - ❖ Digital Signature
  - ❖ Public-Key Certificates
  - ❖ Symmetric Key Exchange Using Public-Key Encryption
  - ❖ Digital Envelopes

SH. A

Cryptographic Tools

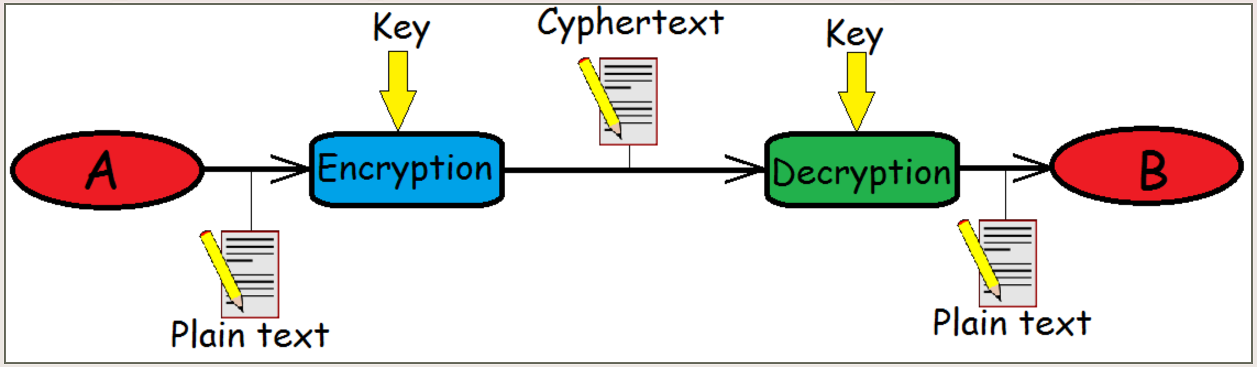
❑ Random and Pseudorandom Numbers

- ❖ The Use of Random Numbers
- ❖ Random versus Pseudorandom

❑ Information Hiding Alternatives

- ❖ Concealment (or Null) Cipher
- ❖ Hiding messages in Media (Steganography)

Definitions



Encryption: is the process of turning a clear-text message (Plaintext) into a meaningless and random sequence of bits (ciphertext). Alternate name (ciphering)

Decryption: is the process of turning ciphertext back into plaintext. Alternate names (decipher – decoding)

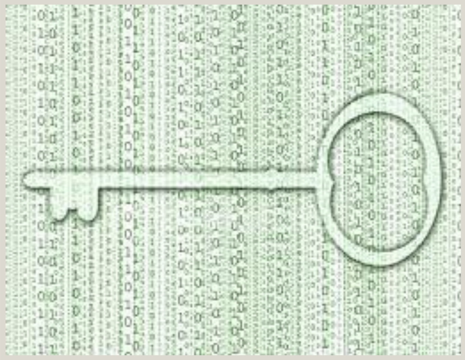
Cryptographic algorithm: is a mathematical function which uses plaintext as the input and produces ciphertext as the output and vice versa. (instructions for how to do the encryption/decryption)

Definitions

Key: is a piece of information that determines the functional output of a cryptographic algorithm or cipher.

Notes:

- Without a key, the algorithm would produce no useful result.
- The security of an encryption system in most cases relies on some key being kept secret.



- A key is often easier to protect (it's typically a small piece of information) than an encryption algorithm, and easier to change if compromised.

Definitions

Cryptography:

- ❑ Is the science of using mathematics to encrypt and decrypt data.
- ❑ Cryptography enables you to store sensitive information or transmit it across insecure networks so that it cannot be read by anyone except the intended recipient
- ❑ Cryptography can be strong or weak. Cryptographic strength is measured in the time and resources it would require to recover the plaintext.
- ❑ The result of strong cryptography is ciphertext that is very difficult to decipher without possession of the appropriate decoding tool

Definitions

Cryptography properties:

Confusion: means that the process drastically changes data from the input to the output.

Diffusion: means that if we change a single character of the input will change many characters of the output, and similarly, if we change single character of the ciphertext, then approximately one half of the plaintext characters should change.

Definitions

Cryptoanalysis: refers to the study of ciphers, ciphertext, or cryptosystems with a view to finding weaknesses in them that will permit retrieval of the plaintext from the ciphertext, without necessarily knowing the key or the algorithm. This is known as breaking the cipher, ciphertext, or cryptosystem.

Cryptanalyst: refers to a person, persons or government organization who is an expert in analyzing and breaking ciphers and codes.

Cryptology: is a branch of mathematics which deals with both cryptography and cryptanalysis.

Cryptography Goals

Cryptography is used to achieve the following security goals:

- ❑ Data confidentiality

  Unauthorized parties cannot access information
- ❑ Data integrity

  Validating the source of the message to ensure the sender is properly identified
- ❑ Authentication

  Assurance that the message was not modified during transmission, accidentally or intentionally
- ❑ non-repudiation.

  A sender cannot deny sending the message at a later date

Classification of Cryptography

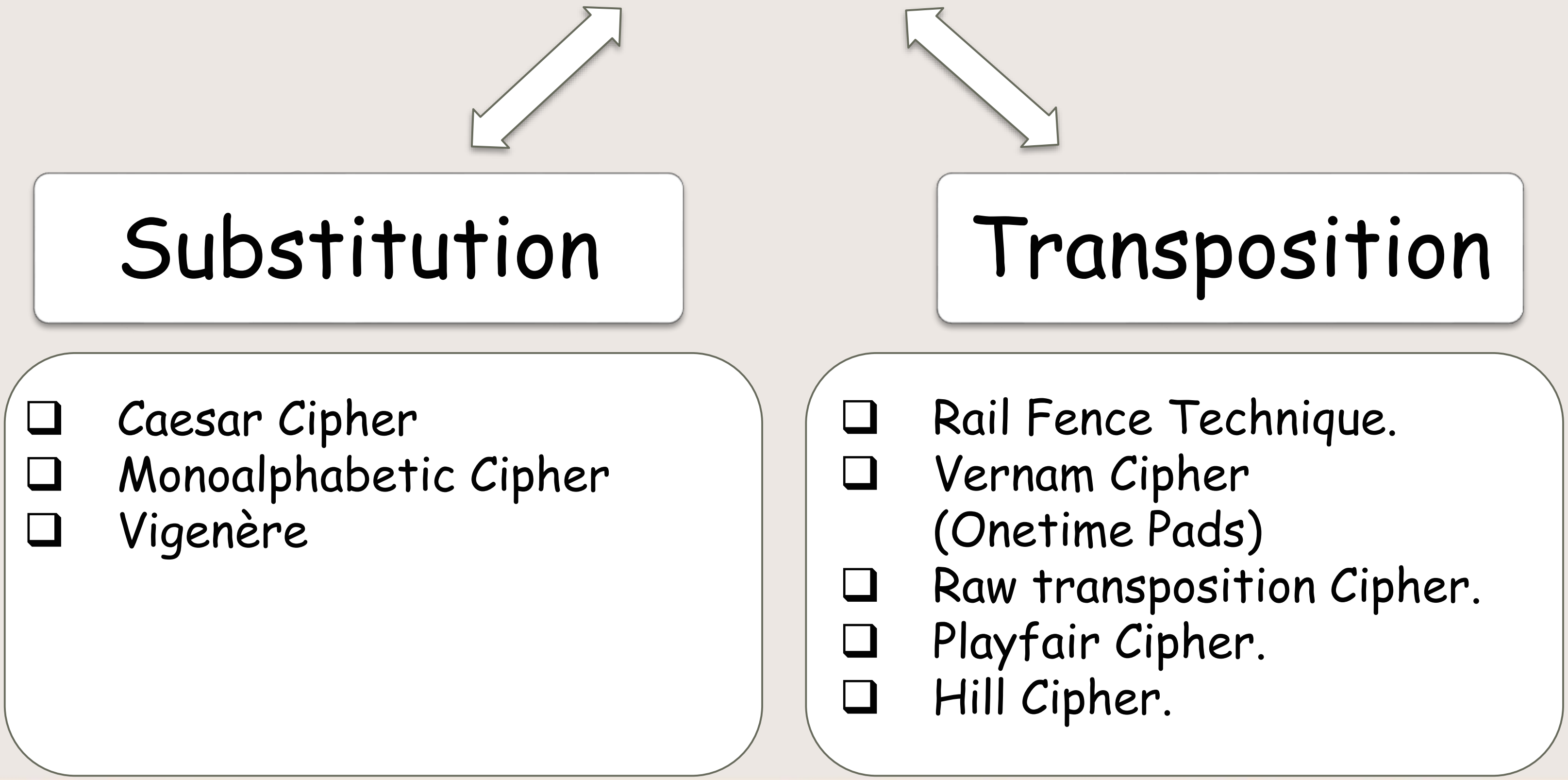
Classification according to transformation operation :

- ❑ Substitution: in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another one. Confusion is achieved by a Substitution.

- ❑ Transposition (Permutation): in which elements re-arranged under the conditions that no information is lost and all operations are reversible. Diffusion is achieved by a Permutation.

Classification of Cryptography

Cryptography according to (transformation operation)



Substitution

- □ Caesar Cipher
- □ Monoalphabetic Cipher
- □ Vigenère

Transposition

- □ Rail Fence Technique.
- □ Vernam Cipher (Onetime Pads)
- □ Raw transposition Cipher.
- □ Playfair Cipher.
- □ Hill Cipher.

Classification of Cryptography

Cryptosystems can be classified into:

- ❑ Symmetric systems: Alternative names:
  - ❖ One (Single) key systems.
  - ❖ Conventional systems
  - ❖ Private key
  - ❖ Secret-key
- ❑ Asymmetric systems: Alternative names:
  - ❖ Public key systems
  - ❖ Two-key systems

Classification of Cryptography

Classification according to transformation way of processing :

- ❑ **Block cipher:** in which the plain text is processed one block of elements at a time and producing an output one block

- ❑ **Stream cipher:** in which the plaintext is processed bit by bit or byte by byte.

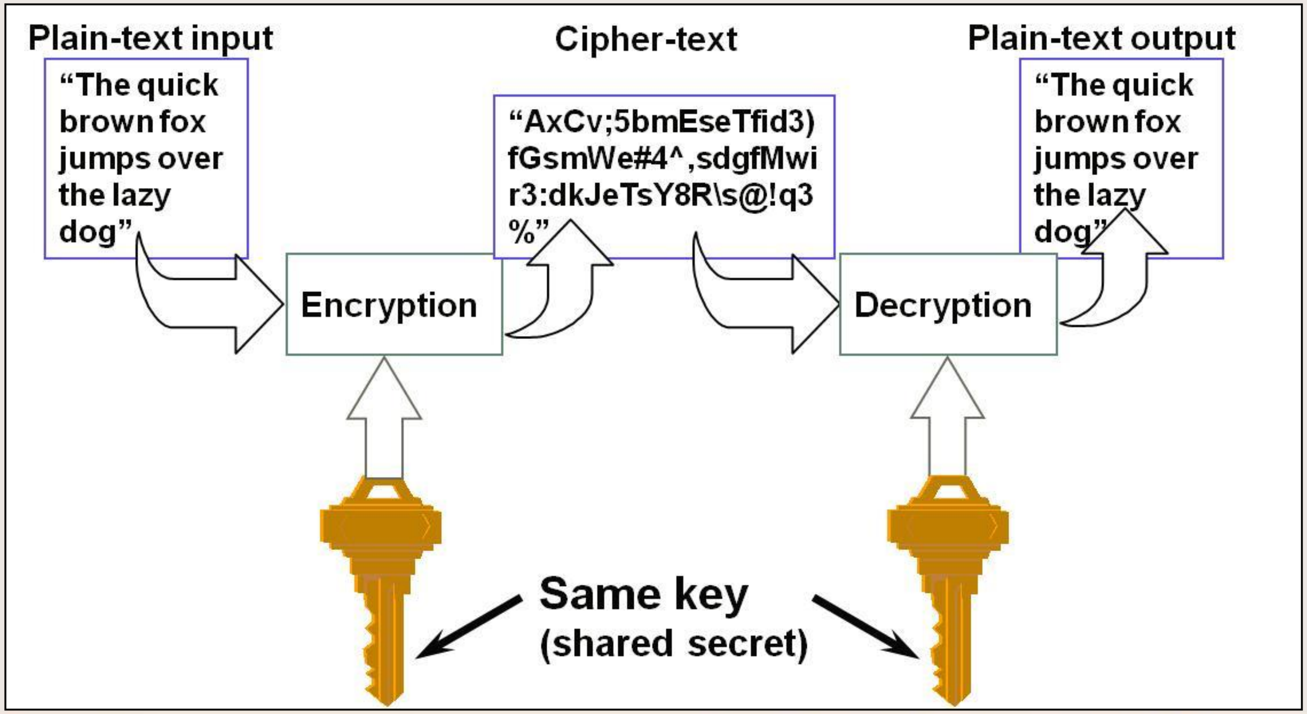
Classification of Cryptography

Classification according to timeline :

- Classic cipher: systems used before computer invention
- Modern cipher: systems used after computer invention

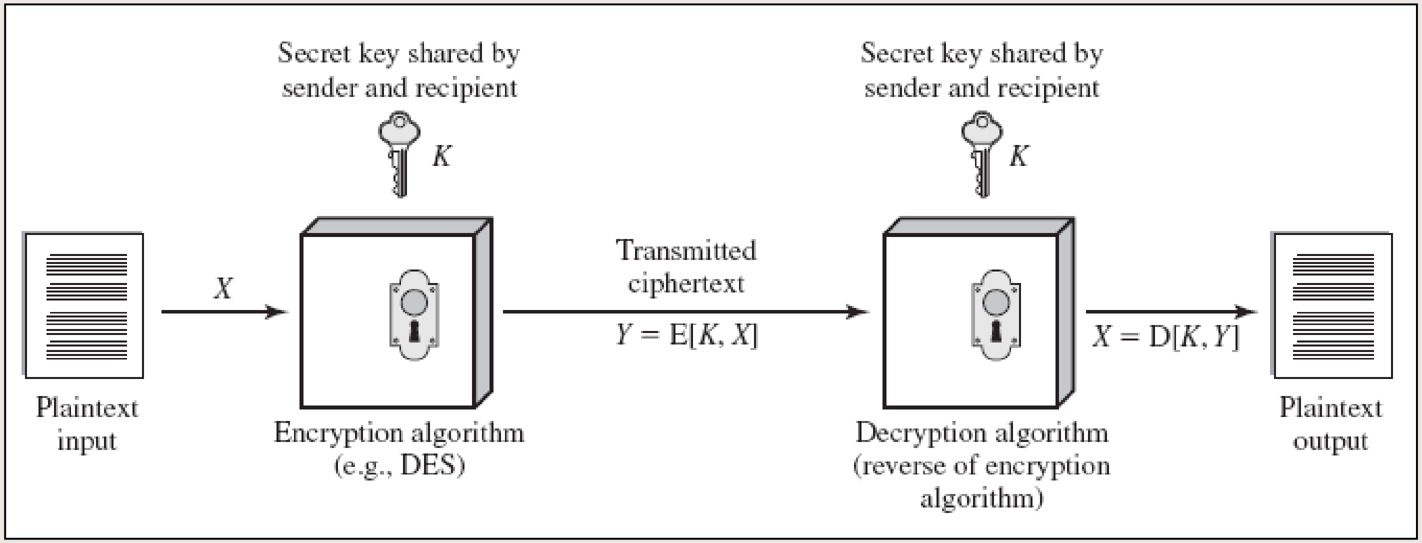
Symmetric Encryption

- ❑ The universal technique for providing confidentiality for transmitted or stored data
- ❑ Referred to as (conventional or single key or private key or secret-key) encryption.


Plain-text input
"The quick brown fox jumps over the lazy dog"
Cipher-text
"AxCv;5bmEseTfid3)fGsmWe#4^,sdgfMwir3:dkJeTsY8R\s@!q3%"
Plain-text output
"The quick brown fox jumps over the lazy dog"
Encryption
Decryption
Same key
(shared secret)

Symmetric Encryption

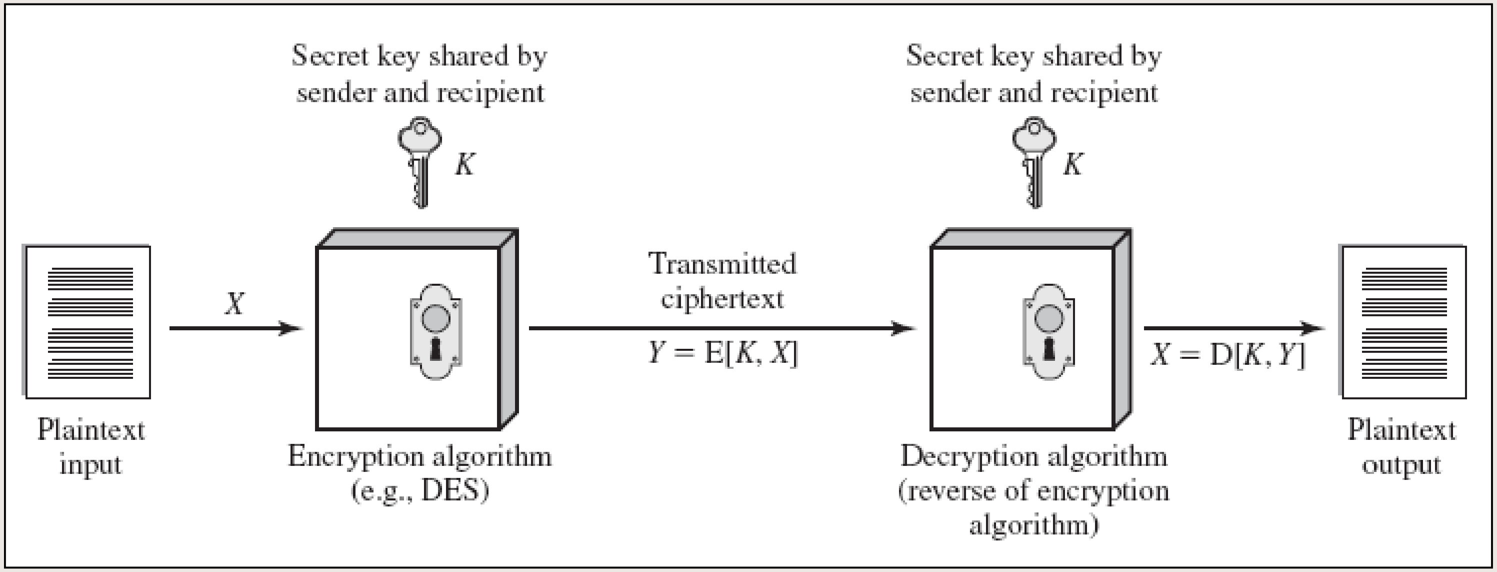
A symmetric encryption has five ingredients:


Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
Y = E[K, X]
X = D[K, Y]
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

1. **Plaintext**: This is the original message or data that is fed into the algorithm as input.
2. **Encryption algorithm**: The encryption algorithm performs various substitutions and transformations on the plaintext. (instructions for how to do the encryption)
3. **Secret key**: The secret key is also input to the encryption algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.

Symmetric Encryption

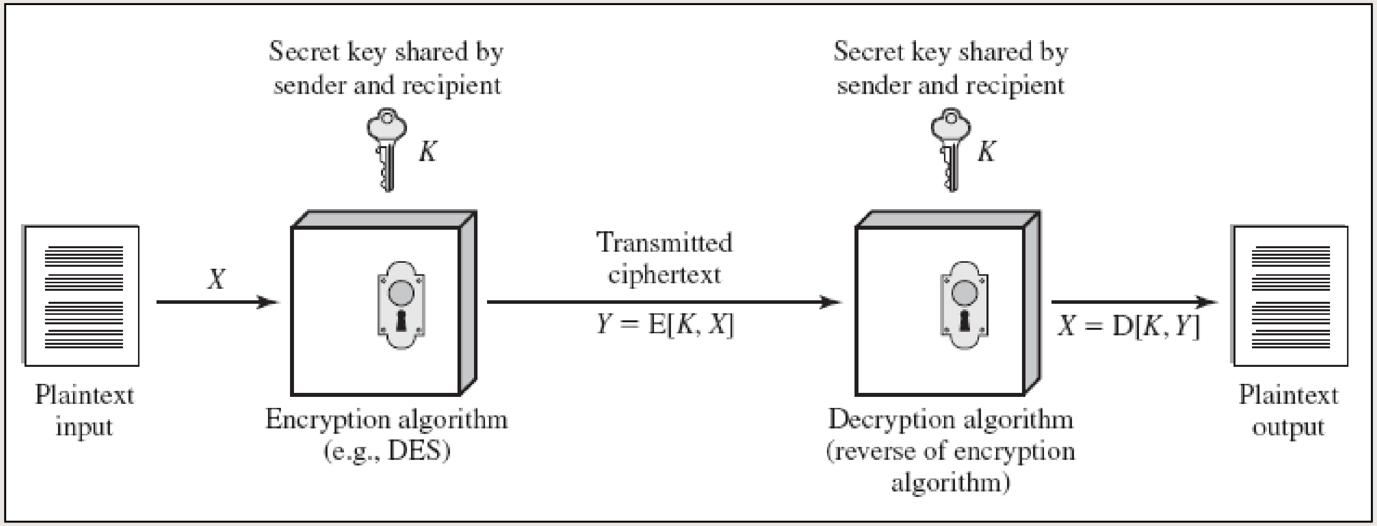
A symmetric encryption has five ingredients:


Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
Y = E[K, X]
X = D[K, Y]
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

4. 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.
5. Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Symmetric Encryption

Two requirements for secure use of symmetric encryption:


Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
Y = E[K, X]
X = D[K, Y]
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

1. We need a strong encryption algorithm
2. Sender and receiver must have obtained copies of the secret key. The key must kept secure.

If someone can discover the key and knows the algorithm, all communication using this key is readable.

Attacking Symmetric Encryption

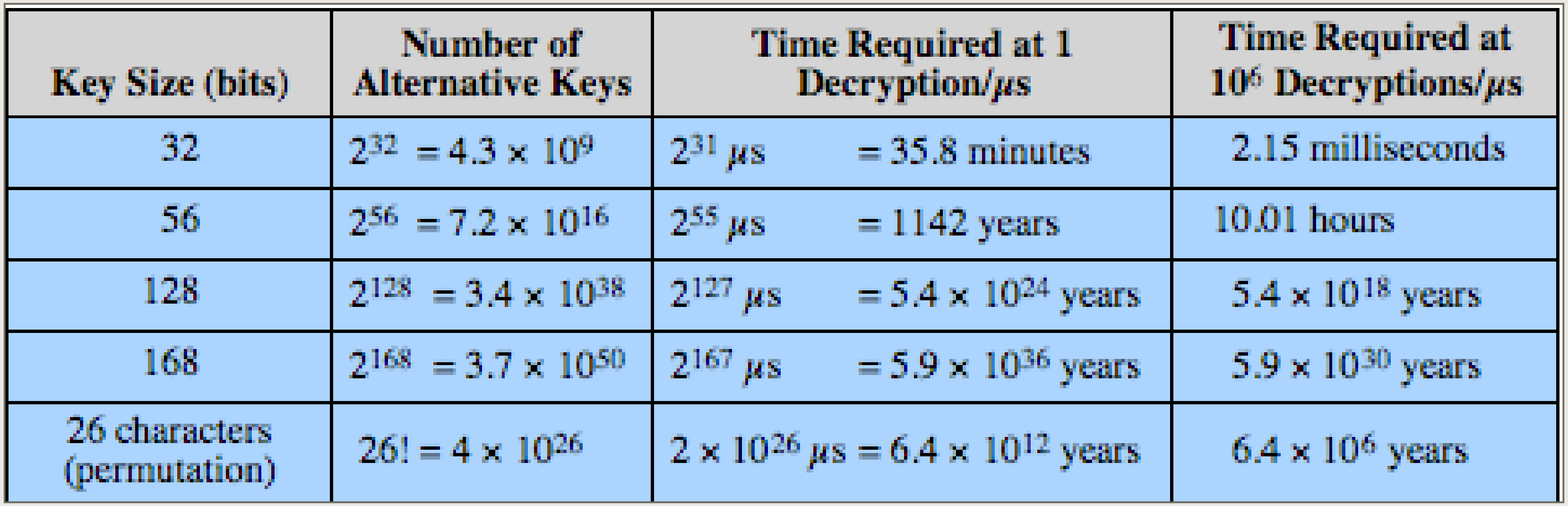
- Cryptanalysis:

Rely on nature of the algorithm plus some knowledge of plaintext characteristics even some sample plaintext - ciphertext pairs exploits characteristics of algorithm to deduce specific plaintext or key.

