DEPT & SEM : CSE & ISEM

SUBJECT NAME: INFORMATION SECURITY

COURSE CODE :

CS4101pC

:UNIT : I

PREPARED BY Anusha K

OUTLINE

- **Computer Security concepts**
- **The OSI Security Architecture**
- **Security attacks**
- **Security services and**
- **Security mechanisms**
- **A model for Network Security**

INTRODUCTION

A Definition of Computer Security

The NIST (National Institute of Standards & Technology) *Computer Security Handbook [NIST95] defines the term computer security* as follows:

COMPUTER SECURITY

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/ data, and telecommunications).

Figure 1.1: Definition

1.1 COMPUTER SECURITY CONCEPTS

This definition introduces three key objectives that are at the heart of computer security:

1. Confidentiality
2. Integrity

1) Confidentiality: This term covers two related concepts:

Data1 confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorized individuals.

Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

2) Integrity: This term covers two related concepts:

Data integrity: Assures that information and programs are changed only in a specified and authorized manner.

System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

3)Availability: Assures that systems work promptly and service is not denied to authorized users.

1.1 COMPUTER SECURITY CONCEPTS

- These three concepts form what is often referred to as the **CIA triad** (Figure shown below)

- The three concepts embody the fundamental security objectives for both data and for information and computing services.

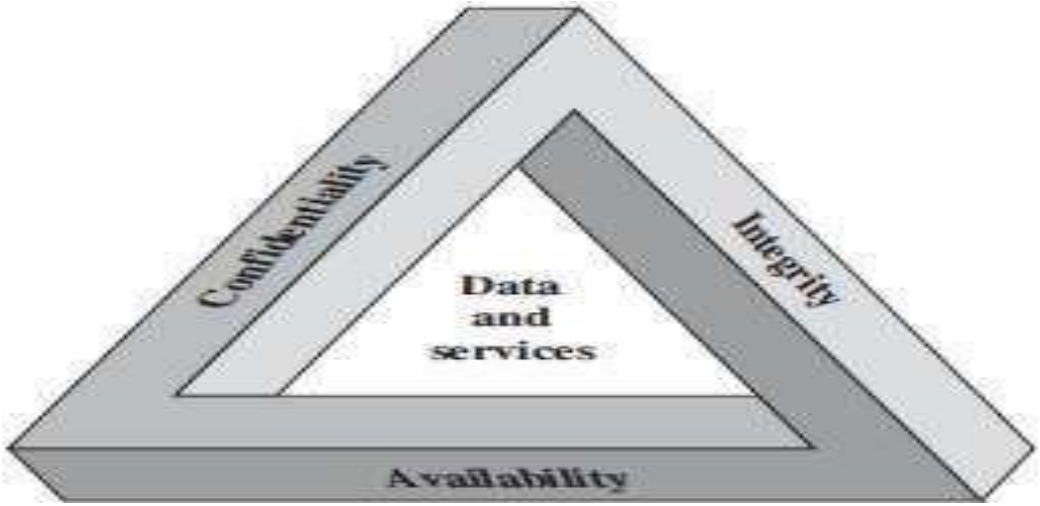


Figure 1.2: The System Security Traid

1.1 COMPUTER SECURITY CONCEPTS

- **Confidentiality:** Preserving authorized restrictions on information access and disclosure.
- **Integrity:** Guarding against improper information modification or destruction.
- **Availability:** Ensuring timely and reliable access to and use of information.
- **Authenticity:** The property of being genuine and being able to be verified and trusted.
- **Accountability:** The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity

- A, or availability **three levels of impact** on organizations or individuals.

a) **Low:** The loss could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals. A limited adverse effect means

(i) cause a degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced;

(ii) result in minor damage to organizational assets;

(iii) result in minor financial loss; or

(iv) result in minor harm to individuals.

b) <u>Moderate:</u> The loss could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.

A serious adverse effect means that, for example, the loss might:

(i) cause a significant degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is significantly reduced;

(ii) result in significant damage to organizational assets;

(iii) result in significant financial loss; or

(iv) result in significant harm to individuals that does not involve loss of life or serious, life-threatening injuries.

c) High: The loss could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.

A severe or catastrophic adverse effect means that, for example, the loss might

(i) cause a severe degradation in or loss of mission capability to an extent and duration that the organization is not able to perform one or more of its primary functions;

(ii) result in major damage to organizational assets;

(iii) result in major financial loss; or

(iv) result in severe or catastrophic harm to individuals involving loss of life or serious, life-threatening injuries.

The Challenges of Computer Security:

1) Security is not as simple.

2) In developing a particular security mechanism or algorithm, one must always consider potential attacks on those security features.

3) The procedures used to provide particular services are often counterintuitive.

4) It is necessary to decide where various security mechanisms and where to use them.

5) Security mechanisms typically involve more than a particular algorithm or protocol. Consider problems with creation, distribution, and protection of that secret information.

6) Computer and network security is essentially a battle of wits between a perpetrator who tries to find holes and the designer or administrator who