Attacking Symmetric Encryption

Brute-force attack:

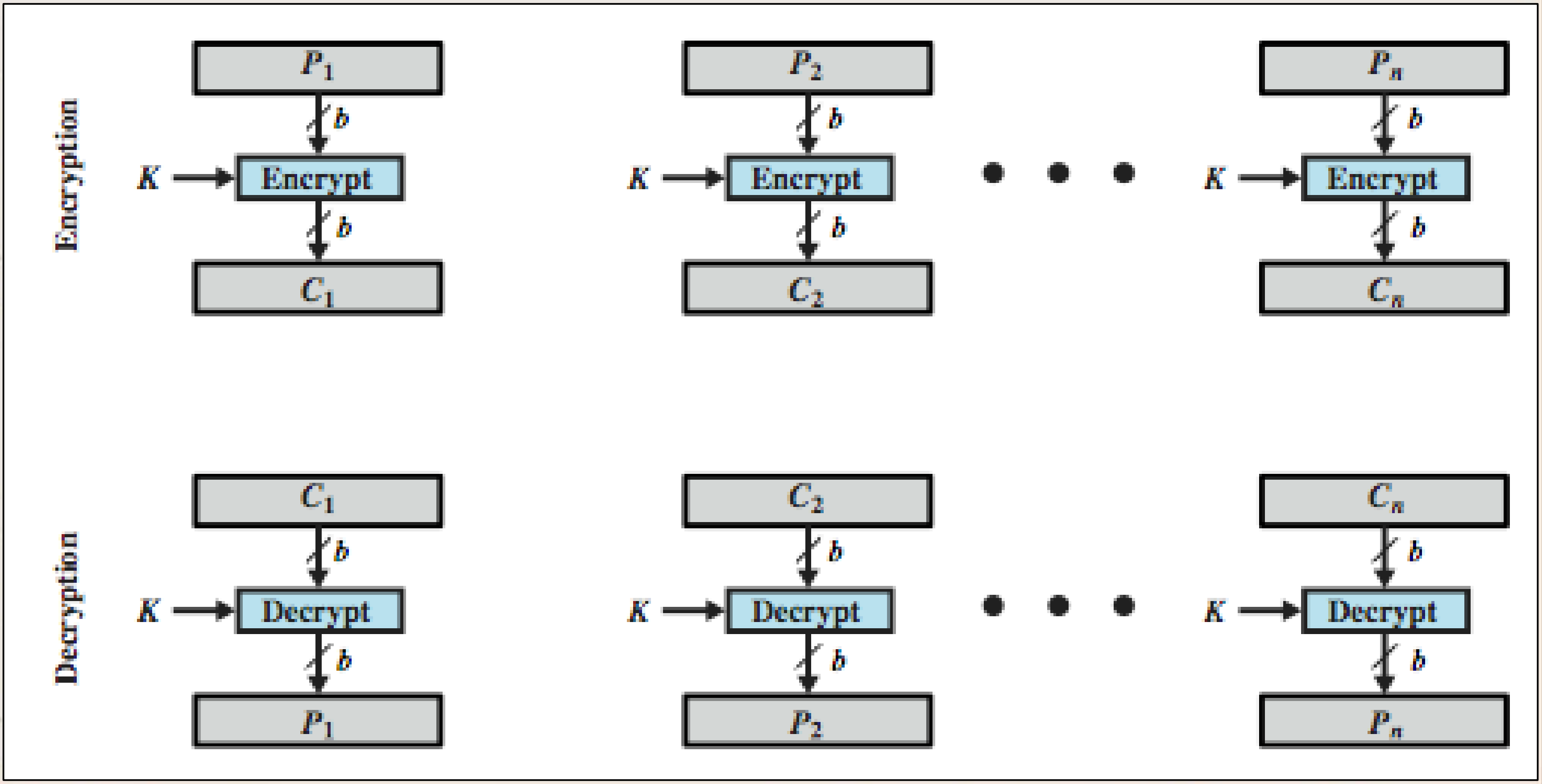
- Try all possible keys on some ciphertext until get an intelligible translation into plaintext
- Brute-force attack (Exhaustive Key Search)



Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/µs	Time Required at 10^6 Decryptions/µs
32	$2^{32} = 4.3 \times 10^9$	2^{31} µs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} µs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} µs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} µs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} µs = 6.4×10^{12} years	6.4×10^6 years

Types of Symmetric Encryption

❑ Block Ciphers



Types of Symmetric Encryption

❑ Block Ciphers

- ❖ The most commonly used symmetric encryption algorithms are block ciphers
- ❖ A block cipher processes the plaintext input in fixed size blocks and produces a block of ciphertext of equal size for each plaintext block
- ❖ The most important symmetric algorithms, all of which are block ciphers, are :
  - ➢ Data Encryption Standard (DES).
  - ➢ Triple DES (TDES - 3DES)
  - ➢ Advanced Encryption Standard (AES).

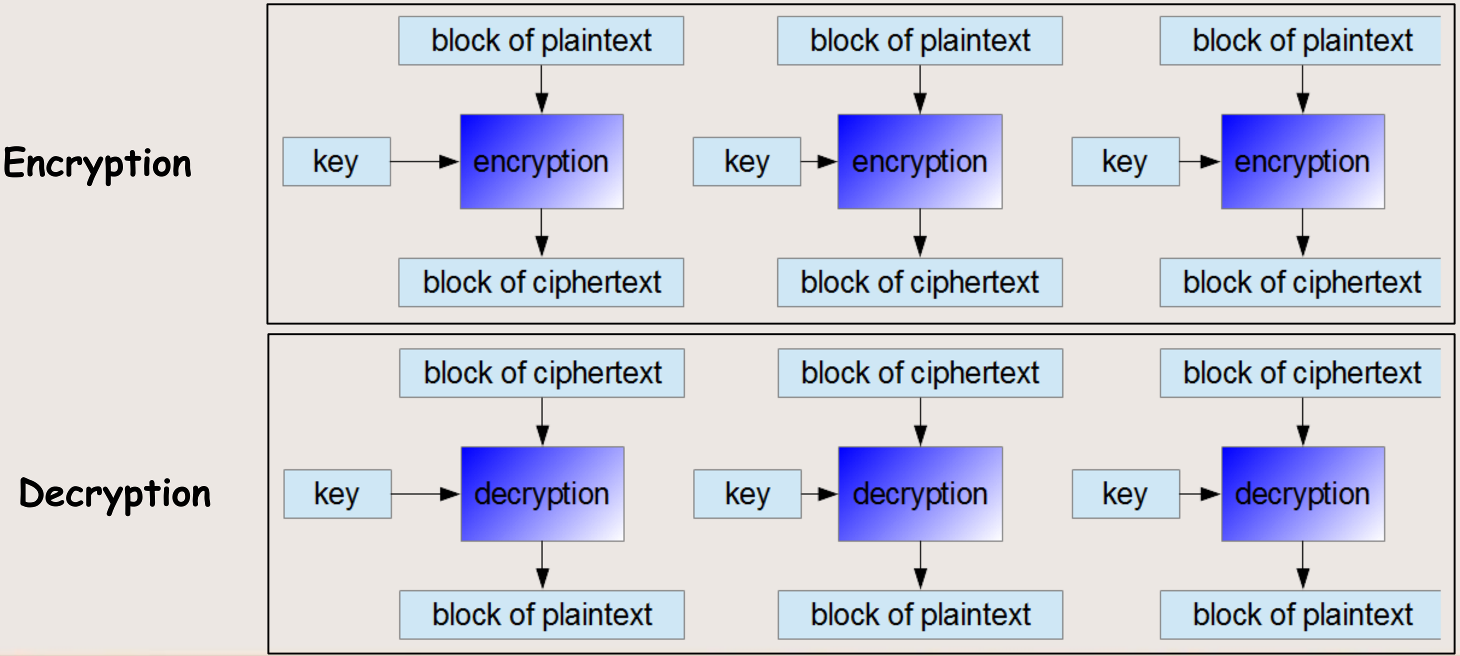
Block Ciphers modes of operation

- ❑ ECB (electronic codebook)
- ❑ CBC (cipher-block chaining) Mode
- ❑ PCBC (propagating or plaintext cipher-block chaining)
- ❑ CFB (cipher feedback)
- ❑ OFB (output feedback)

Block Ciphers modes of operation

❑ ECB (electronic codebook) Mode

- ❖ It is the simplest mode of encryption.
- ❖ Each plaintext block is encrypted separately.
- ❖ Similarly, each ciphertext block is decrypted separately.
- ❖ DES use ECB mode for encryption



Block Ciphers modes of operation

❑ ECB (electronic codebook) Mode

Advantages:

- ❖ Any part of encrypted message could be easily decrypted (or re-encrypted after modification)
- ❖ Error multiplication properties:

If ciphertext is modified by attacker, modifications in plaintext would be random, unpredictable and inside one block only

Block Ciphers modes of operation

❑ ECB (electronic codebook) Mode

Disadvantages: attacker can detect:

❖ Whether two ECB-encrypted messages are identical;

❖ Whether two ECB-encrypted messages share a common prefix;

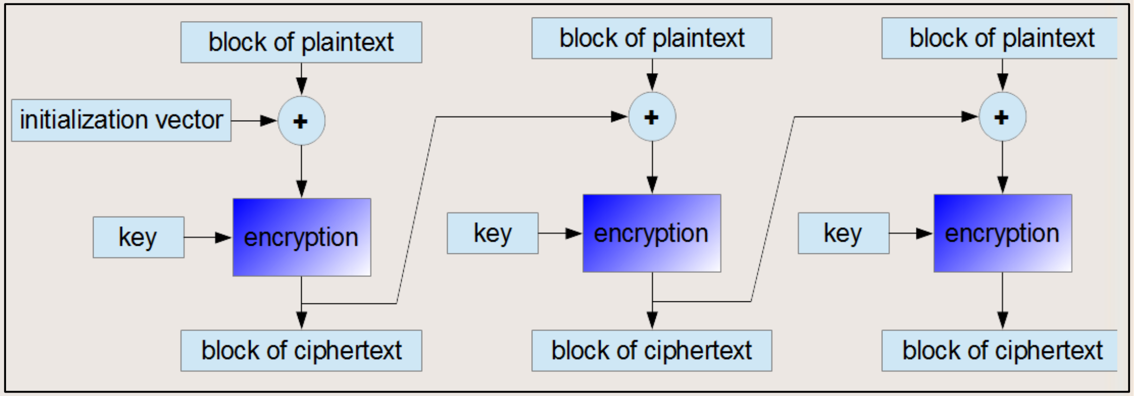
❖ Whether two ECB-encrypted messages share other common substrings, as long as those substrings are aligned at block boundaries; or

❖ Whether (and where) a single ECB-encrypted message contains repetitive data (such as long runs of spaces or null bytes, repeated header fields or coincidentally repeated phrases in text).

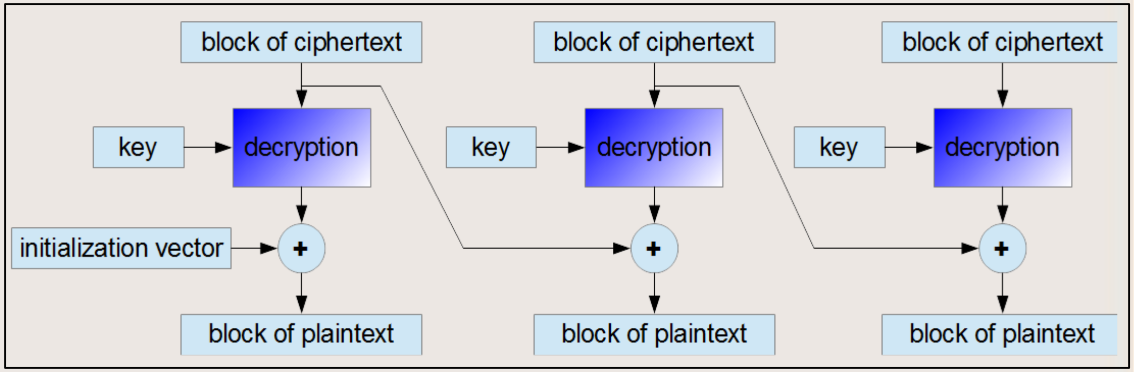
Block Ciphers modes of operation

❑ CBC (cipher-block chaining) Mode

- ❖ Adding XOR each plaintext block to the ciphertext block that was previously produced.
- ❖ The result is then encrypted using the cipher algorithm in the usual way.
- ❖ Each subsequent ciphertext block depends on the previous one. The first plaintext block is added XOR to a random initialization vector . The vector has the same size as a plaintext block.



Encryption



Decryption

Block Ciphers modes of operation

❑ CBC (cipher-block chaining) Mode

Advantages:

❖ Equal messages using the same keys will be encrypted to different ciphertexts (using different Initial Vectors)

❖ Message can be decrypted from any part.

❖ Error multiplication properties (single bit + the next block)

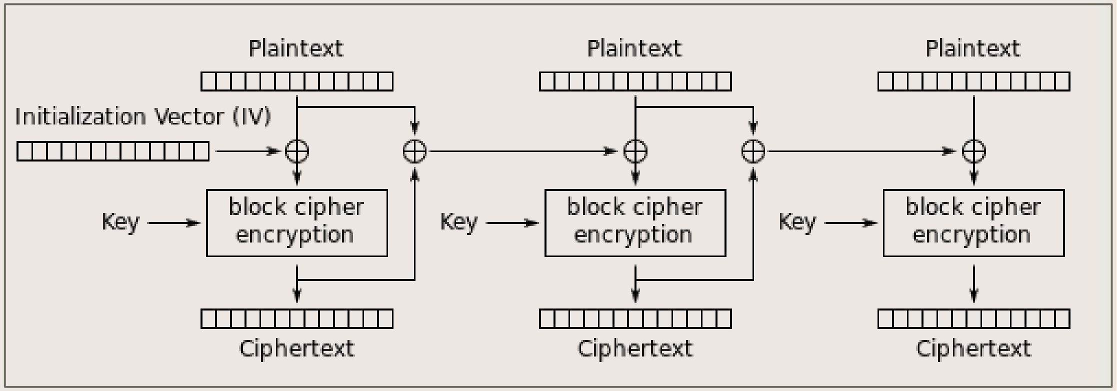
Disadvantages:

❖ A single bit error in a ciphertext block affects the decryption of all subsequent blocks.

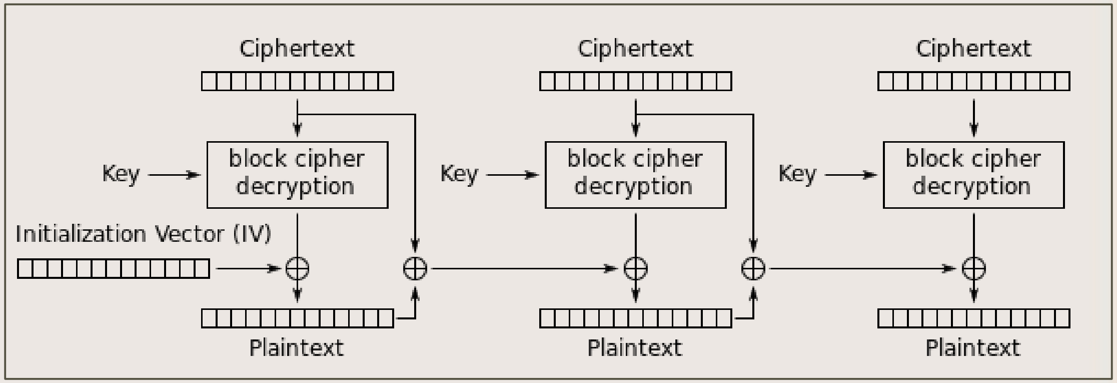
❖ Rearrangement of the order of the ciphertext blocks causes decryption to become corrupted.

Block Ciphers modes of operation

- ❑ PCBC (propagating or plaintext cipher-block chaining) Mode



Encryption



Decryption

- ❖The PCBC mode is similar to the CBC mode
- ❖It also mixes bits from the previous and current plaintext blocks, before encrypting them

Block Ciphers modes of operation

❑ PCBC (propagating or plaintext cipher-block chaining) Mode

Advantages:

❖ if two adjacent ciphertext blocks are exchanged, this does not affect the decryption of subsequent blocks.

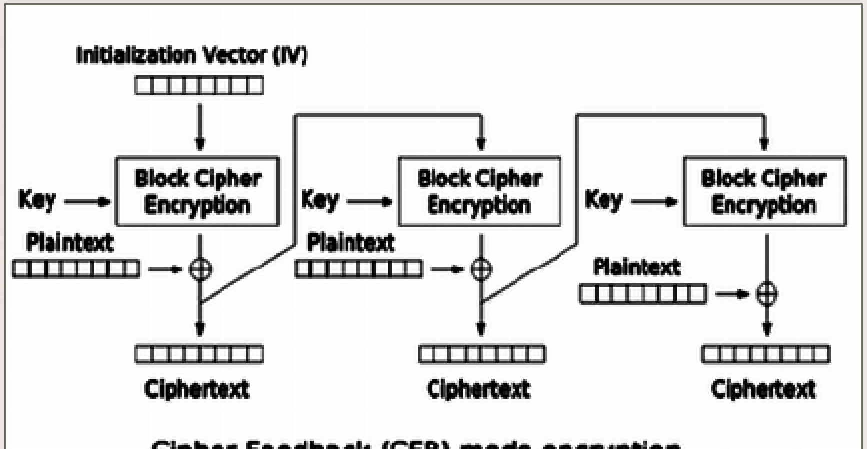
Disadvantages:

❖ Both encryption and decryption can be performed using only one block at a time.

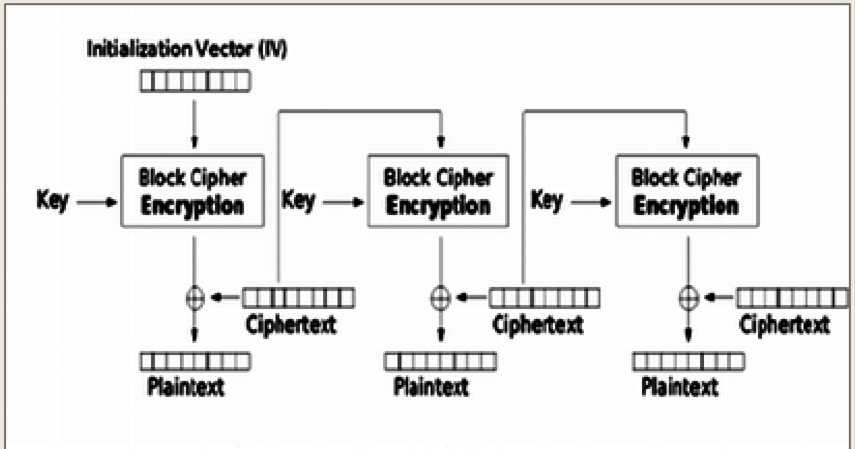
❖ IF Two different IV's are used for encryption and decryption, only the first block will be affected

Block Ciphers modes of operation

❑ CFB (cipher feedback) Mode



Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption

❖ Encrypt ciphertext data from the previous round and then add the output to the plaintext bits

Block Ciphers modes of operation

❑ CFB (cipher feedback) Mode

Advantages:

- ❖ Equal messages using the same keys will be encrypted to different ciphertexts
- ❖ Message length can be arbitrary
- ❖ Randomness of IV is not needed
- ❖ Decryption implementation is not needed

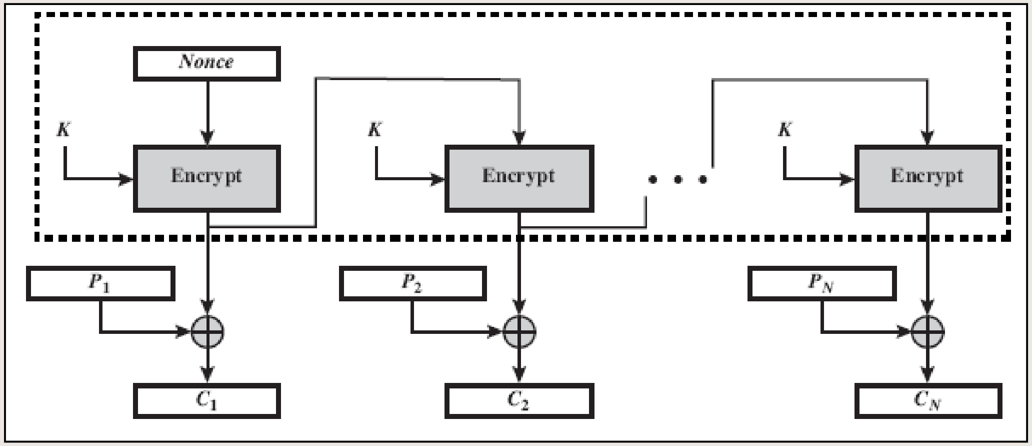
Disadvantages:

- ❖ Message blocks cannot be decrypted from any part or re-encrypted after modification
- ❖ Encryption speed is significantly slower

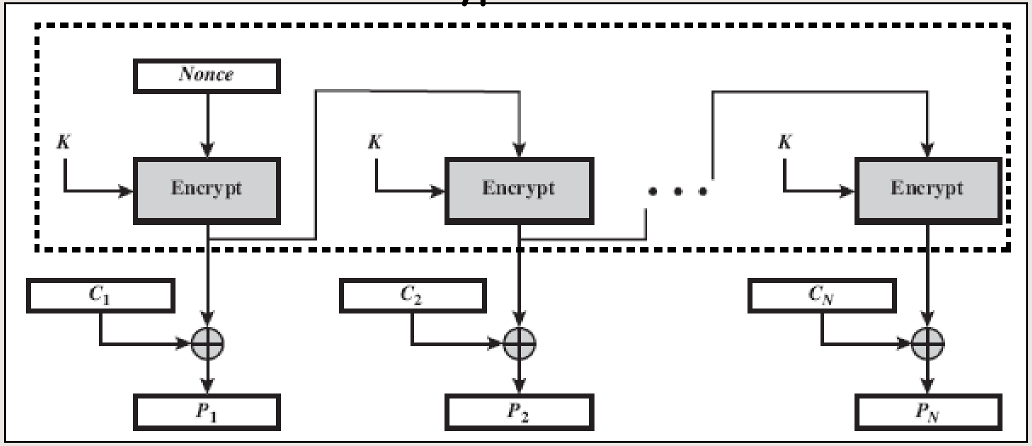
Block Ciphers modes of operation

❑ OFB (output feedback) Mode

- ❖ The algorithm create key stream bits that are used for encryption subsequent data blocks.
- ❖ The way of working of the block cipher becomes similar to the way of working of a typical stream cipher.



Encryption



Decryption

Block Ciphers modes of operation

❑ OFB (output feedback) Mode

Advantages:

- ❖ Equal messages using the same keys will be encrypted to different cryptograms (ciphertexts)
- ❖ Message length can be arbitrary
- ❖ Randomness of IV is not needed
- ❖ Decryption implementation is not needed

Disadvantages:

- ❖ No error multiplication properties
- ❖ Message blocks cannot be decrypted from any part or re-encrypted after modification

Block Ciphers modes of operation

❑ Quiz (final exam 2016/2017)

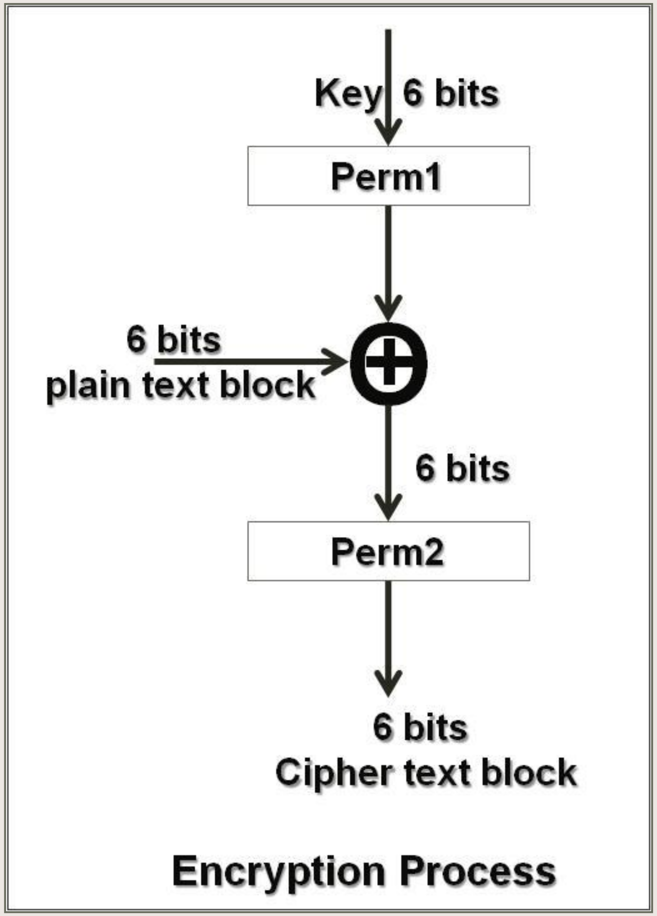
A simple cipher system uses a 6 bit plain text and 6 bit key work as a follow:

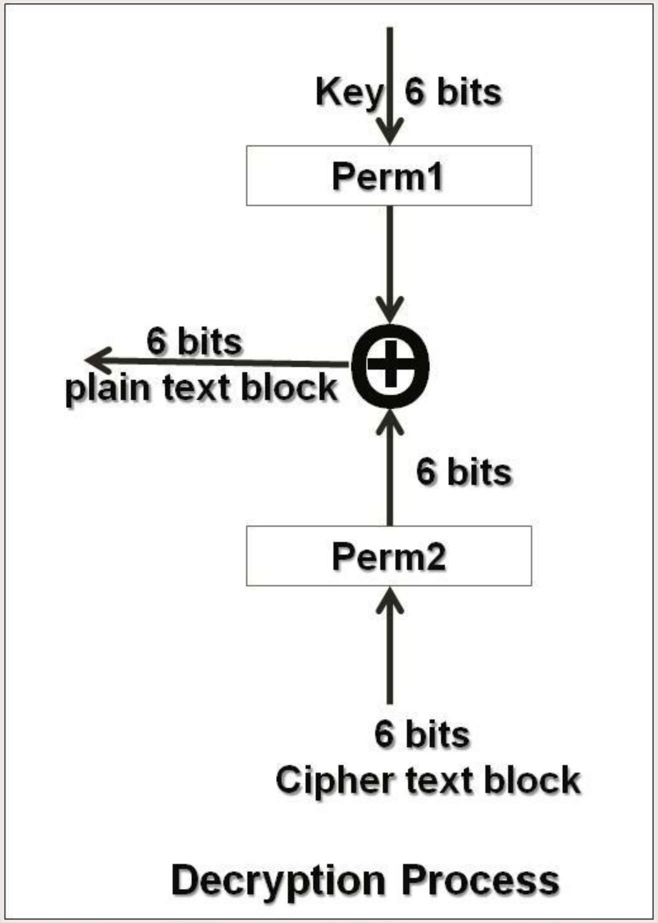
- A 6 bit key is permuted according to (3,2,4,1,2,5) to produce different key in each round.
- XOR result key with the plain text.
-Permuting the resulting bits according to (3,2,4,6,5,1).

Draw a block diagram for encryption and decryption process.

Block Ciphers modes of operation

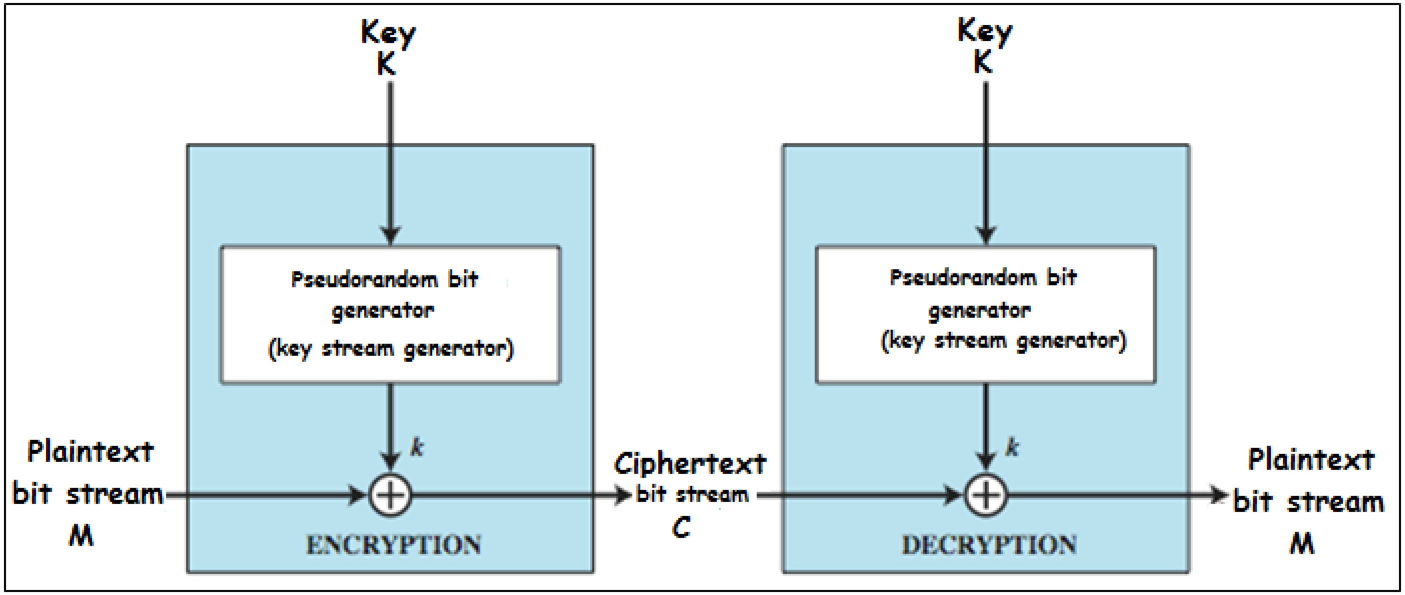
- ❑ Answer:





Types of Symmetric Encryption

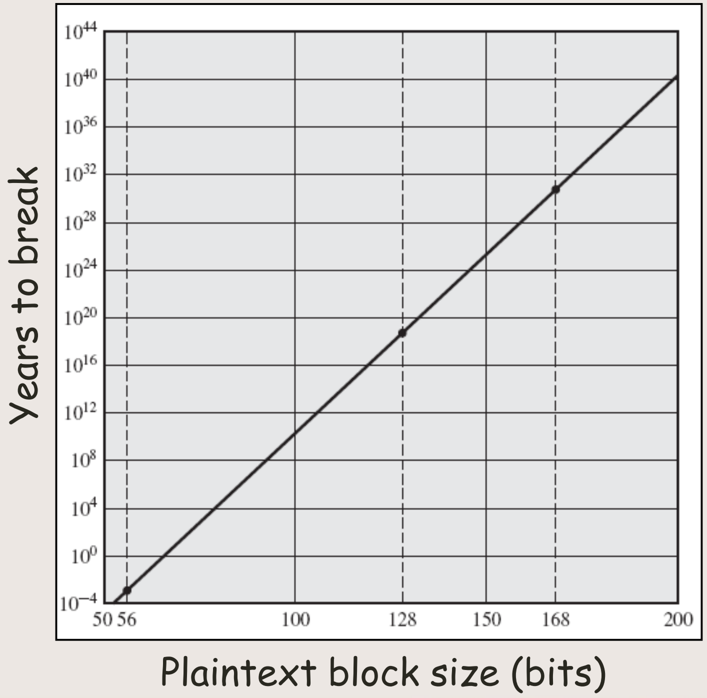
❑ Stream Ciphers


Key
K
Pseudorandom bit generator (key stream generator)
k
Plaintext bit stream M
ENCRYPTION
Ciphertext bit stream C
Key
K
Pseudorandom bit generator (key stream generator)
k
DECRYPTION
Plaintext bit stream M

- ❑ A method of encrypting in which a cryptographic key and algorithm are applied to each binary digit in a data stream, one bit at a time.
- ❑ This method is not much used in modern cryptography.

Symmetric Algorithms Key Length & Block Size

	DES	TDES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112/168	128/192/256



Block Vs stream cipher

Stream Cipher	Block Cipher
Operates on smaller units of plaintext	Operates on larger block of data
Faster than block cipher	Slower than Stream Cipher
Processes the input element continuously producing output one element at a time	Processes the input one block of element at a time, producing an output block for each input block
Require less code	Requires more code
Only one time of key used.	Reuse of key is possible
Ex: One time pad	Ex: DES (Data Encryption Standard)
Application: SSL (secure connection on the web)	Application: Database, file encryption.
More suitable for hardware implementation	Easier to implement in software.

DES (Data Encryption Standard)

❑ Data Encryption Standard (DES):

- ❖ NIST issued for DES contest proposal in August 1974
- ❖ An implementation of a Feistel Cipher designed by IBM company won the contest.
- ❖ In 1977 by the NIST issued Feistel as DES winner
- ❖ The most widely used encryption scheme
- ❖ Uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block

DES (Data Encryption Standard)

How does DES Encryption Algorithm work:

- ❑ DES works by encrypting blocks of 64 message bits, which is equivalent to 16 hexadecimal numbers.
- ❑ DES uses "keys" 16 hexadecimal numbers long (64 bits). every 8th key bit is ignored so that the effective key size is 56 bits.
- ❑ Message must be padded with some extra bytes, so the message length should be a multiple of block size. For example:

Message (M) = "Your lips are smoother than vaseline"

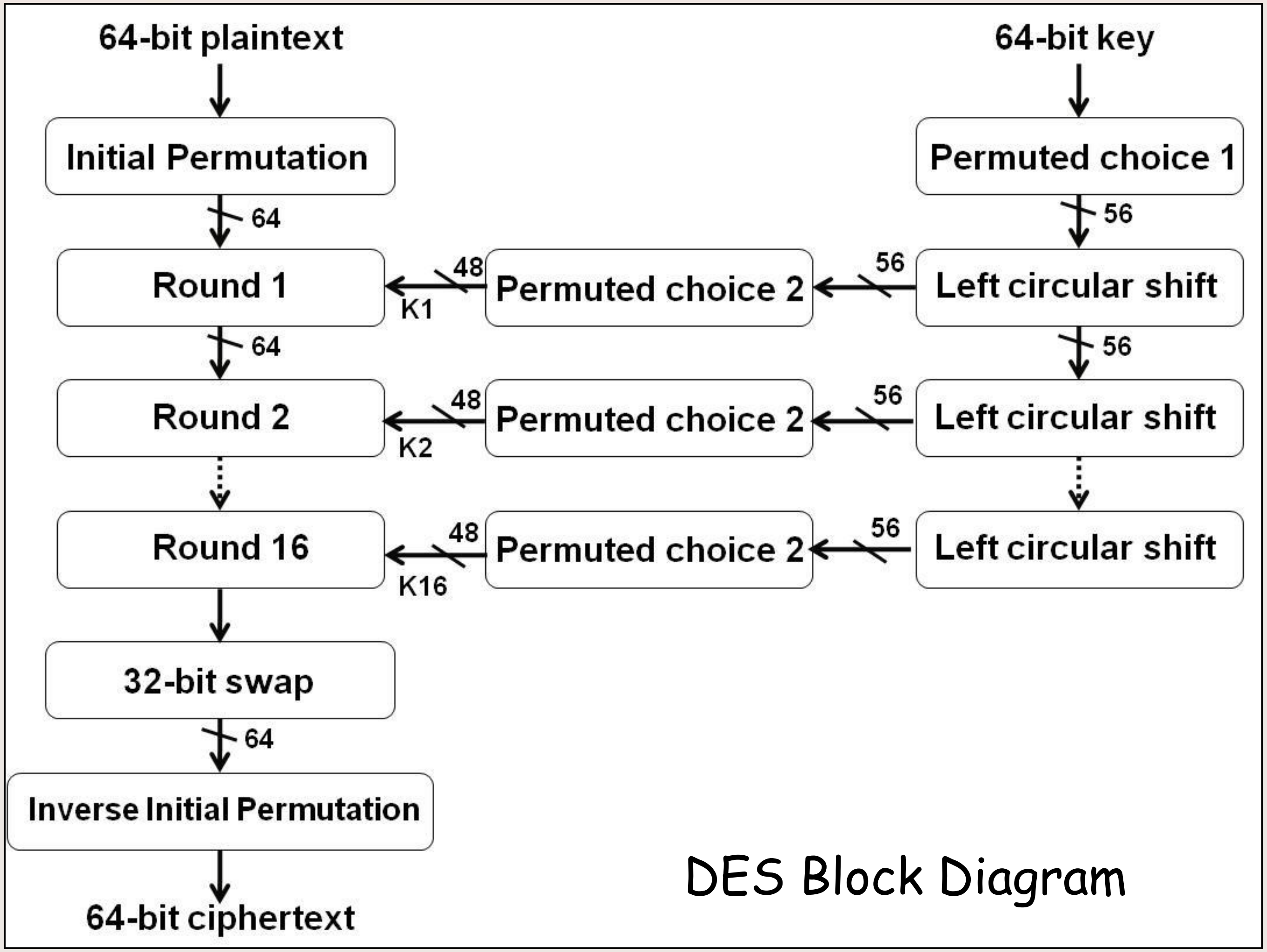
(M) In Hexa ="596F7572206C6970 732061726520736D 6F6F746865722074 68616E2076617365 6C696E650D0A"

(M) Padded = "596F7572206C6970 732061726520736D 6F6F746865722074 68616E2076617365 6C696E650D0A*0000*".

DES key = "133457799BBCDFF1"

K = 0001001~~1~~0011010~~0~~0101011~~1~~0111100~~1~~1001101~~1~~1011110~~0~~1101111~~1~~1111000~~1~~

DES (Data Encryption Standard)



DES Block Diagram

DES (Data Encryption Standard)

❑Initial Permutation

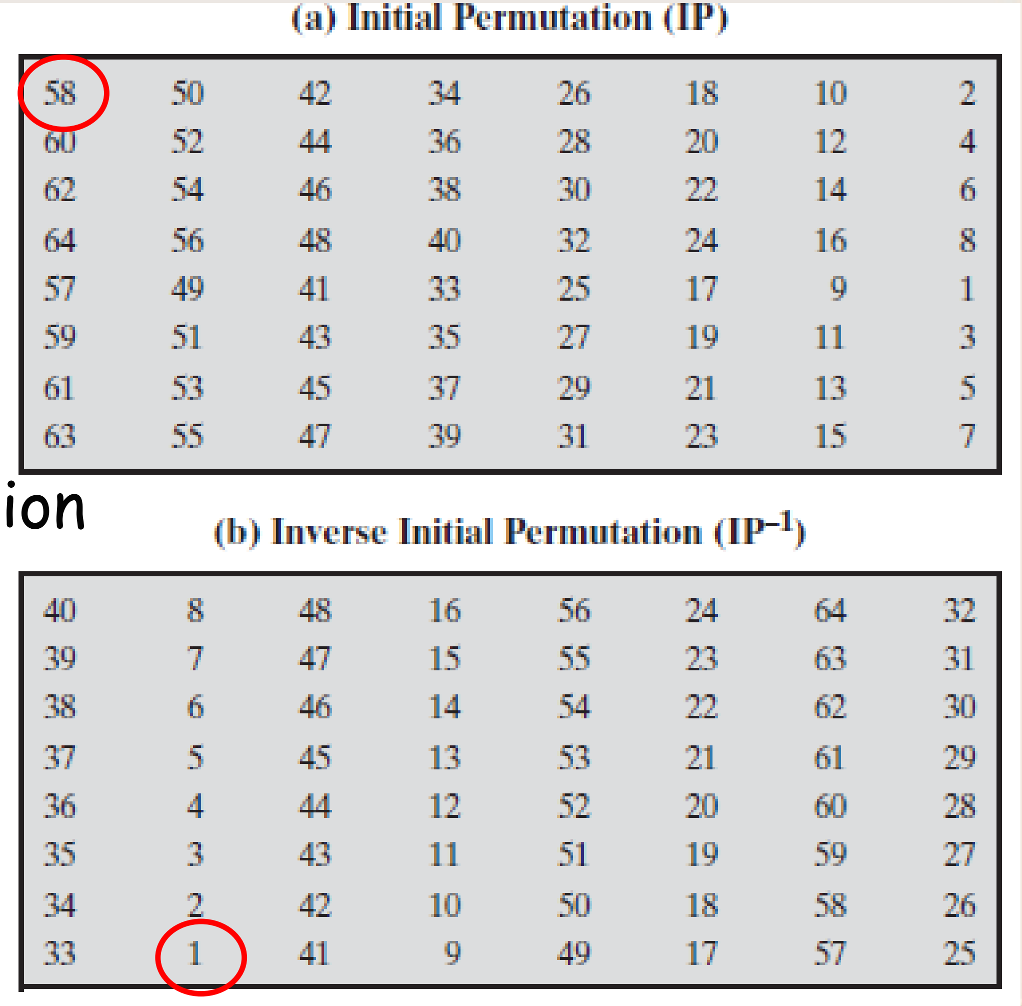
The permutation
X = IP(*M*)

Input 64 bits
Output 64 bits

❑The inverse permutation

Y = $IP^{-1}(X) = IP^{-1}(IP(M))$
The original ordering
is restored

Input 64 bits
Output 64 bits



(a) Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

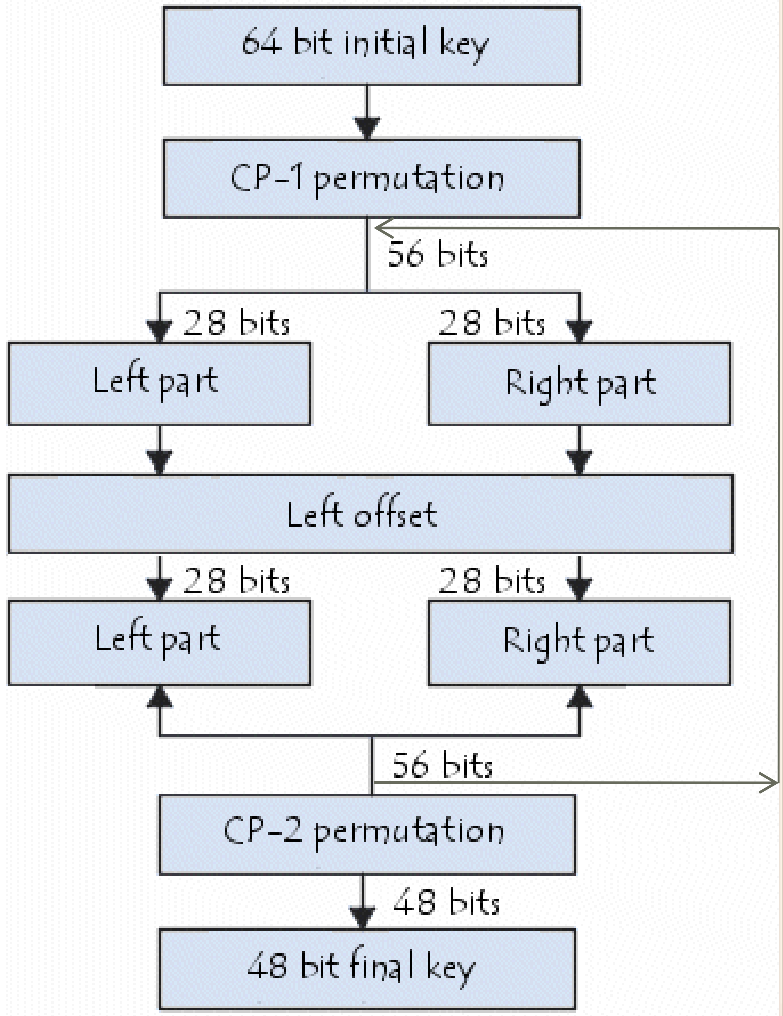
(b) Inverse Initial Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

DES (Data Encryption Standard)

❑ Generation of keys

- ❖ Input key of 64 bit
- ❖ Produce key output of length 48 bit
- ❖ The process is repeated to generate a key of each round (16 rounds)



DES (Data Encryption Standard)

❑ Permutation choice-1 (CP-1)

- ❖ Input key of 64 bit
- ❖ Produce key output of length 56 bit

Permuted Choice-1

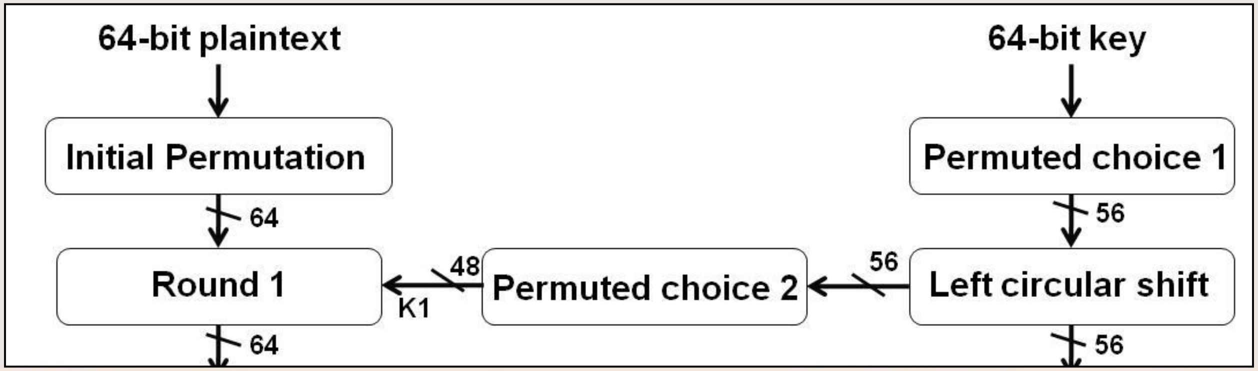
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

DES (Data Encryption Standard)

❑ Left circular shift

- ❖ Split 56 bit this key into left and right halves, where each half has 28 bits.
- ❖ To do a left shift, for each of the two 28 bit block move each bit one place to the left, except for the first bit, which is cycled to the end of the block. Number of shits according to the right table

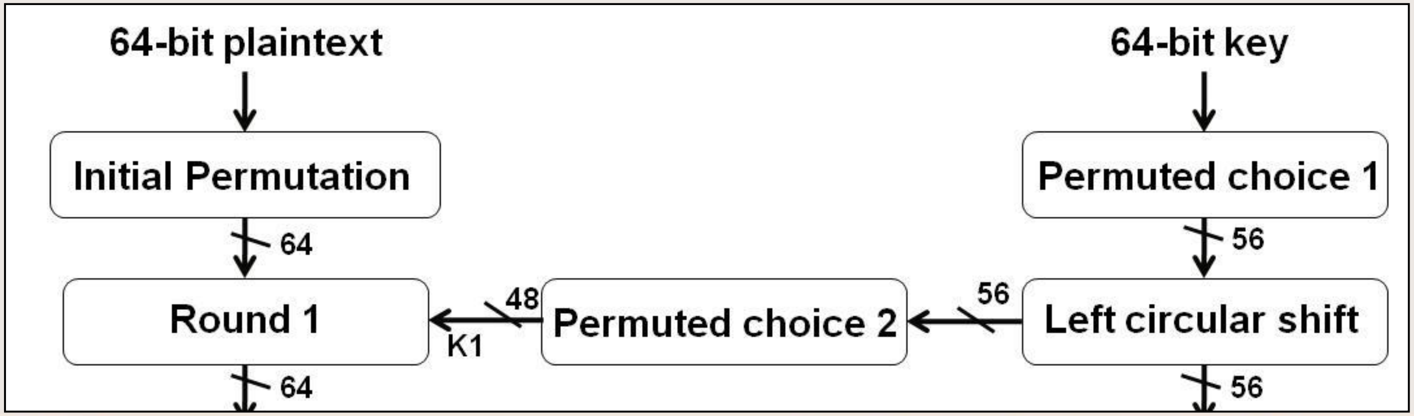
Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1



DES (Data Encryption Standard)

- □ Permutation choice-2 (CP-2) Permuted Choice-2
  - ❖ Input key of 56 bit
  - ❖ Produce key output of length 48 bit

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32



64-bit plaintext

64-bit key

Initial Permutation

Permuted choice 1

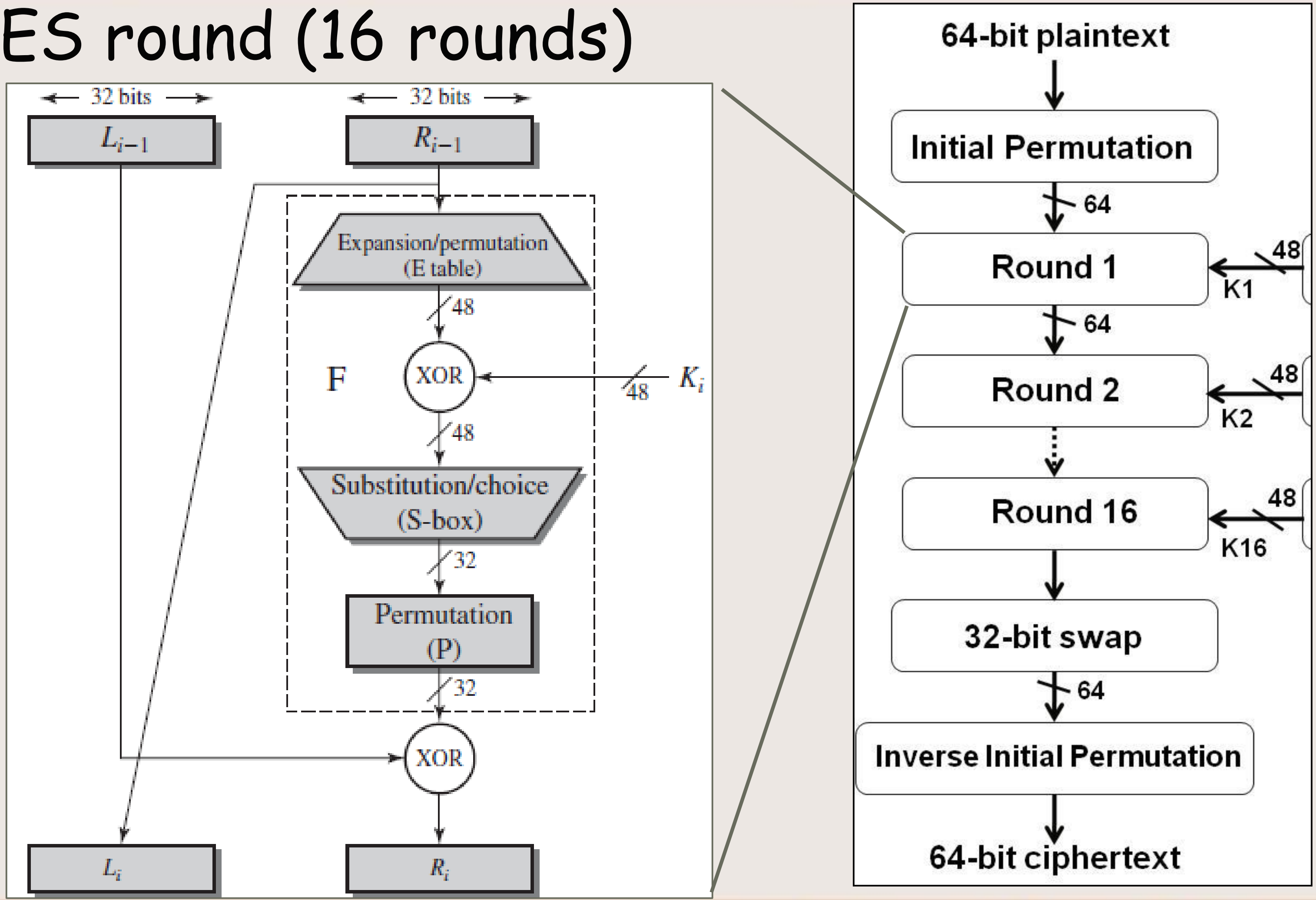
Round 1

Permuted choice 2

Left circular shift

DES (Data Encryption Standard)

- ❑ DES round (16 rounds)



Triple-DES (TDES or 3DES)

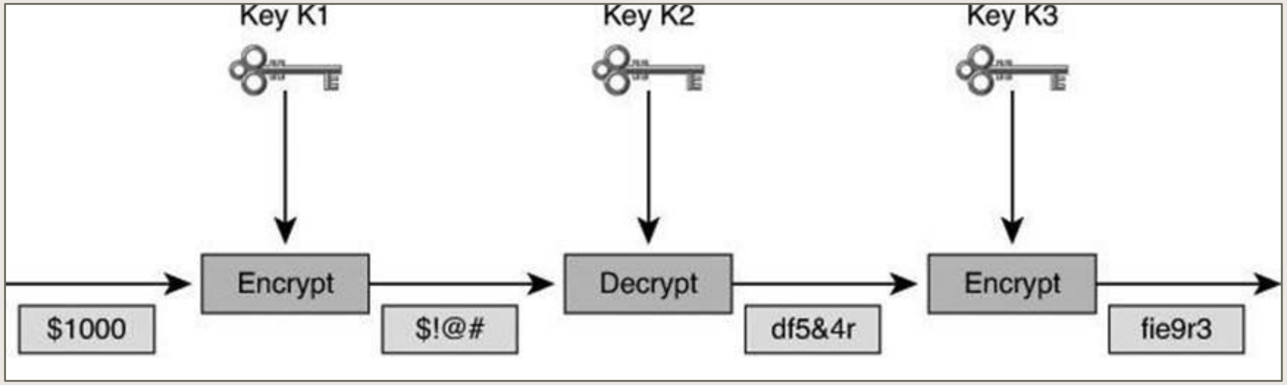
- ❑ Provides a relatively simple method of increasing the key size of DES to protect against attacks, without the need to design a completely new block cipher algorithm.
- ❑ Repeats basic DES algorithm three times
- ❑ Much more secure but also much slower
- ❑ Using either two or three unique keys
- ❑ Key size : 168 (3*56) or 112 (2*56)

Triple-DES (TDES or 3DES)

- ❑ Encrypting the data three times with three different keys does not significantly increase security
- ❑ 3DES keying options:

Option #1: which provides 168-bit strength, uses three independent 56-bit values for K1, K2, and K3.

Option #2: K1 = K3 use the same value and K2 is a different value, so two different 56-bit keys produce a 112-bit strength.



- 3DES-EDE Method: Encrypt using K1 then Decrypt using K2 then Encrypt using K3
- IF K1 = K3, key length=112-bit
- IF K1 ≠ K3, key length=168-bit

Advanced Encryption Standard (AES)

- ❑ Needed a better replacement for 3DES.
- ❑ NIST called for proposals in 1997.
- ❑ Selected Rijndael in Nov 2001.
- ❑ Symmetric block cipher.
- ❑ Uses 128 bit data & 128/192/256 bit keys.
- ❑ Now widely available commercially.
- ❑ Exist Software & Hardware versions.

DES, 3DES, AES Comparison

DES	3DES	AES
56	56, 112,168	128,192,256 bits
Weak	Moderate	Strong
Slow	Slower than DES	Faster than 3DES
Software	Software	Software/hardware

Others Symmetric Algorithms

❑ RC5

- ❖ Block cipher, with variable block, key, and round .
- ❖ Block sizes are 32, 64, and 128
- ❖ Rounds up to 256. Key size up to 2048 bits

❑ Twofish

- ❖ Block cipher, with 128-bit blocks in 16 rounds .
- ❖ Key sizes up to 256 bits.

Symmetric key Encryption Weaknesses

Key distribution & management

- □ It requires a secure mechanism to deliver keys properly.
- □ It requires a good mechanism for key management

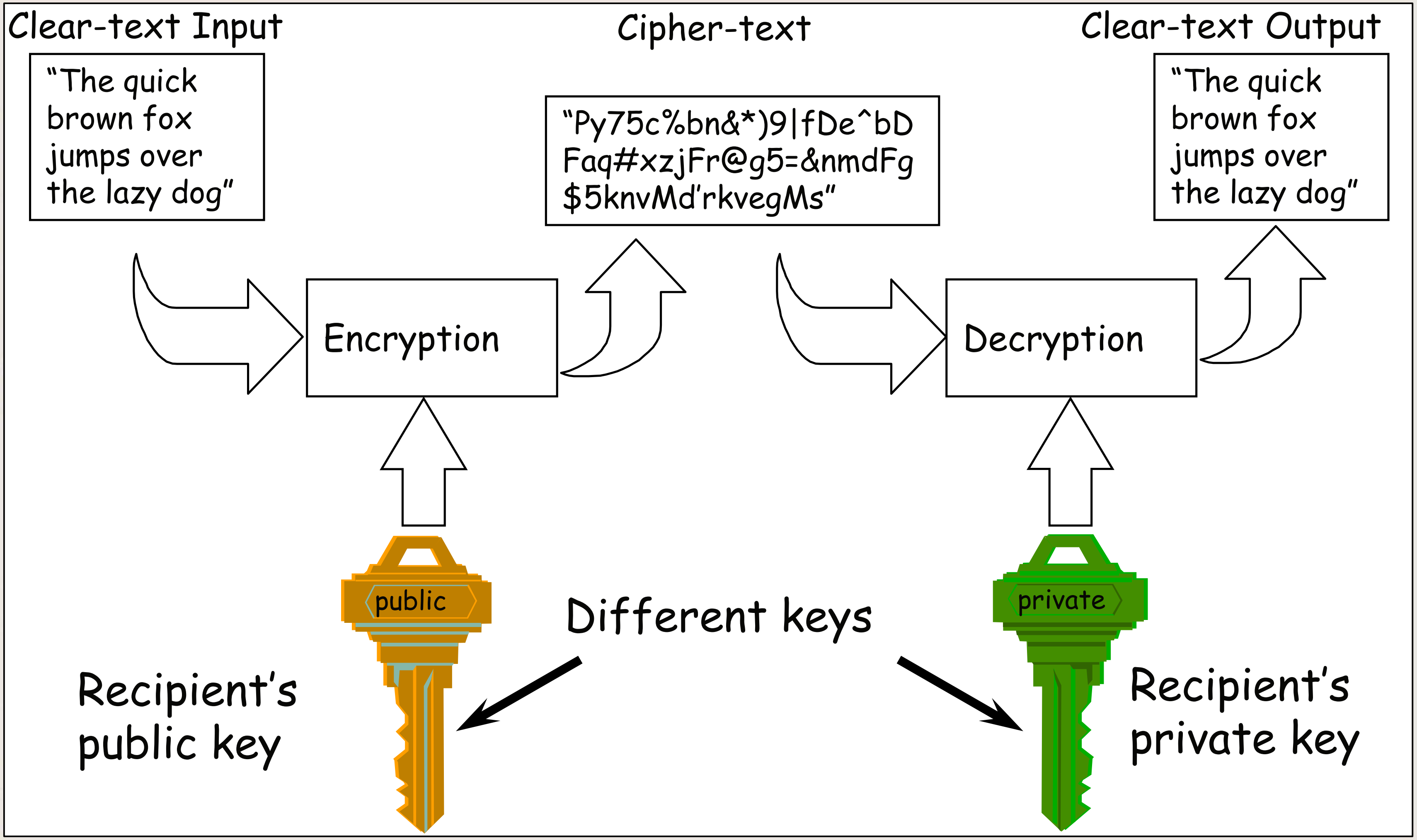
Scalability

Each pair of users needs a unique pair of keys, so the number of keys grow exponentially

Limited security

It can provide confidentiality, but limited authenticity and cannot provide non-repudiation.

Public Key (asymmetric)Encryption

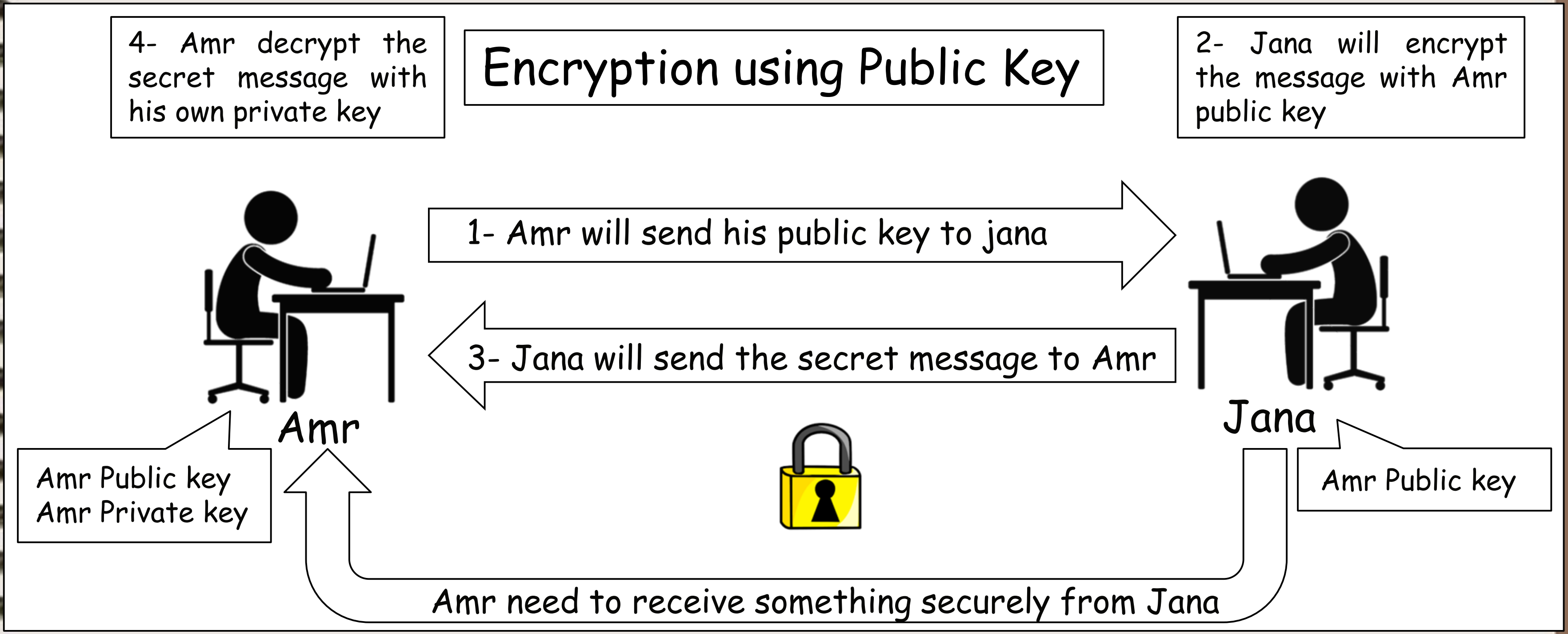

Clear-text Input
Cipher-text
Clear-text Output
"The quick brown fox jumps over the lazy dog"
"Py75c%bn&*)9|fDe^bD Faq#xzjFr@g5=&nmdFg $5knvMd'rkvegMs"
"The quick brown fox jumps over the lazy dog"
Encryption
Decryption
public
private
Different keys
Recipient's public key
Recipient's private key

Public Key Requirements

- Easy to create key pairs.
- Easy for sender knowing public key to encrypt messages.
- Easy for receiver knowing private key to decrypt ciphertext.
- Infeasible for opponent to determine private key from public key.
- Infeasible for opponent , knowing the public key, and a ciphertext to recover the original message.
- Either of the two related keys can be used for encryption, with the other used for decryption.

Public Key Encryption

Using two keys: Public key ----------- (Any one can get a copy of it)
Private key ---------- (Maintained be the owner)



Encryption using Public Key

4- Amr decrypt the secret message with his own private key

2- Jana will encrypt the message with Amr public key

1- Amr will send his public key to jana

3- Jana will send the secret message to Amr

Amr Public key
Amr Private key

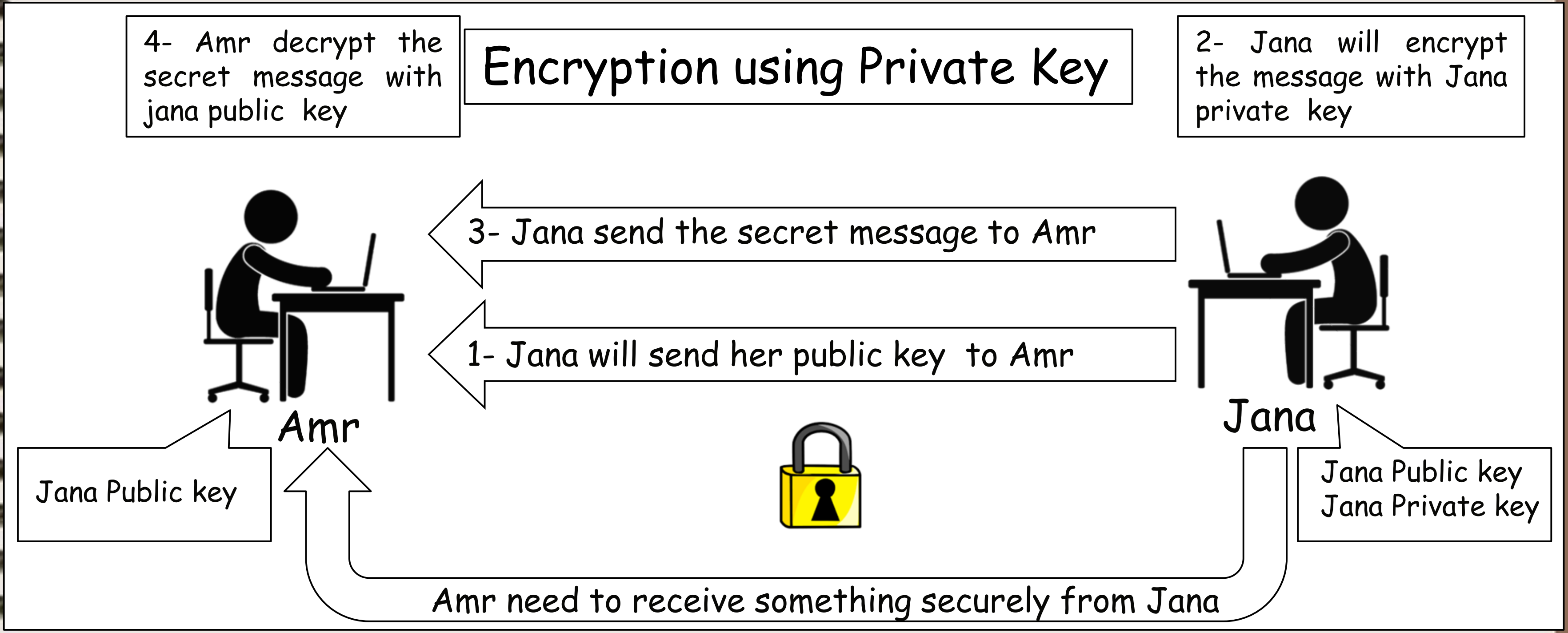
Amr Public key

Amr need to receive something securely from Jana

- ❑ Jana assumed the one who sent the public key is Amr not someone else
- ❑ Jana assured that Amr is the only one who can decrypt the message

Public Key Encryption

Using two keys: Public key ----------- (Any one can get a copy of it)
Private key ---------- (Maintained be the owner)



Encryption using Private Key

4- Amr decrypt the secret message with jana public key

2- Jana will encrypt the message with Jana private key

3- Jana send the secret message to Amr

1- Jana will send her public key to Amr

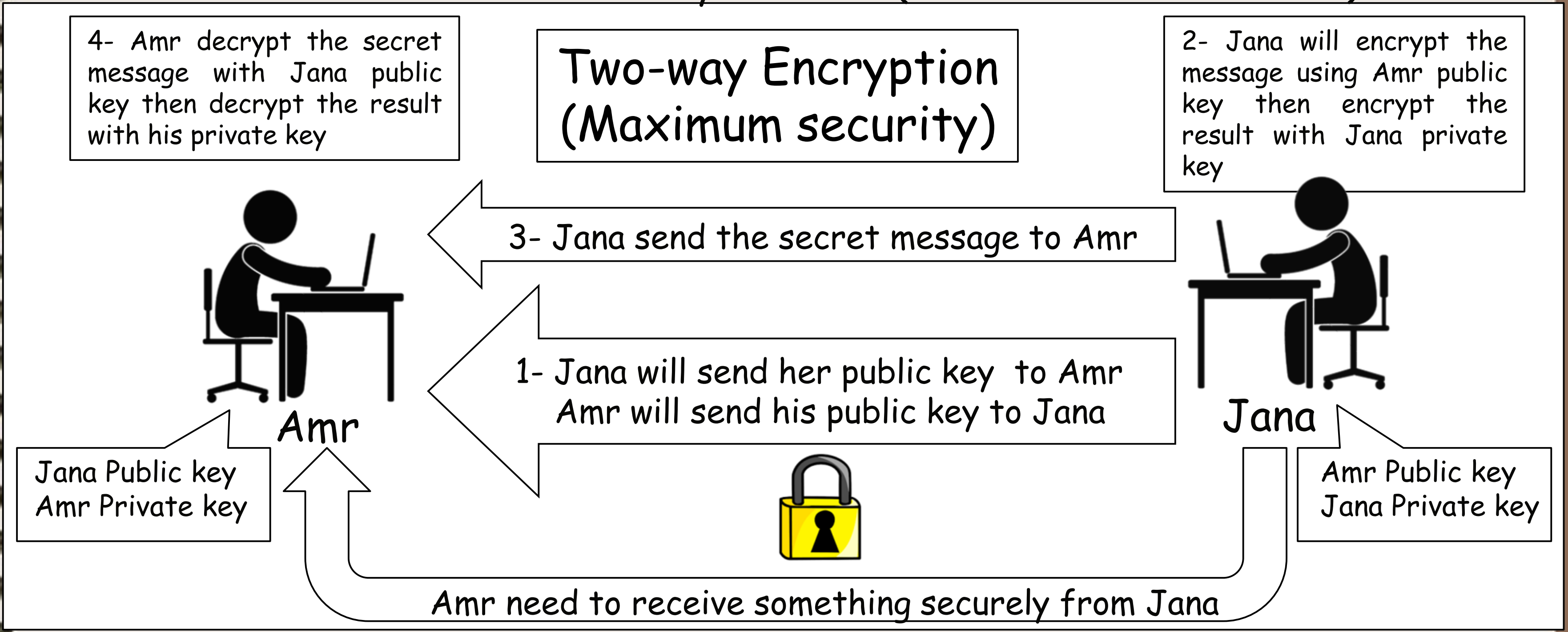
Amr need to receive something securely from Jana

- ❑ Amr assumed the one who sent the public key is Jana not someone else
- ❑ Amr assured that Jana is the one who encrypt the message

Public Key Encryption

Using two keys:

Public key ----------- (Any one can get a copy of it)
Private key ---------- (Maintained be the owner)



Two-way Encryption (Maximum security)

1- Jana will send her public key to Amr
Amr will send his public key to Jana

2- Jana will encrypt the message using Amr public key then encrypt the result with Jana private key

3- Jana send the secret message to Amr

4- Amr decrypt the secret message with Jana public key then decrypt the result with his private key

Amr

Jana Public key
Amr Private key

Jana

Amr Public key
Jana Private key

Amr need to receive something securely from Jana

- ❑ Jana assumed the one who sent the public key is Amr not someone else
- ❑ Amr assumed the one who sent the public key is Jana not someone else
- ❑ Jana assured that Amr is the only one who can decrypt the message
- ❑ Amr assured that Jana is the one who encrypt the message

Public Key Algorithms

Diffie-Hellman Algorithm

- ❑ Published in 1976 as the first public key algorithm used for key exchange algorithm.
- ❑ A number of commercial products employ it.
- ❑ Its purpose is to enable two users to exchange a secret key securely that can then be used for subsequent encryption of messages.
- ❑ The algorithm itself is limited to the exchange of the secret keys.

Public Key Algorithms

RSA (Rivest, Shamir, Adleman)

- ❑ Developed in 1977 by Rivest, Shamir, Adleman
- ❑ Only widely accepted public-key encryption.
- ❑ The security of RSA depends wholly on the problem of factoring large numbers.
- ❑ The public and private keys are functions of a pair of large (100 to 200 digits or even larger) prime numbers.
- ❑ Recovering the plaintext from the public key and the ciphertext is conjectured to be equivalent to factoring the product of the two primes.
- ❑ 1024-bit key size is considered strong enough.

Public Key Algorithms

RSA Encryption Algorithm

- ❑ Chose number ($n = p * q$))
  - ❖ p & q are two prime numbers,
  - ❖ p and q must remain secret.
- ❑ Public Key: Chose value of (e) < φ
  WHERE : $\varphi = (p - 1)(q - 1)$
  e is a prime number,
  e and n are coprime numbers.
- ❑ Private Key: Compute a value for (d) such that
  $(d * e) \bmod (\varphi) = 1$
- ❑ Encrypting: $c = m^e \bmod n$
- ❑ Decrypting: $m = c^d \bmod n$

e … Encryption key

d … Decryption key

m … plain message

c … cipher message

Public Key Algorithms

RSA Encryption Algorithm Example:

1) Chose key encryption pairs (e & d):

- ❑ *Chose p and q*
- ❑ *Let p = 47 and q = 71*
- ❑ *n = (p * q) = 3337, φ = (p-1)(q-1) = 3220*
- ❑ *Chose randomly e = 79*
- ❑ *d = e^{-1} mod (φ) = 79^{-1} mod 3220*
  = 1019
- ❑ *Public key is (e, n) : (79,3337)*
- ❑ *private key is (d, n) : (1019,3337)*

Public Key Algorithms

RSA Encryption Algorithm Example:

2) Enc./Dec. message (m) using key (e):

- □ *Let m = 688232687966683*
- □ *Break (m) it into small blocks. For example 3-digit blocks. The message is split into six blocks:*

$$m_1 = 688,\ m_2 = 232,\ m_3 = 687,\ m_4 = 966,\ m_5 = 668,\ m_6 = 003$$

- □ *To encrypt:* $m_1 = 688^{79} \bmod 3337 = 1570 = c_1$
- □ *Repeat for other blocks :*

$$c_2 = 2756,\ c_3 = 2019,\ c_4 = 2276,\ c_5 = 2423,\ c_6 = 158$$

- □ *Then c = 1570 2756 2091 2276 2423 158*
- □ *To decrypt:* $c_1 = 1570^{1019} \bmod 3337 = 688 = m_1$

Public Key Algorithms

RSA Encryption Algorithm Simple Example:

- ❑ Choose $p = 3$ and $q = 11$
- ❑ Compute $n = p * q = 3 * 11 = 33$
- ❑ Compute $\varphi = (p - 1) * (q - 1) = 2 * 10 = 20$
- ❑ Choose (e) such that $1 < e < \varphi$ Let $e = 7$
- ❑ Compute a value for (d) such that $(d * e) \% \varphi = 1$.
  One solution is $d = 3$ $[(3 * 7) \% 20 = 1]$
- ❑ Public key is (e, n) => (7, 33)
- ❑ Private key is (d, n) => (3, 33)
- ❑ The encryption of $m = 2$ is $c = 2^7 \% 33 = 29$
- ❑ The decryption of $c = 29$ is $m = 29^3 \% 33 = 2$

Public Key Algorithms

RSA Encryption Algorithm Comments:

- ❑ RSA is implemented on hardware. Many different hardware chips perform RSA encryption
- ❑ In hardware, DES is about 100 times faster than RSA.
- ❑ In software, DES is about 1000 times faster than RSA.

Remember: it is not enough to have a secure cryptographic algorithm. The entire cryptosystem must be secure, and the cryptographic protocol must be secure. A failure in any of those three areas makes the overall system insecure.

Public Key Algorithms

- ❑ Digital Signature Standard (DSS):
  Provides only a digital signature function (1991)

- ❑ Elliptic curve cryptography (ECC):
  New, security like RSA, but with much smaller keys.

- ❑ El-Gamal scheme for digital signature

Public Key Algorithms

- Applications for Public-Key Cryptosystems
  - Digital signatures.
  - Key management.
  - Key distribution.

Where is public-key enc. from data enc.?

Symmetric systems used for what?

Algorithm	Digital Signature	Symmetric Key Distribution	Encryption of secret key
RSA	Yes	Yes	Yes
Diffie-Hellman	No	Yes	No
DSS	Yes	No	No
Elliptic curve	Yes	Yes	Yes

Cryptanalysis attacks

Cryptanalysis attacks can be:

- □ Cipher text only attack.

  Giving algorithm & ciphertext, plaintext or key is extracted

- □ Known plaintext attack.

  Giving Plain text & ciphertext, key is required

- □ Chosen cipher text attack.

  Chosen pairs of ciphertext and Plaintext, key is extracted

- □ Chosen plaintext attack.

  Chosen pairs of Plaintext then ciphertext key is extracted

Cryptanalysis attacks

Cryptanalysis attacks can be:

Cipher text only attack (COA).

- ❑ An attack model for cryptanalysis where the attacker is assumed to know algorithm & ciphertext.
- ❑ The attack is completely successful if the corresponding plaintexts can be deduced (extracted) or, even better, the key.
- ❑ The ability to obtain any amount of information from the underlying ciphertext is considered a success

Cryptanalysis attacks

Cryptanalysis attacks can be:

Known plaintext attack (KPA)

- ❑ The attacker has samples of both the plaintext and its encrypted version then he can use them to expose further secret information after calculating the secret key.

Cryptanalysis attacks

Cryptanalysis attacks can be:

Chosen cipher text attack (CCA)

- ❑ Cryptanalyst is assumed to have a way to trick someone who knows the secret key into decrypting arbitrary message blocks and tell him the result, and he can do this a number of times.
- ❑ The cryptanalyst is successful if he has a significant chance of being able to deduce the key

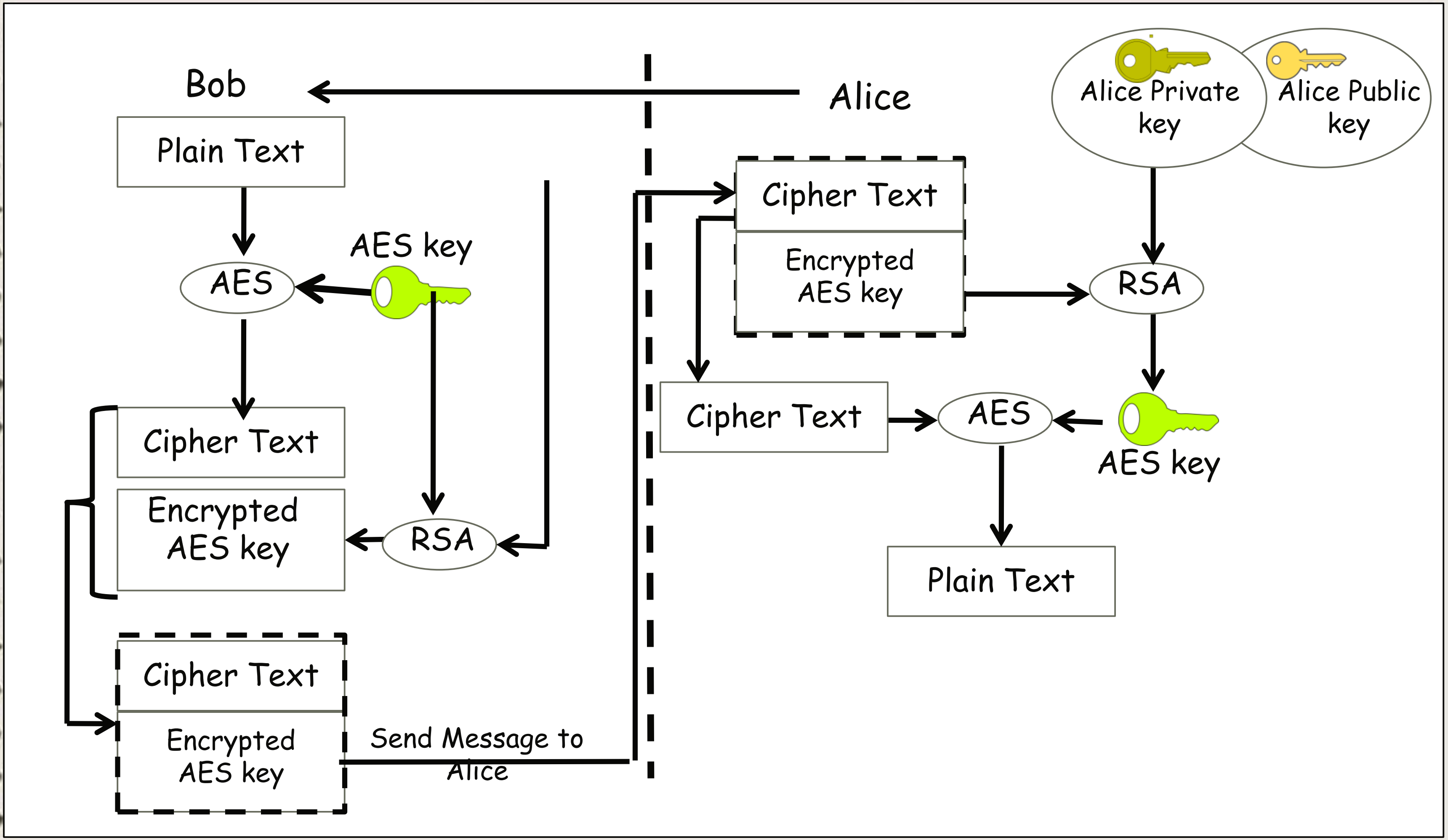
Cryptanalysis attacks

Cryptanalysis attacks can be:

Chosen plaintext attack (CPA).

- ❑ The attacker can choose random plaintexts to be encrypted and obtain the corresponding ciphertexts.
- ❑ The goal of the attack is to gain some further information which reduces the security of the encryption scheme.
- ❑ In the worst case, a chosen-plaintext attack could expose secret information after calculating the secret key.

Hybrid Encryption (Symmetric + Asymmetric) (Real world)



How the system work

Hybrid Encryption (Symmetric + Asymmetric) (Real world)

❑ Hybrid encryption incorporates a combination of asymmetric and symmetric encryption to benefit from the strengths of each form of encryption. These strengths are respectively defined as speed and security.

❑ Hybrid encryption is considered a highly secure type of encryption as long as the public and private keys are fully secure.

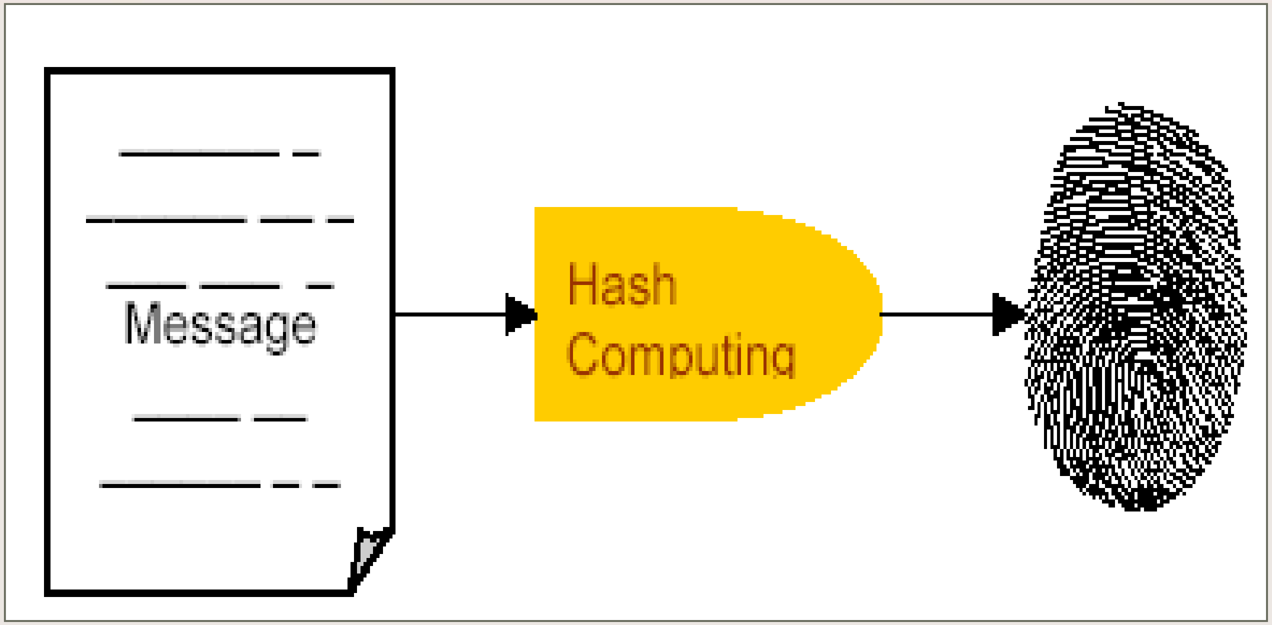
❑ Hybrid encryption is achieved through data encryption using unique session key along with symmetrical encryption.

❑ The recipient then uses the public key encryption method to decrypt the symmetric key.

❑ Once the symmetric key is recovered, it is then used to decrypt the message.

Hash Function
“No key Encryption”

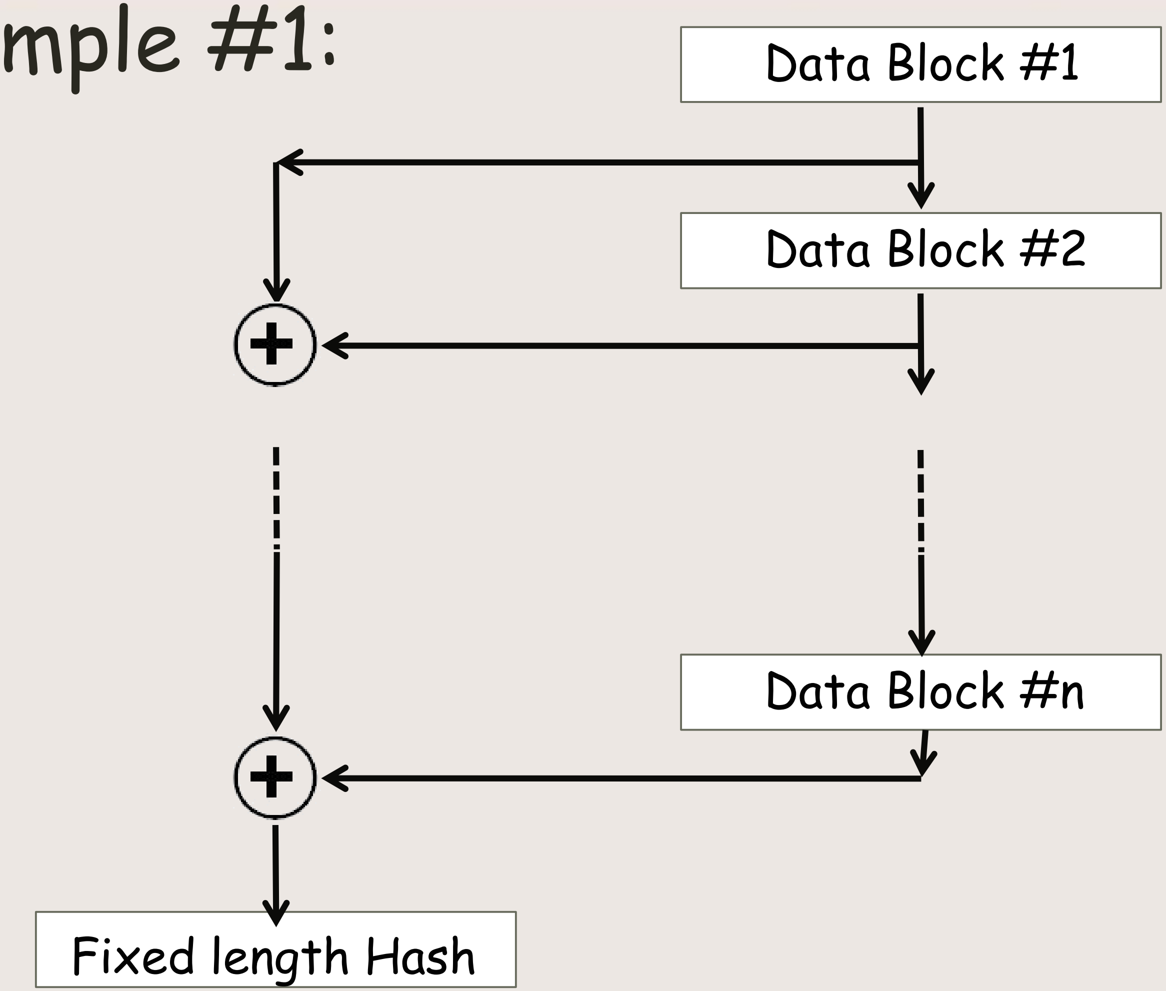
One-Way Hash Function



- ❑ Hashing used to obtain a "digital fingerprint" (hash) for a given message.
- ❑ The hash code has a fixed-length (normally 128 or 160 bits) and it's designed to be unique

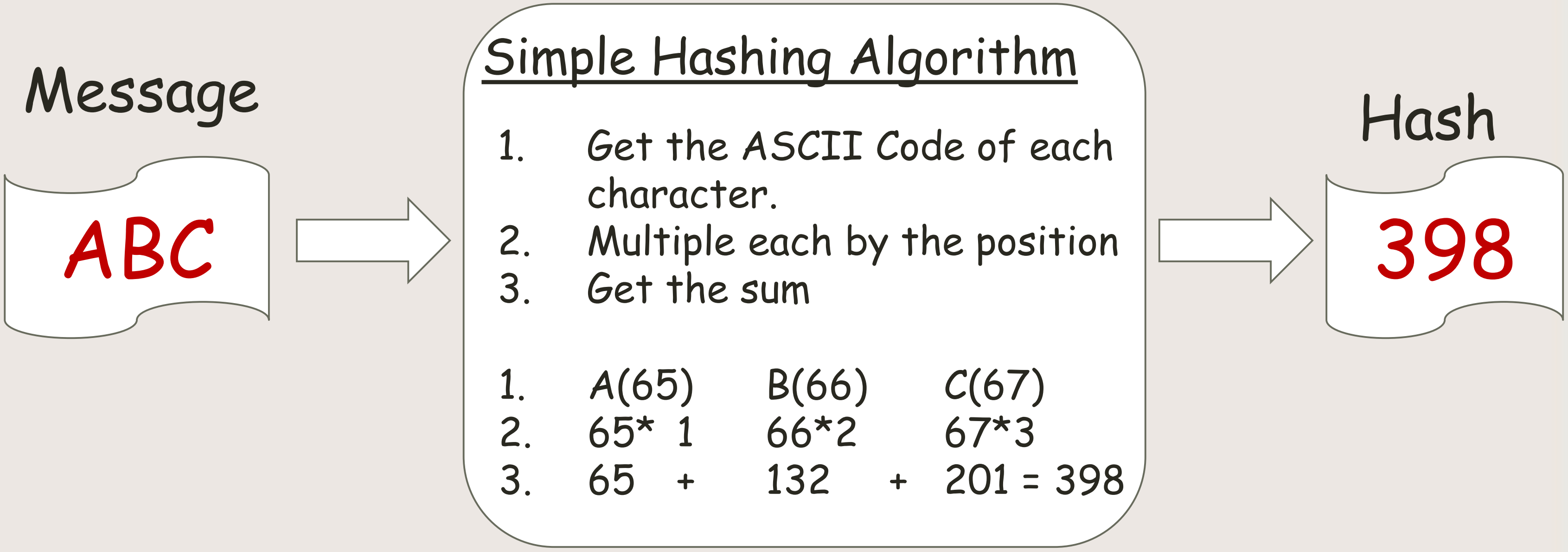
One-Way Hash Function

❑ Example #1:



One-Way Hash Function

- ❑ Example #2:



Message

ABC

Simple Hashing Algorithm

1. Get the ASCII Code of each character.
2. Multiple each by the position
3. Get the sum

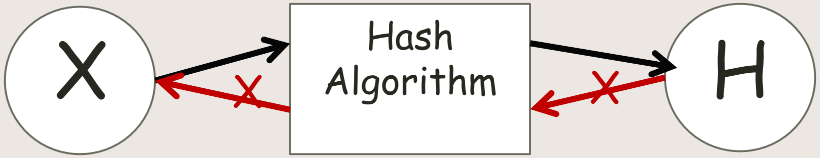
1.	A(65)		B(66)		C(67)	
2.	65* 1		66*2		67*3	
3.	65	+	132	+	201	= 398

Hash

398

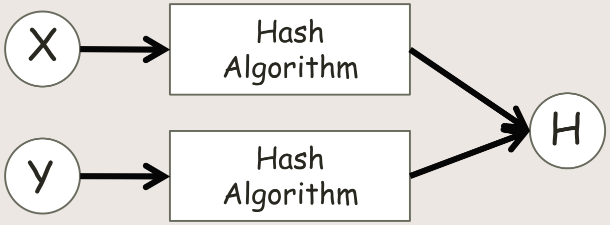
One-Way Hash Function

Properties

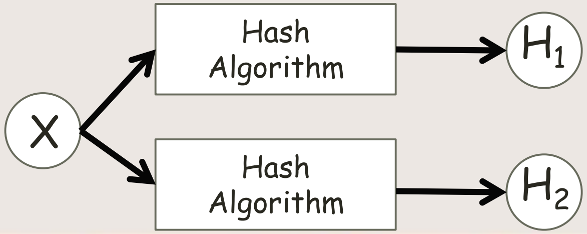


- ☐ One-way property.
- ☐ Applied to any size data.
- ☐ H(X) produces a fixed-length output.
- ☐ H(X) is relatively easy to compute for any given X.

- ☐ No way to have the same hash value (H) for two different messages (X, Y).



- ☐ No way to find two different hash values (H_1, H_2) for a message X.



One-Way Hash Function

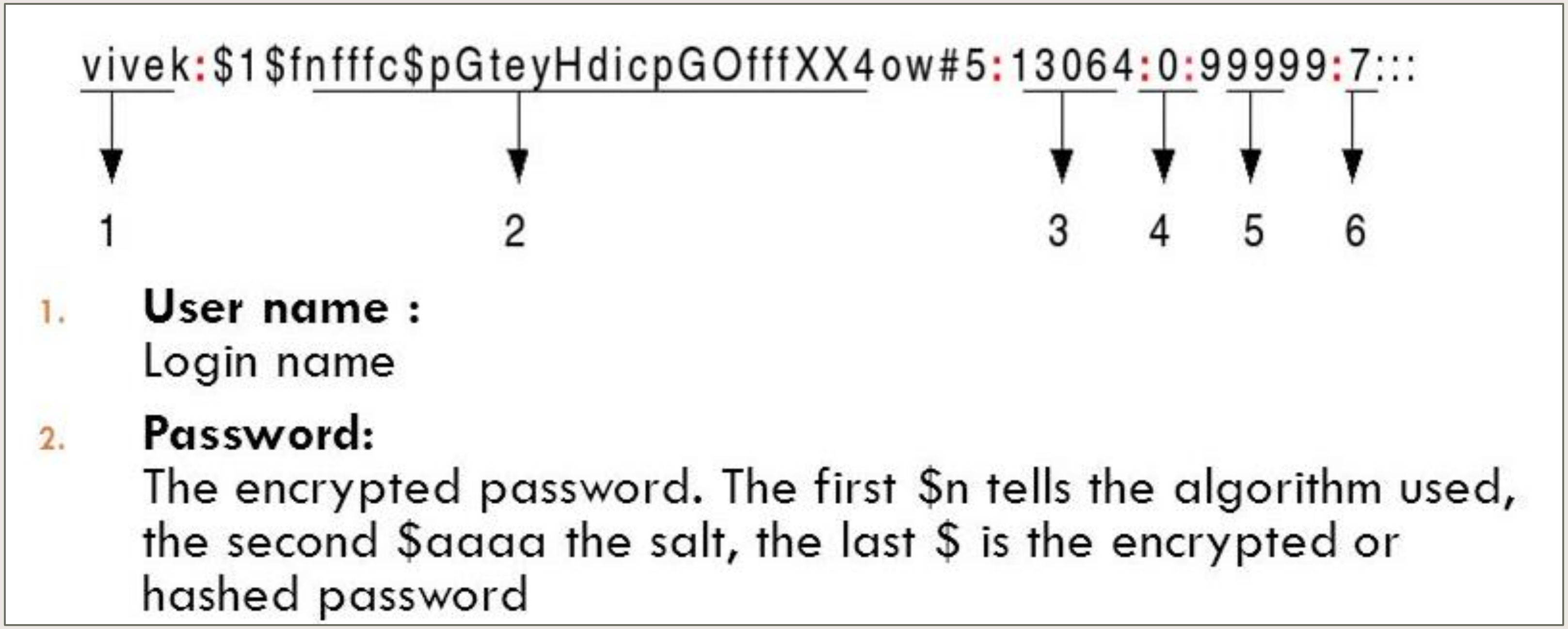
- ❑ Famous Hashing Algorithms
  - ❖ SHA-1 gives 160-bit hash.
  - ❖ SHA-256, SHA-384, SHA-512 provide improved size and security.
  - ❖ MD2, MD4, MD5 (128 bit)
  - ❖ HAVAL (variation of MD5; variable length)

One-Way Hash Function

❑ Usage: Securing passwords

Any OS system that requires password authentication must contain a database of passwords, usually hashed

Unix OS keep user info in file: \etc\passwd



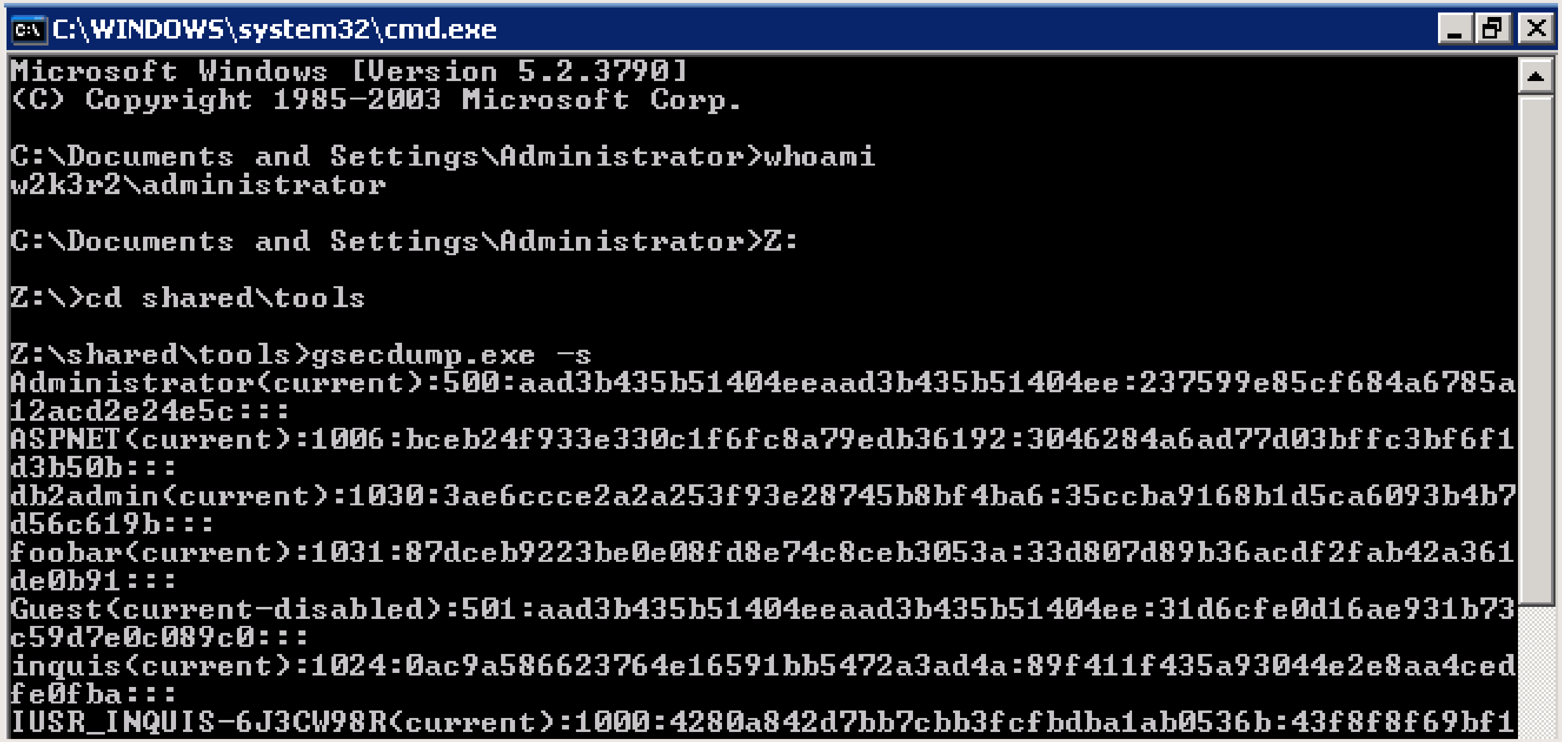
```
vivek:$1$fnfffc$pGteyHdicpGOfffXX4ow#5:13064:0:99999:7:::
```

1. **User name :**
   Login name
2. **Password:**
   The encrypted password. The first $n tells the algorithm used, the second $aaaa the salt, the last $ is the encrypted or hashed password

One-Way Hash Function

❑ Usage: Securing passwords

Windows OS keep user info in file: C:\windows\system32\config\SAM



```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>whoami
w2k3r2\administrator

C:\Documents and Settings\Administrator>Z:

Z:\>cd shared\tools

Z:\shared\tools>gsecdump.exe -s
Administrator(current):500:aad3b435b51404eeaad3b435b51404ee:237599e85cf684a6785a
12acd2e24e5c:::
ASPNET(current):1006:bceb24f933e330c1f6fc8a79edb36192:3046284a6ad77d03bffc3bf6f1
d3b50b:::
db2admin(current):1030:3ae6ccce2a2a253f93e28745b8bf4ba6:35ccba9168b1d5ca6093b4b7
d56c619b:::
foobar(current):1031:87dceb9223be0e08fd8e74c8ceb3053a:33d807d89b36acdf2fab42a361
de0b91:::
Guest(current-disabled):501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73
c59d7e0c089c0:::
inquis(current):1024:0ac9a586623764e16591bb5472a3ad4a:89f411f435a93044e2e8aa4ced
fe0fba:::
IUSR_INQUIS-6J3CW98R(current):1000:4280a842d7bb7cbb3fcfbdba1ab0536b:43f8f8f69bf1
```

(SAM) Security Accounts Manager

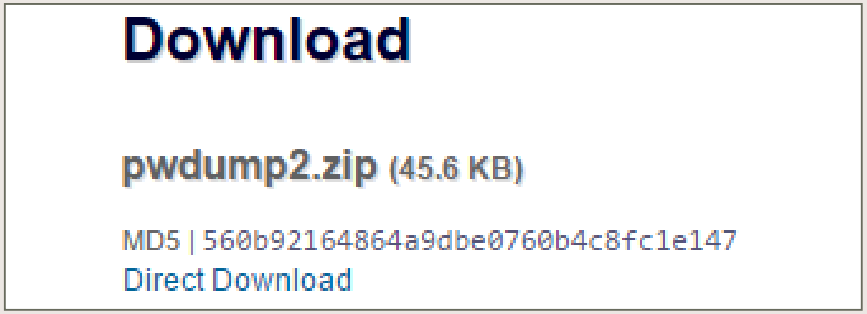
One-Way Hash Function

- ❑ Usage: File integrity

Example :OpenOffice.org MD5 sums

```
e0d123e5f316bef78bfdf5a008837577  OOo_2.0.1_LinuxIntel_install.tar.gz
35d91262b3c3ec8841b54169588c97f7  OOo_2.0.1_LinuxIntel_install_wJRE.tar.gz
cc273fe9d442850fa18c31c88c823e07  OOo_2.0.1_SolarisSparc_install.tar.gz
ff6626c69507a6f511cc398998905670  OOo_2.0.1_SolarisSparc_install_wJRE.tar.gz
ce099d7e208dc921e259b48aadef36c1  OOo_2.0.1_Solarisx86_install.tar.gz
4fb319211b2e85cace04e8936100f024  OOo_2.0.1_Solarisx86_install_wJRE.tar.gz
66bd00e43ff8b932c14140472c4b8cc6  OOo_2.0.1_Win32Intel_install.exe
2d86c4246f3c0eb516628bf324d6b9a3  OOo_2.0.1_Win32Intel_install_wJRE.exe
```

Example :download from internet



Download

pwdump2.zip (45.6 KB)

MD5 | 560b92164864a9dbe0760b4c8fc1e147

Direct Download

Breaking Hash Functions

❑ Attack approaches

❖ Cryptanalysis:

exploit logical weakness is algorithm.

❖ Brute-force attack:

- Trial many inputs.
- Strength proportional to size of hash code.
- Use more computer processing time and less storage than Rainbow Table attack

❖ Rainbow Table:

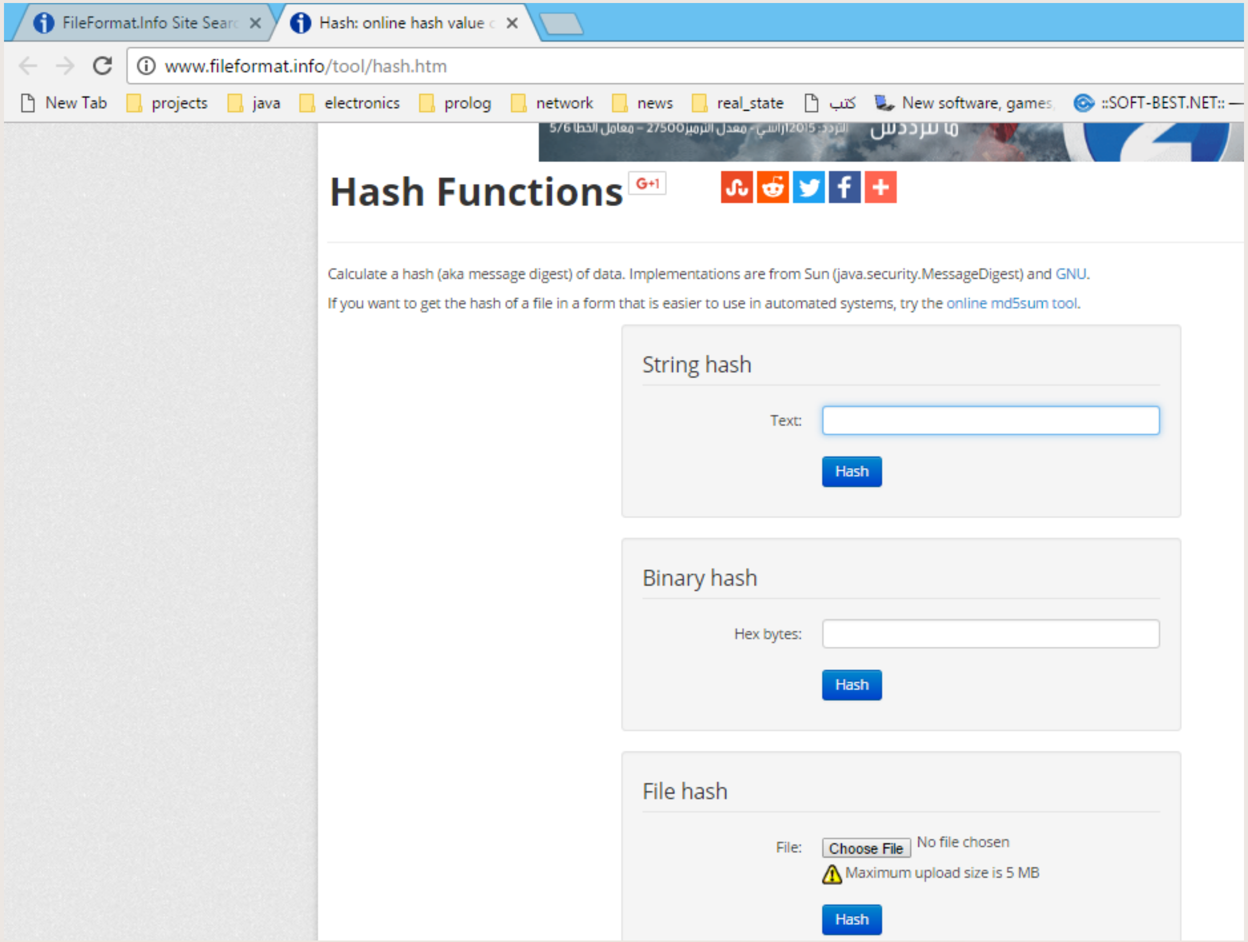
Breaking Hash Functions

❑ Rainbow Table:

- ❖ Is a pre-computed table for reversing cryptographic hash functions, usually for cracking password hashes.
- ❖ Usually used in recovering a plaintext password up to a certain length consisting of a limited set of characters
- ❖ Once an attacker gains access to a system's password database, the password cracker compares the rainbow table's precompiled list of potential hashes to hashed passwords in the database.
- ❖ Use less computer processing time and more storage than a brute-force attack which calculates a hash on every attempt

Breaking Hash Functions

- ❑ To calculate Hash value for a given string:



Hash Functions

Calculate a hash (aka message digest) of data. Implementations are from Sun (java.security.MessageDigest) and GNU.

If you want to get the hash of a file in a form that is easier to use in automated systems, try the online md5sum tool.

String hash

Text:

Hash

Binary hash

Hex bytes:

Hash

File hash

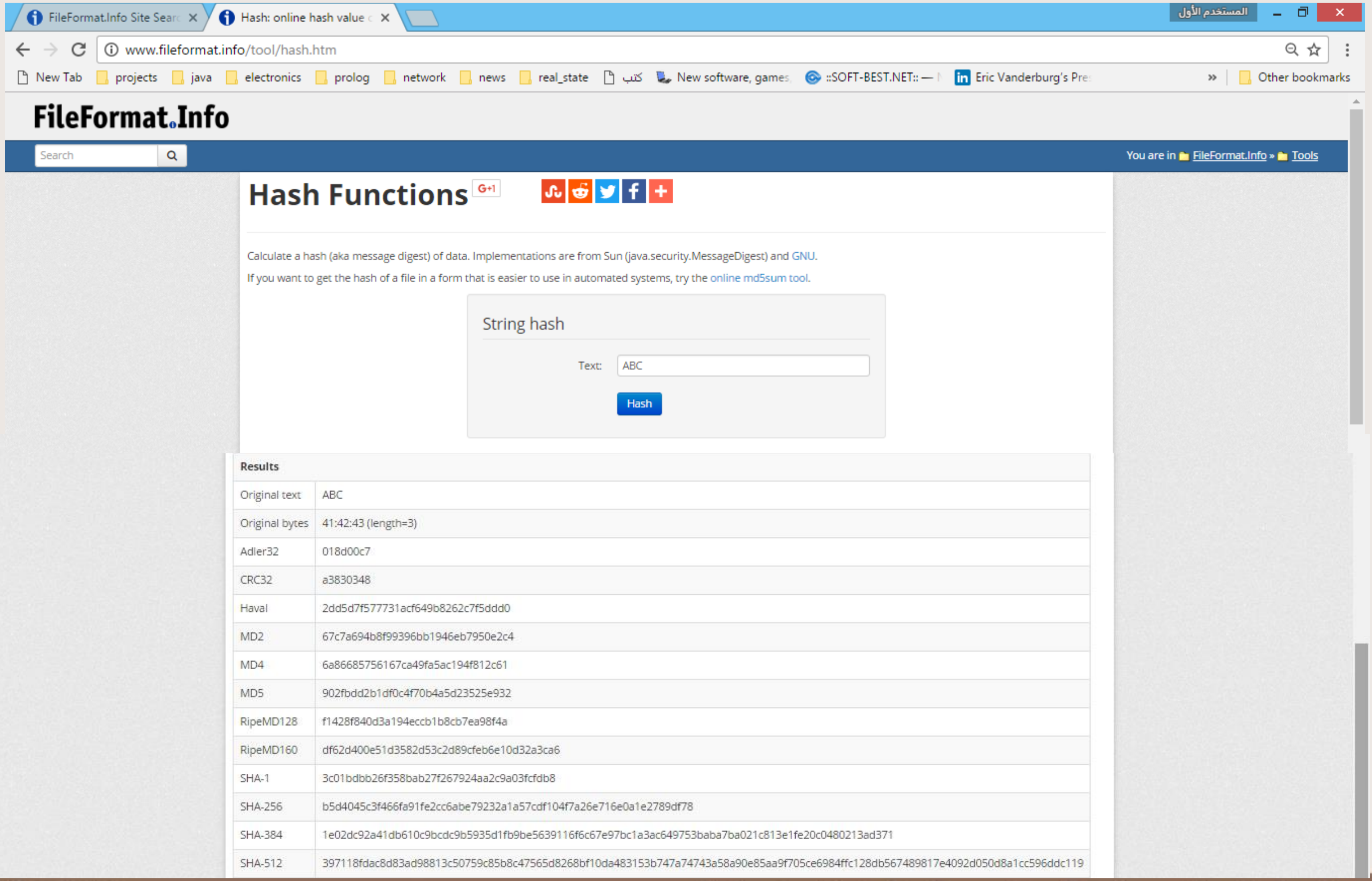
File: Choose File No file chosen

⚠ Maximum upload size is 5 MB

Hash

Breaking Hash Functions

- ❑ To calculate Hash value for a given string:



Hash Functions

Calculate a hash (aka message digest) of data. Implementations are from Sun (java.security.MessageDigest) and GNU.

If you want to get the hash of a file in a form that is easier to use in automated systems, try the online md5sum tool.

String hash

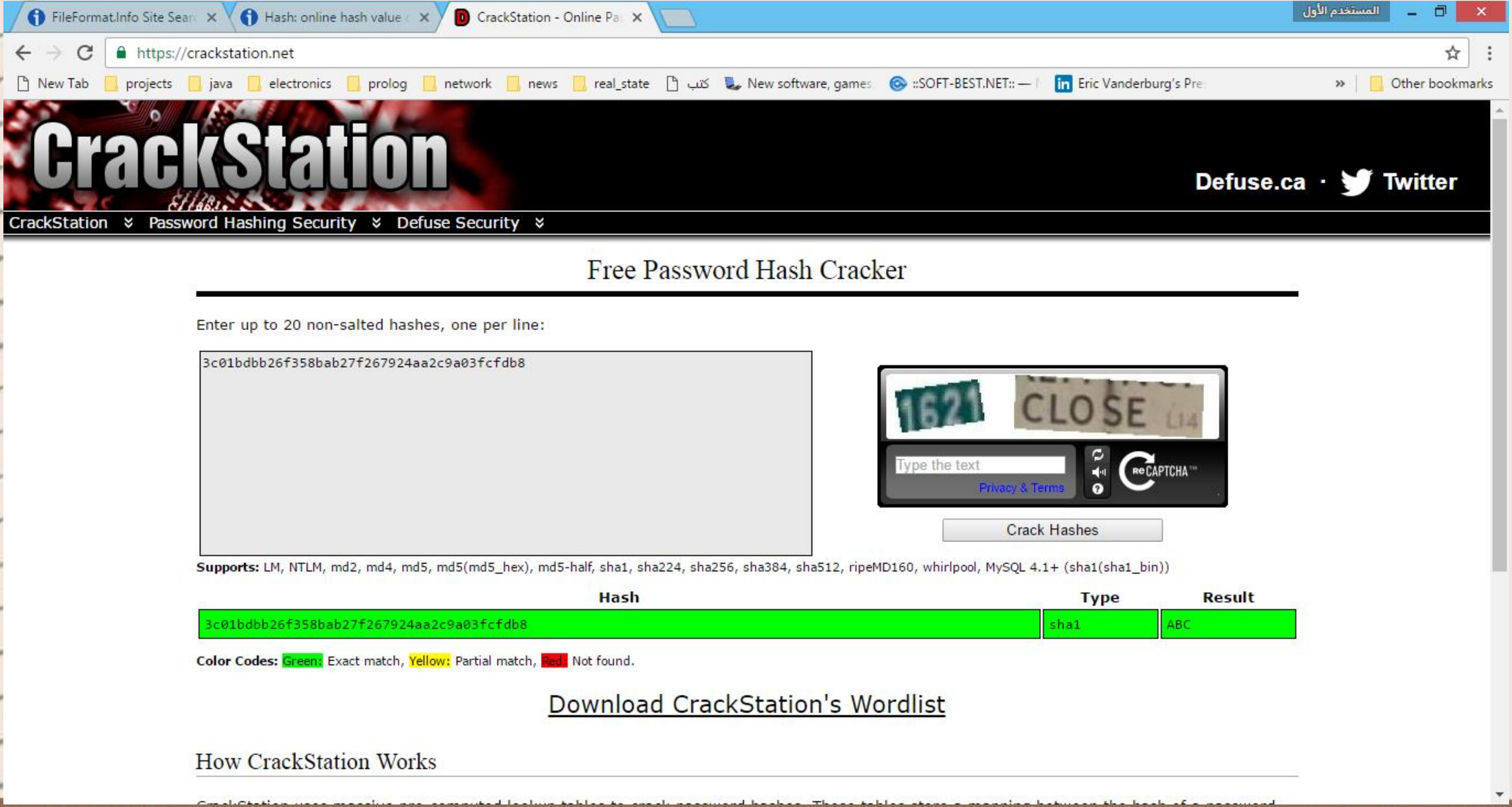
Text: ABC

Hash

Results	
Original text	ABC
Original bytes	41:42:43 (length=3)
Adler32	018d00c7
CRC32	a3830348
Haval	2dd5d7f577731acf649b8262c7f5ddd0
MD2	67c7a694b8f99396bb1946eb7950e2c4
MD4	6a86685756167ca49fa5ac194f812c61
MD5	902fbdd2b1df0c4f70b4a5d23525e932
RipeMD128	f1428f840d3a194eccb1b8cb7ea98f4a
RipeMD160	df62d400e51d3582d53c2d89cfeb6e10d32a3ca6
SHA-1	3c01bdbb26f358bab27f267924aa2c9a03fcfdb8
SHA-256	b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78
SHA-384	1e02dc92a41db610c9bcdc9b5935d1fb9be5639116f6c67e97bc1a3ac649753baba7ba021c813e1fe20c0480213ad371
SHA-512	397118fdac8d83ad98813c50759c85b8c47565d8268bf10da483153b747a74743a58a90e85aa9f705ce6984ffc128db567489817e4092d050d8a1cc596ddc119

Breaking Hash Functions

- ❑ Reverse engineer Hash value using Rainbow Table:



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
3c01bdbb26f358bab27f267924aa2c9a03fcfdb8
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin))

Hash	Type	Result
3c01bdbb26f358bab27f267924aa2c9a03fcfdb8	sha1	ABC

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password

Authentication

Authentication

- ❑ Encryption protects against passive attack (eavesdropping).
- ❑ A different requirement is to protect against active attack (falsification/modification of data and transactions). Protection against such attacks is known as message or data authentication (observe data integrity objective).
- ❑ A message, file, document, or other collection of data is said to be authentic when it is genuine and came from its alleged source.
- ❑ Authentication allows communicating parties to verify that received or stored messages are authentic :

1. The contents of the message have not been altered (integrity)
2. The source is authentic (authentication)
3. It has not been artificially delayed and replayed
4. Sequence relative to other messages flowing between two parties)

Authentication Using Symmetric Encryption

- ❑ Symmetric encryption can be used to ensure authentication
- ❑ If we assume that only the sender and receiver share a key, then only the genuine sender would be able to encrypt a message successfully for the other participant, provided the receiver can recognize a valid message.
- ❑ Symmetric encryption alone is not a suitable tool for authentication , extra data may be added to the message:
  - error-detection code,
  - a sequence number,
  - a timestamp,

  The receiver is assured that no alterations have been made and that sequencing is proper.

Remember authentication is an integrity objective

Authentication without Message Encryption

- ❑ There are several approaches to message authentication that do not rely on message encryption
- ❑ An authentication tag (Signature, digest) is generated and appended to each message for transmission.
- ❑ The message itself is not encrypted and can be read at the destination independent of the authentication function at the destination.
- ❑ In this case integrity not confidentiality of the message is provided.
- ❑ Many situations message authentication without confidentiality is preferable:

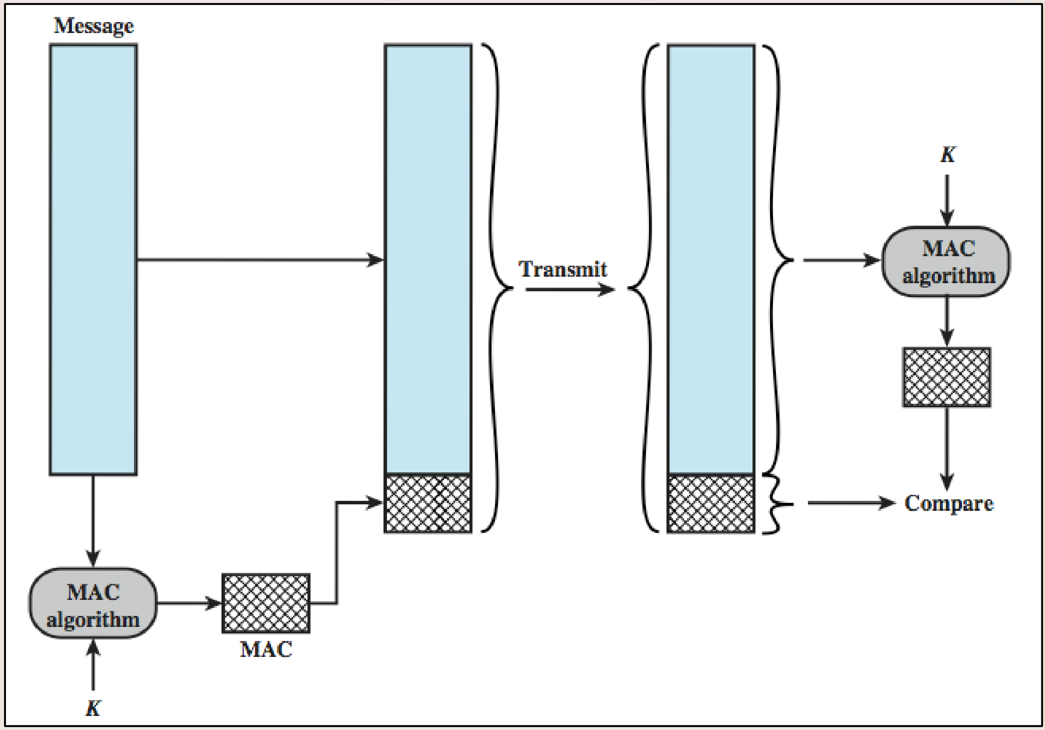
  1- Message is broadcast to many destination

  2- If one side has a heavy load and cannot afford the time to decrypt all incoming messages

  3- Authentication of a computer program in plaintext is an attractive service

Message Authentication Codes (MAC)

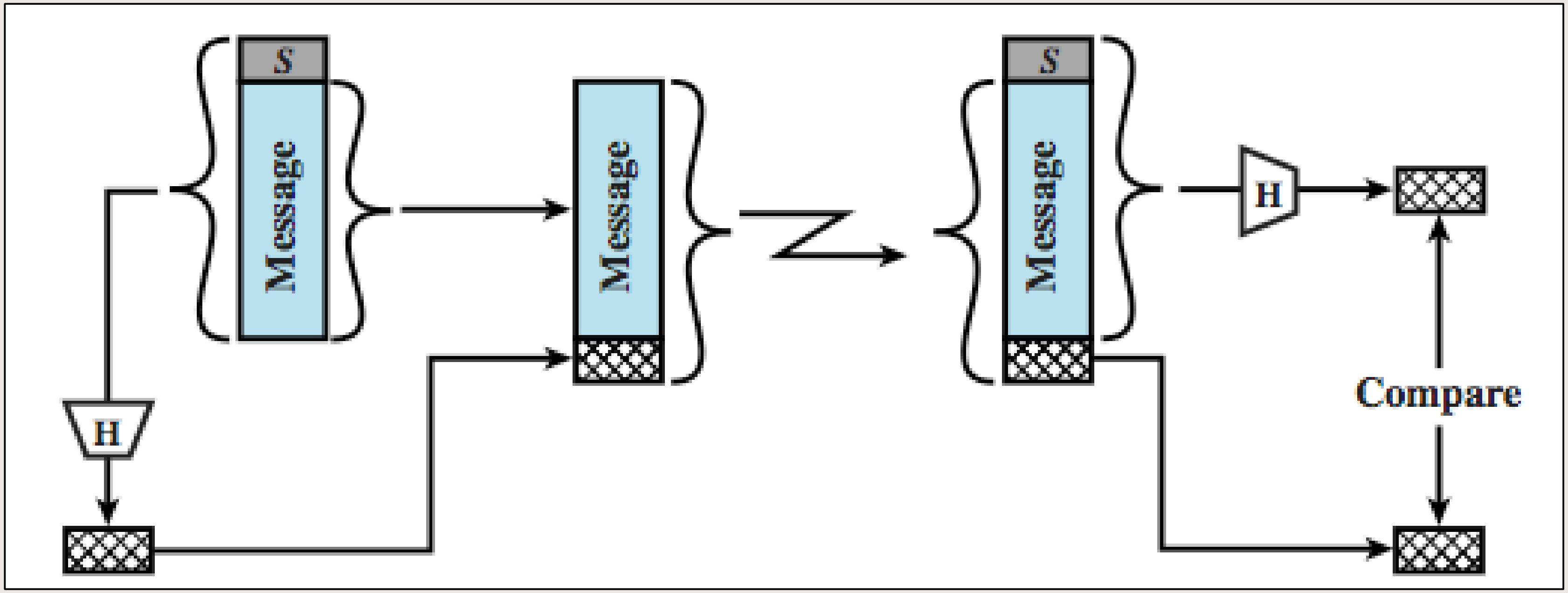
- ❑ This technique (not algorithm) involves the use of a secret key to generate a small block of data, known as a message authentication code (message digest), that is appended to the message



- ❑ DES is used to generate an encrypted version of the message, and the last number of bits of ciphertext are used as the code. A 16- or 32-bit code is typical

Message Authentication

- ❑ Using secret value & hash value

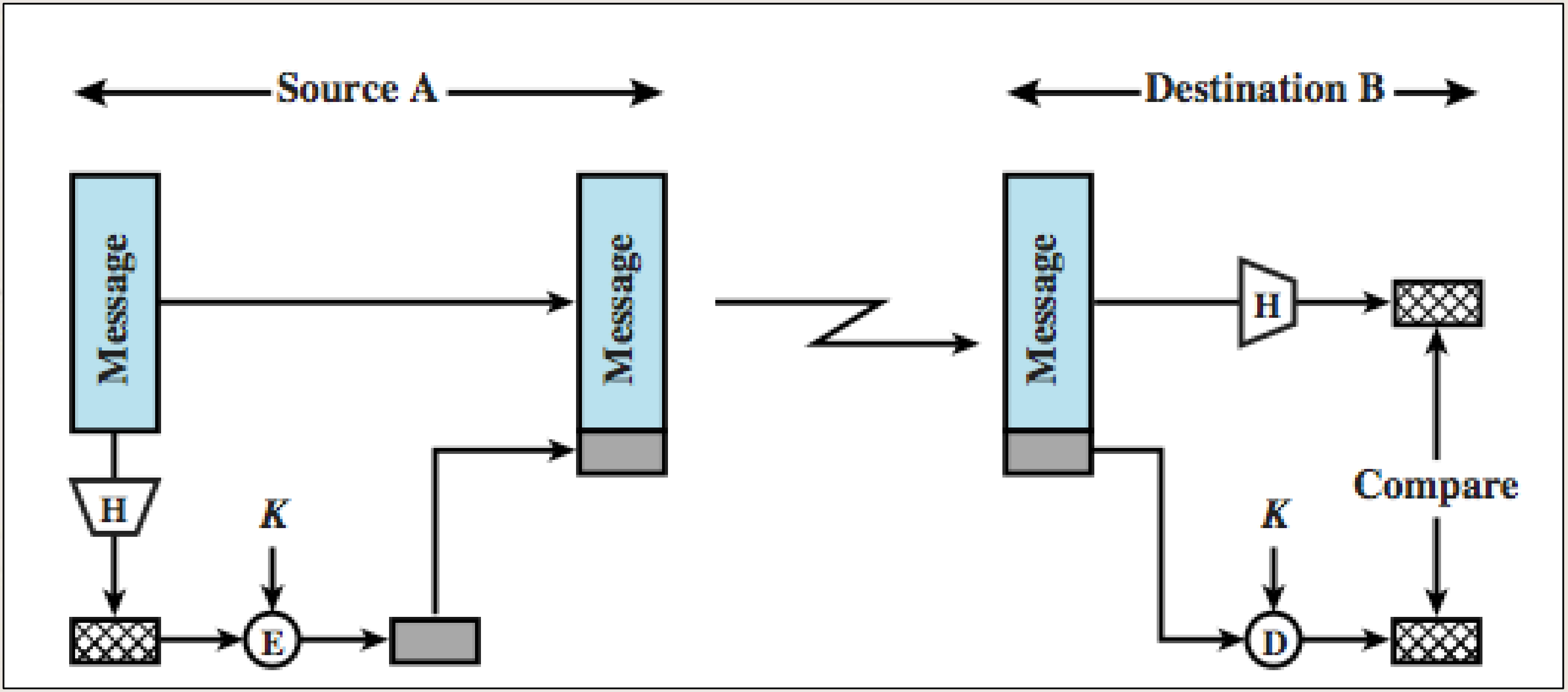


Drawback of this technique :

Secret value (key) must be sent before sending the message

Message Authentication

❑ Using symmetric encryption & hash value

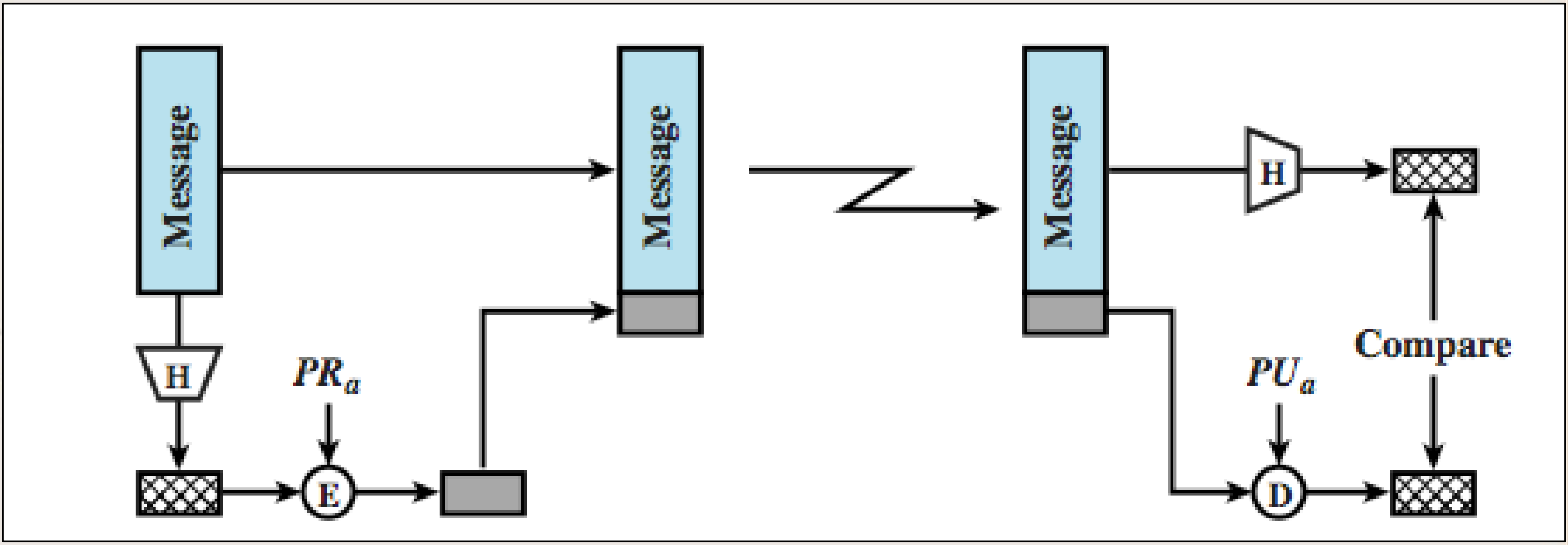


Drawback of this technique :

Secret key must be sent before sending the message

Message Authentication

- ❑ Using asymmetric encryption & hash value



Two outcomes of using this technique :

1- Message is authentic (genuine/not altered)

2- Sender is authentic (Sender non-repudiation)

No secret key (value) needed to communicate

DIGITAL SIGNATURES
&
KEY MANAGEMENT

DIGITAL SIGNATURES & KEY MANAGEMENT

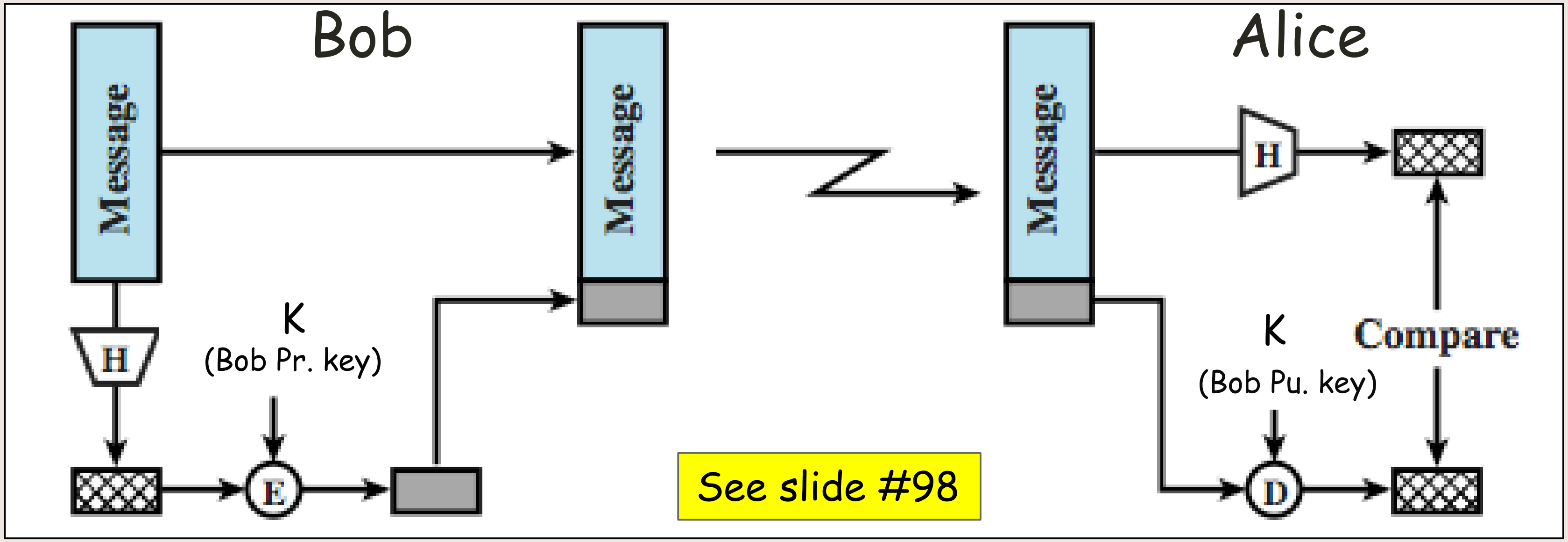
Slide #69

Public-Key Cryptosystems applications:

- ❑ Digital signature.
- ❑ Key management & distribution:
  - ❖ The secure distribution of keys
  - ❖ The use of public-key encryption to distribute secret keys
  - ❖ The use of public-key encryption to handle temporary key (session key) for message encryption

Digital Signature

- ❑ Bob uses a secure hash function to generate a hash value for the message
- ❑ Encrypts the hash code with his private key, creating a digital signature.
- ❑ Sends the message with the signature attached.

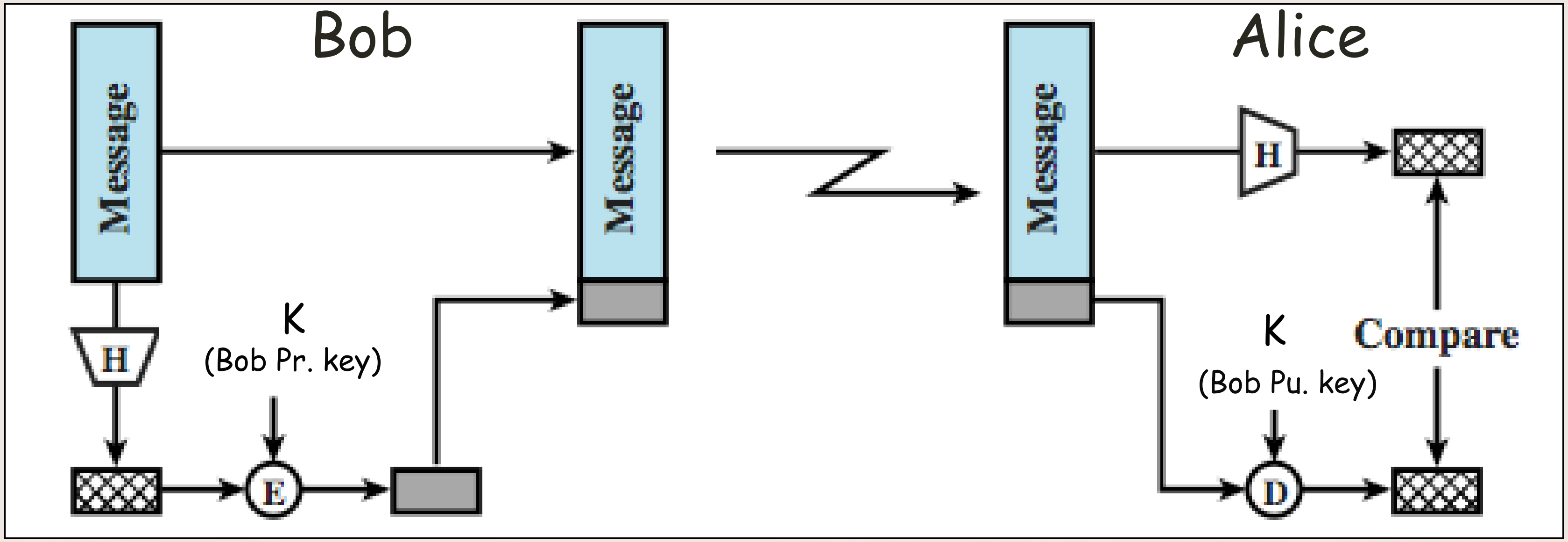


See slide #98

Digital Signature

When Alice receives the message plus signature, she:

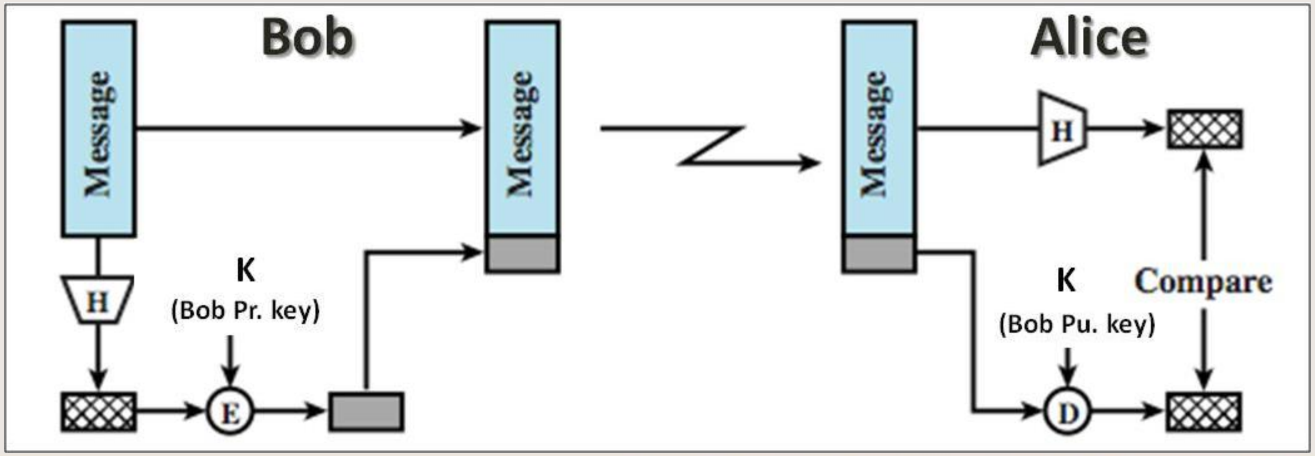
- ❑ Calculates a hash value for the message.
- ❑ Decrypts the signature using Bob's public key.
- ❑ Compares the calculated hash value to the decrypted hash value.



Digital Signature

When Alice receives the message plus signature, she:

- ❑ If the two hash values match, Alice is assured that the message must have been signed by Bob. (Authentication)
- ❑ No one else has Bob's private key and therefore no one else could have created a ciphertext that could be decrypted with Bob's public key. (non-repudiation)
- ❑ In addition, it is impossible to alter the message without access to Bob's private key. (Integrity)



Distribution of Public Keys

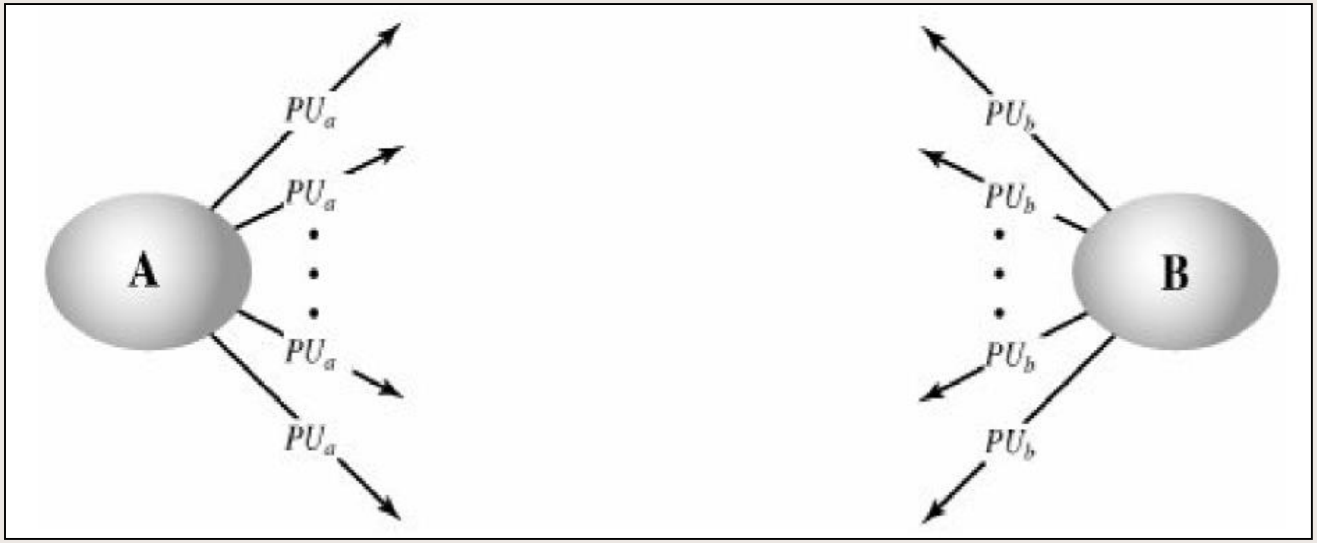
Keys can be distributed using one of:

- ❑ Public announcement
- ❑ Publicity available directory
- ❑ Public key authority.
- ❑ Public key certificates

Distribution of Public Keys

Public announcement

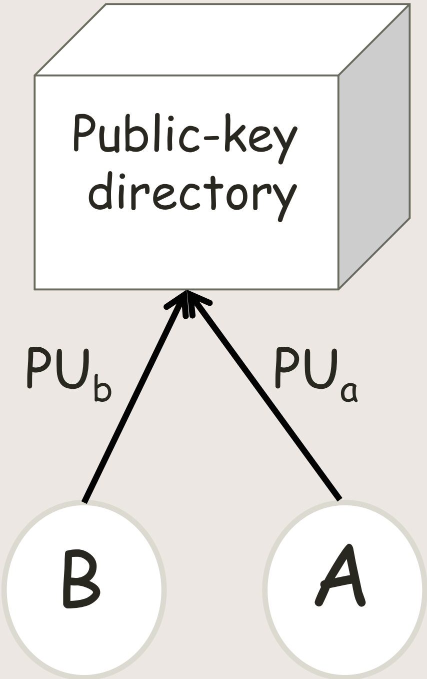
- ❑ Users distribute keys to recipients or broadcast to community at large
- ❑ Major weakness is forgery (anyone can create a key claiming to be someone else and broadcast it)



Distribution of Public Keys

Publicity available directory

- ❑ Can obtain greater security by registering keys with a public directory.
- ❑ Directory must be trusted with properties :
  - ❖ Contains name, public key entries
  - ❖ Participants can replace key securely with directory
  - ❖ Participants can replace key at any time
  - ❖ Directory is periodically published
  - ❖ Directory can be accessed electronically
- ❑ Still vulnerable to tampering forgery

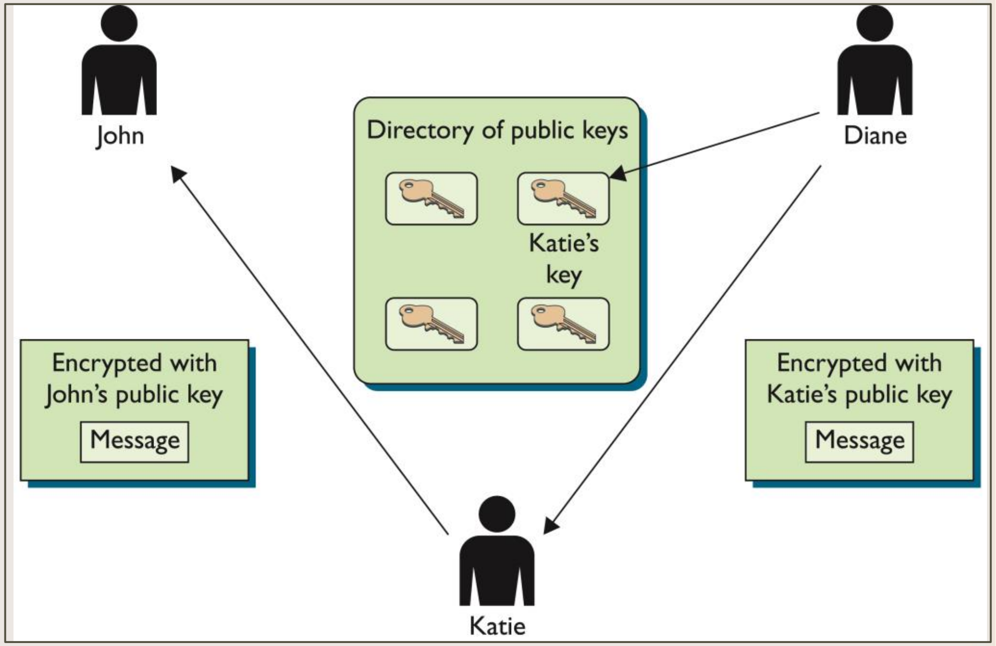


Can You Trust That Key?

Distribution of Public Keys

Publicity available directory attack

Man-in-the-Middle Attack

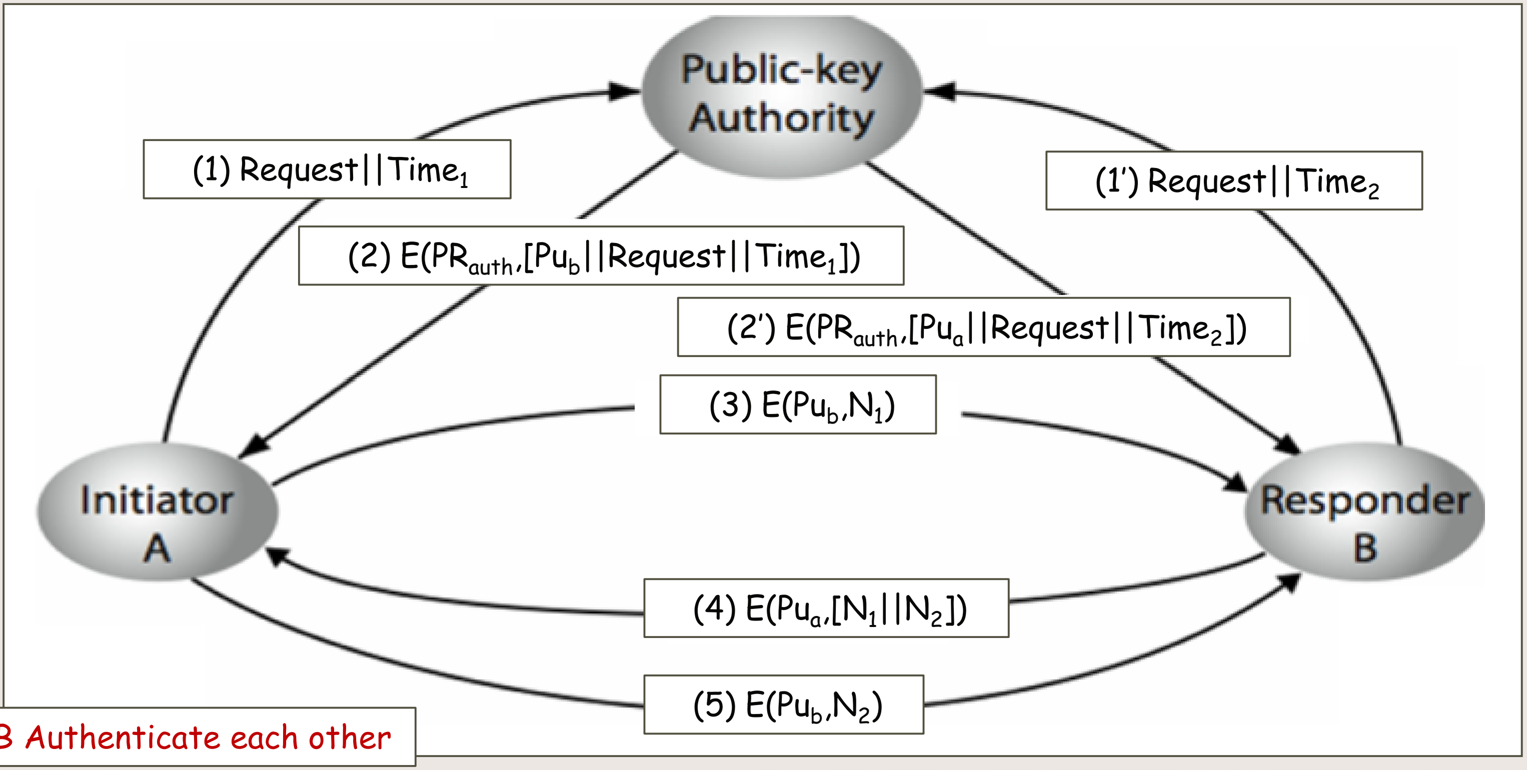

John
Directory of public keys
Diane
Katie's key
Encrypted with John's public key
Message
Encrypted with Katie's public key
Message
Katie

- ❑ Katie replaces John's public key with her key in the public accessible directory
- ❑ Diane extracts what she thinks is John's key, but it is in fact Katie's key.
- ❑ Katie can now read all encrypted messages from Diane to John.
- ❑ After Katie decrypts and reads Diane's message, she encrypts it with John's real public key and sends it on to him so he will not notice the attack.

Distribution of Public Keys

Public key authority.

- ❑ Improve security by tightening control over distribution of keys
- ❑ Has properties of directory
- ❑ Assume that a central authority maintains a dynamic directory of all participants



Public-key Authority

Initiator A

Responder B

(1) Request||Time$_1$

(1') Request||Time$_2$

(2) E(PR$_{auth}$,[Pu$_b$||Request||Time$_1$])

(2') E(PR$_{auth}$,[Pu$_a$||Request||Time$_2$])

(3) E(Pu$_b$,N$_1$)

(4) E(Pu$_a$,[N$_1$||N$_2$])

(5) E(Pu$_b$,N$_2$)

A & B Authenticate each other

Distribution of Public Keys

Public key certificates

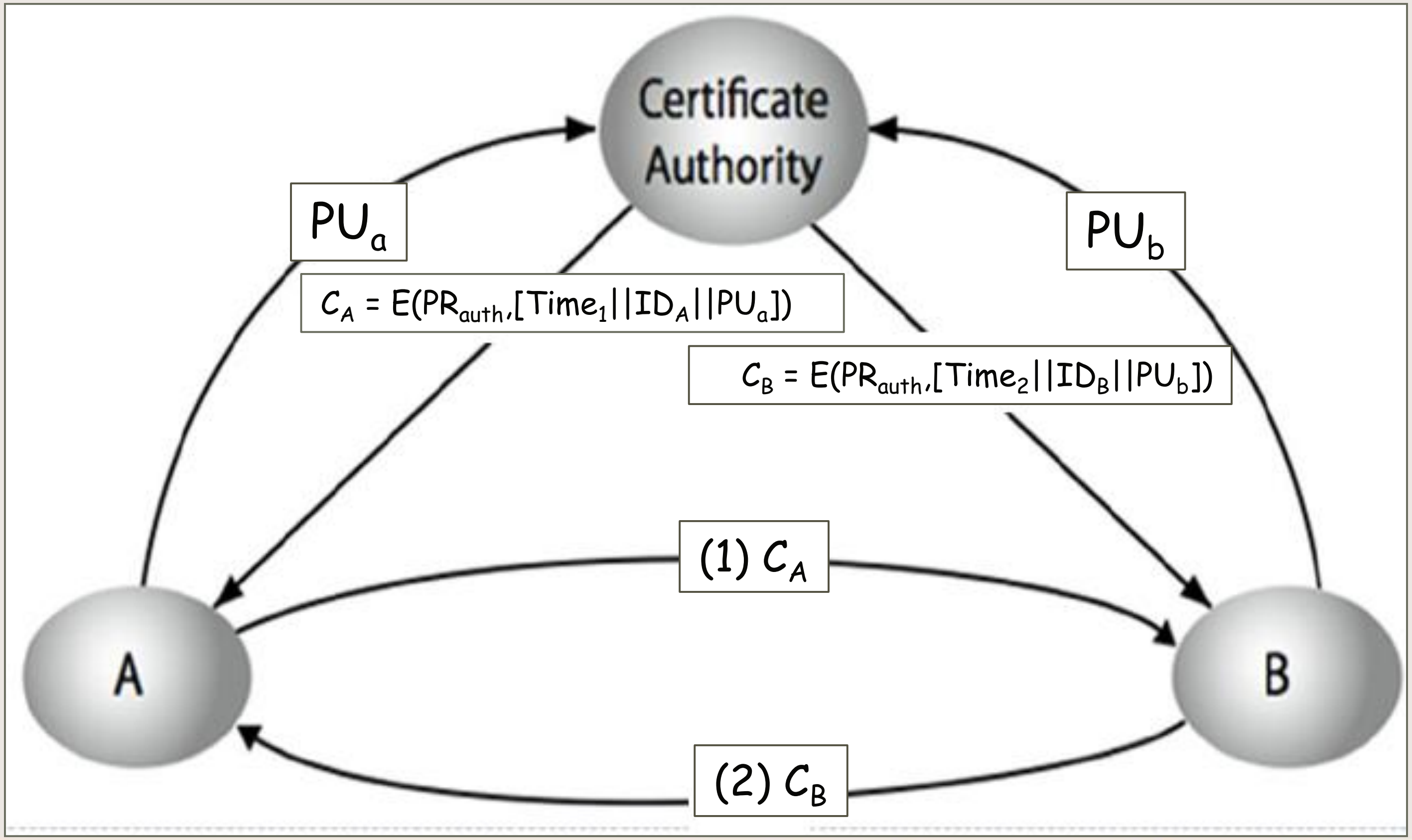
- ❑ The certificates provide a secure way of publishing public keys, so that their validity can be trusted.
- ❑ A certificate consists of a public key plus a user ID of the key owner in addition to some information about the certificate authority (CA) plus an indication of the period of validity of the certificate
- ❑ With all contents signed by a trusted Certificate Authority (CA)
- ❑ CA is a trusted third party trusted by the user community, government agency or a financial institution.
- ❑ A user can present his public key to the authority in a secure manner and obtain a certificate.
- ❑ The user can then publish the certificate . Anyone needed this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature

Distribution of Public Keys

Public key certificates

Any other participant, who reads and verifies the certificate as follows:

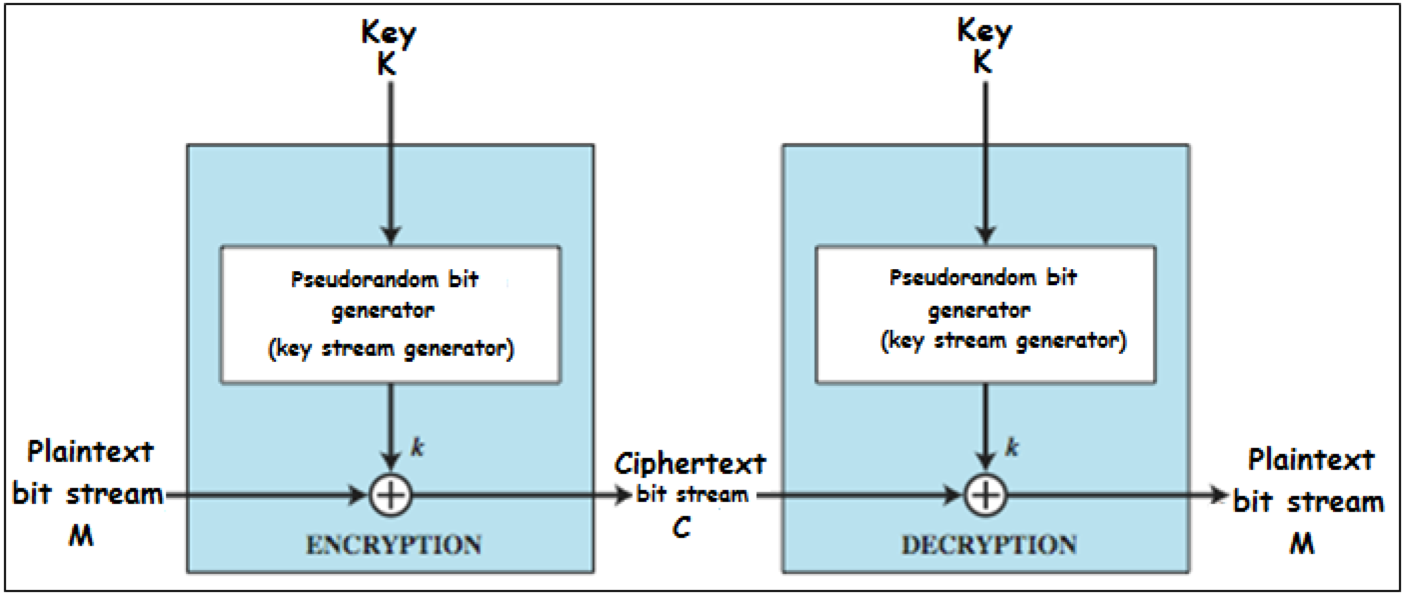
$D(PU_{auth}, CA) = D(PU_{auth}, E(PR_{auth}, [T||IDA||PU_a])) = (T||IDA||PU_a)$


Certificate Authority
PU_a
PU_b
$C_A = E(PR_{auth}, [Time_1||ID_A||PU_a])$
$C_B = E(PR_{auth}, [Time_2||ID_B||PU_b])$
(1) C_A
(2) C_B
A
B

Random Numbers

Random Numbers

❑ Stream Ciphers

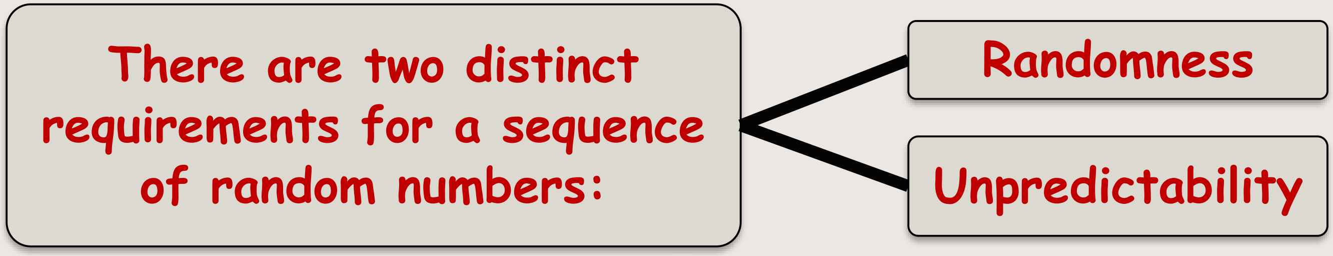

Key
K
Pseudorandom bit generator (key stream generator)
k
Plaintext bit stream M
ENCRYPTION
Ciphertext bit stream C
Key
K
Pseudorandom bit generator (key stream generator)
k
DECRYPTION
Plaintext bit stream M

A method of encrypting in which a cryptographic key (K) and encryption algorithm are applied to each binary digit (M) in a data stream, one bit at a time, result one bit cipher strem (C).

Random Numbers

A number of security algorithms and protocols based on cryptography make use of random binary numbers:

- □ Key distribution and reciprocal authentication schemes
- □ Session key generation
- □ Generation of keys for the RSA public-key encryption algorithm
- □ Generation of a bit stream for symmetric stream encryption



There are two distinct requirements for a sequence of random numbers:

Randomness

Unpredictability

Randomness

❑ Two criteria are used to validate that a sequence of numbers is random:

I- Uniform distribution

The frequency of occurrence of ones and zeros should be approximately equal

II- Independence

No one subsequence in the sequence can be inferred from the others

Unpredictability

- ❑ The requirement is not just that the sequence of numbers be statistically random, but that the successive members of the sequence are unpredictable
- ❑ With "true" random sequences each number is statistically independent of other numbers in the sequence and therefore unpredictable.
- ❑ It is more common to implement algorithms that generate sequences of numbers that appear to be random
- ❑ Care must be taken that an opponent not be able to predict future elements of the sequence on the basis of earlier elements

Pseudorandom Number Generator (PRNG)

- ❑ Often use algorithmic technique to create pseudorandom numbers:
  - ❖ Which satisfy statistical randomness tests
  - ❖ But likely to be predictable
- ❑ PRNG takes as input a fixed value, called the seed, and produces a sequence of output bits using a deterministic algorithm.
- ❑ The important thing to note is that the output bit stream is determined solely by the input value, so that an attacker who knows the algorithm and the seed can reproduce the entire bit stream.

True Random Number Generator (TRNG)

- ❑ True random number generators (TRNG) use a nondeterministic source
  - ❖ e.g. radiation, gas discharge, leaky capacitors
  - ❖ increasingly provided on modern processors

Information Hiding Alternatives

Concealment (or Null) Cipher

- ❑ Letters are hidden by a algorithm.
- ❑ For example: every third word in a sentence:
  - ❖ “The old red rooster hit head first bypassing rules.”
  - ❖ The secret message is “red head rules”.

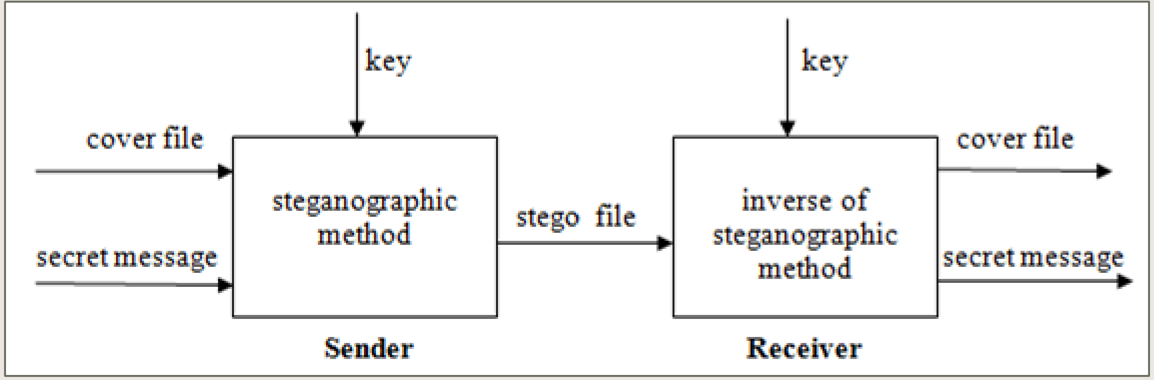
Hiding messages in Media

Steganography

- ❑ Defined by : “the practice of concealing messages or information within other non secret text or data”
- ❑ Serves to hide secret messages in other messages, such that the Secret’s very existence is concealed.
- ❑ Invisible and hard to detect without tools. You cannot even claim there is a message.
- ❑ Can hide information in a variety of formats:
  - ❖ High Quality graphic files.
  - ❖ MP3 and other audio files.
  - ❖ Text files

Hiding messages in Media

Steganography components



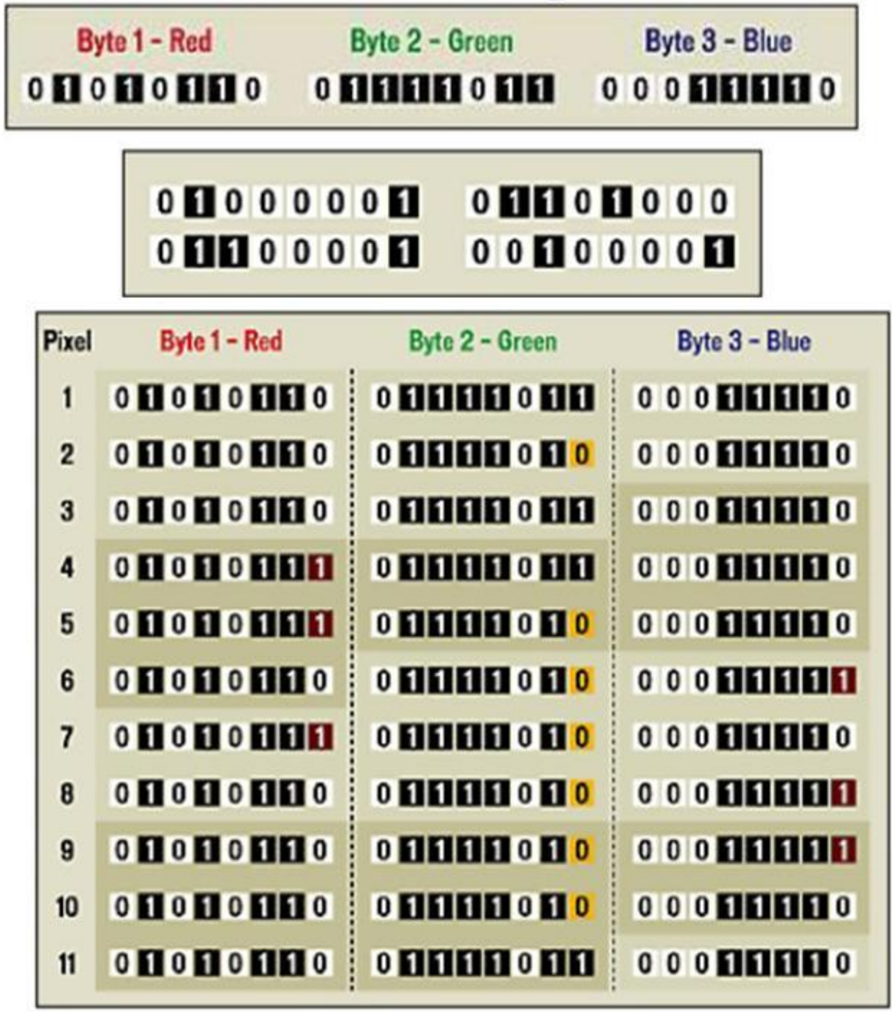
Secret message: The secret message or information to hide.
Cover file: The data or medium which concealed the secret message.
Stego file: A modified version of cover that contains the secret message.
key: Additional secret data that is needed for the embedding and extracting processes and must be known to both, the sender and the recipient
Steganographic method: A steganographic function that takes cover, secret message and key as parameters and produces stego as output.
Inverse of steganographic method: A steganographic function that has stego and key as parameters and produces secret message as output.

Hiding messages in Media

Steganography

- ❑ Combining Steganography and Cryptography one can achieve better security.
- ❑ There are distributed steganography methods, including methodologies that distribute the payload through multiple carrier files in diverse locations to make detection more difficult.

Steganography Explained



Byte 1 - Red	Byte 2 - Green	Byte 3 - Blue
0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0

0 1 0 0 0 0 0 1	0 1 1 0 1 0 0 0
0 1 1 0 0 0 0 1	0 0 1 0 0 0 0 1

Pixel	Byte 1 - Red	Byte 2 - Green	Byte 3 - Blue
1	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0
2	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 0
3	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0
4	0 1 0 1 0 1 1 1	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0
5	0 1 0 1 0 1 1 1	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 0
6	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 1
7	0 1 0 1 0 1 1 1	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 0
8	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 1
9	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 1
10	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 0
11	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0

1 pixel = 24 bits or 3 bytes
256 value for each color
256*256*256=16777216 or 16.7 Million colors

← The 4 characters of Aha! – in ASCII binary

← We now hide one bit at the end of each of the color code bytes of 11 different pixels

← In the diagram, each maroon or gold box represents a bit had to be changes to include the hidden message.

← Notice that only 15 of 264 bits (less than 6% had to be changed and only 8 of 11 pixels were altered.

← Changes will not be visible to a human eye .

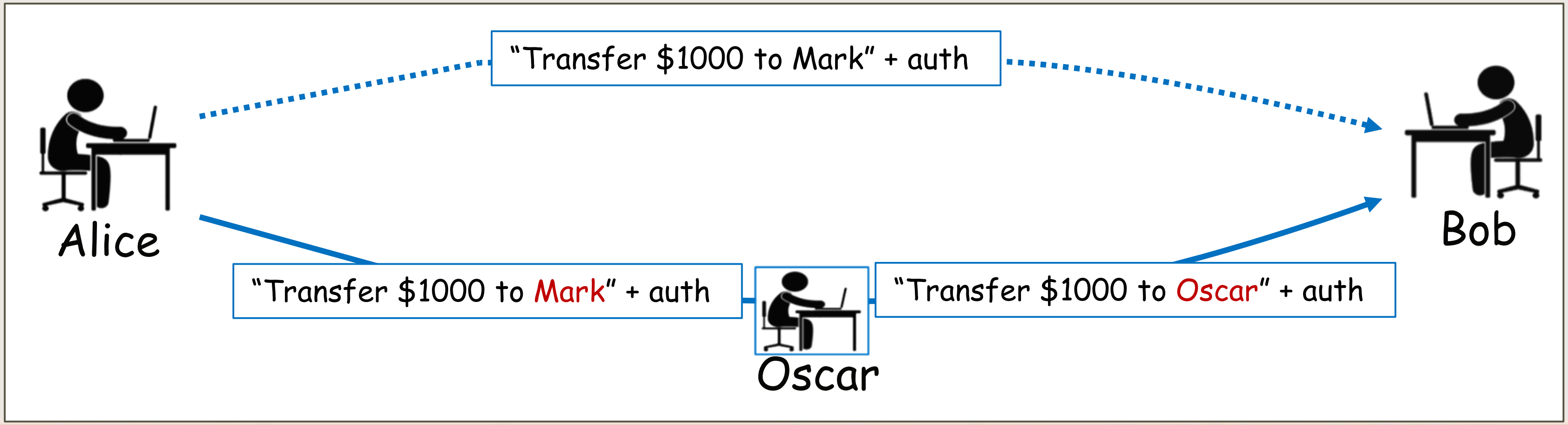
Encryption & Steganography

Encryption	Steganography
The study of hiding information	Composing hidden messages so that only the sender and the receiver know that the message even exists
The existence of the encrypted message is visible to the world	Only the sender and the receiver know the existence of the message. Due to this, Steganography removes the unwanted attention coming to the hidden message
Protect the content of a message	Hide both the message as well as the content.

Questions

- Consider we use security service provided by digital signatures (DS).
- We assume that Oscar is able to observe all messages sent from Alice to Bob and vice versa.
- Oscar has no knowledge of any keys but the public one in case of DS.
- The message signature "auth(x)" is computed with a DS algorithm.
- **State whether and how DS protect against each attack?**

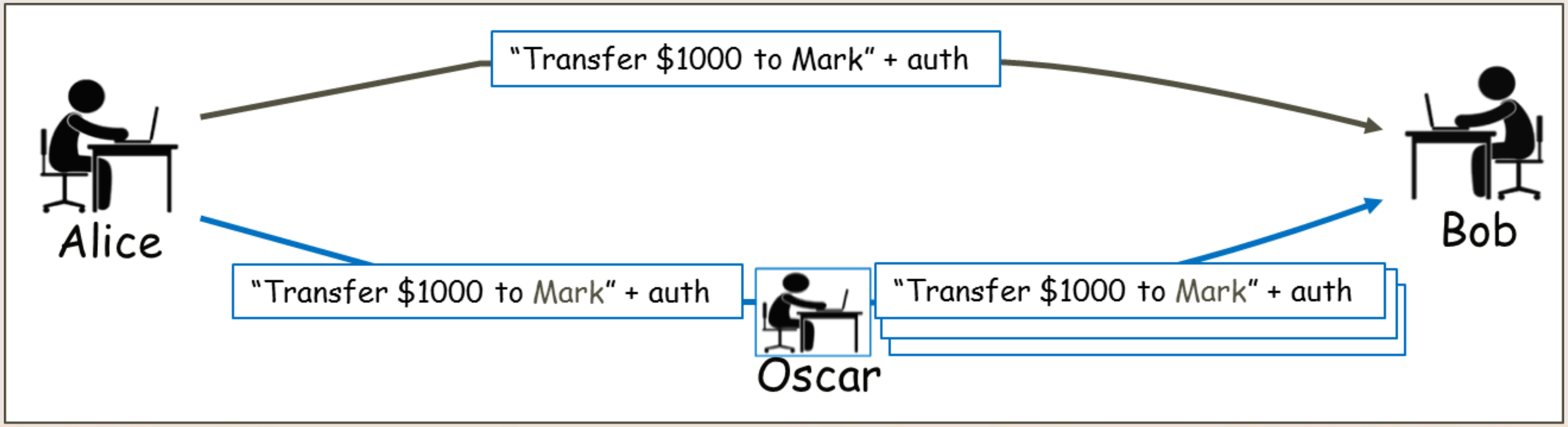
(Message integrity) Alice sends a message x "Transfer $1000 to Mark" in the clear and also sends auth(x) to Bob. Oscar intercepts the message and replaces "Mark" with "Oscar". Will Bob detect this?


"Transfer $1000 to Mark" + auth
Alice
Bob
"Transfer $1000 to Mark" + auth
Oscar
"Transfer $1000 to Oscar" + auth

Questions

- Consider we use security service provided by digital signatures (DS).
- We assume that Oscar is able to observe all messages sent from Alice to Bob and vice versa.
- Oscar has no knowledge of any keys but the public one in case of DS.
- The message signature "auth(x)" is computed with a DS algorithm.
- **State whether and how DS protect against each attack?**

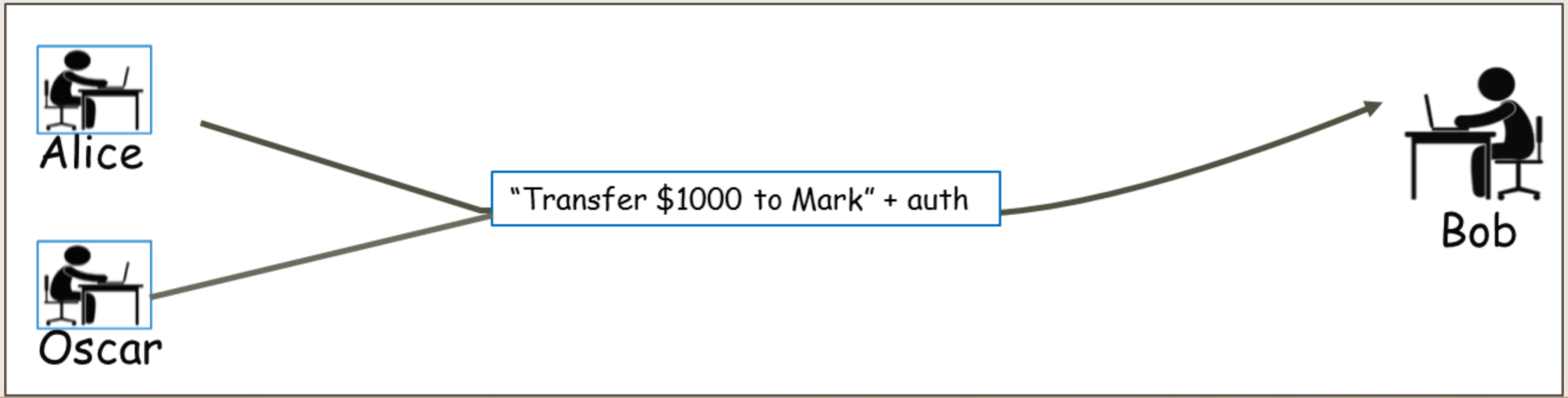
(Replay) Alice sends a message x "Transfer $1000 to Mark" in the clear auth(x) to Bob. Oscar observes the message and signature and sends them 100 times to Bob. Will Bob detect this?


"Transfer $1000 to Mark" + auth
Alice
Bob
"Transfer $1000 to Mark" + auth
Oscar
"Transfer $1000 to Mark" + auth

Questions

- Consider we use security service provided by digital signatures (DS).
- We assume that Oscar is able to observe all messages sent from Alice to Bob and vice versa.
- Oscar has no knowledge of any keys but the public one in case of DS.
- The message signature “auth(x)” is computed with a DS algorithm.
- **State whether and how DS protect against each attack?**

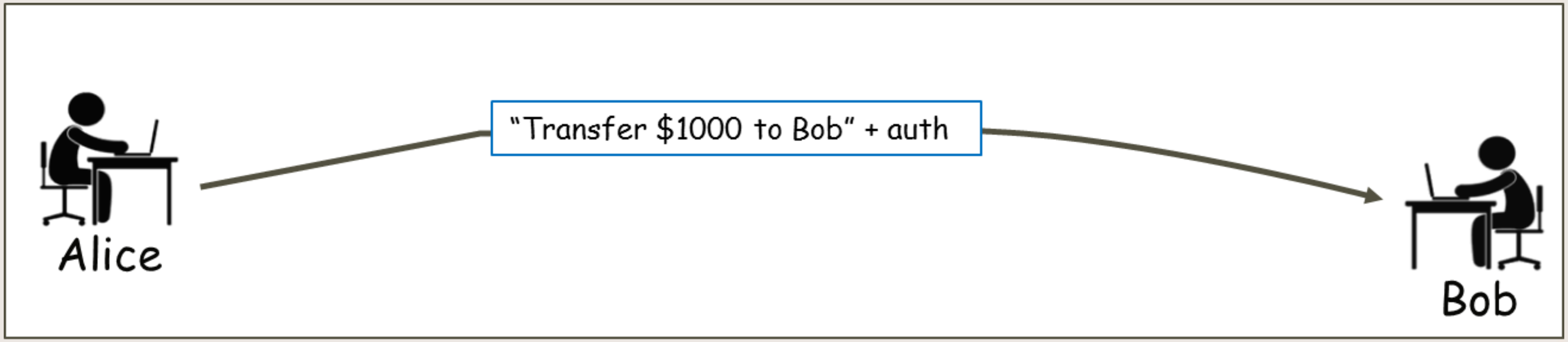
(Sender Authentication with cheating third party) Oscar claims that he sent some message x with a valid auth(x) to Bob but Alice claims the same. Can Bob clear the question in either case?



Questions

- Consider we use security service provided by digital signatures (DS).
- We assume that Oscar is able to observe all messages sent from Alice to Bob and vice versa.
- Oscar has no knowledge of any keys but the public one in case of DS.
- The message signature "auth(x)" is computed with a DS algorithm.
- **State whether and how DS protect against each attack?**

(Authentication with Bob cheating) Bob claims that he received a message x with a valid signature auth(x) from Alice (e.g., "Transfer $1000 from Alice to Bob") but Alice claims she has never sent it. Can Alice clear this question in either case?



Questions

Alice and Bob have each generated a public and private key pair. However, they do not know each others' keys yet. Now they are trying to exchange a message "M" over a network.

(a) What is the procedure to exchange the message confidentially, if only passive attacks need to be considered?

(b) What can be done to mitigate the risk if active attacks are possible?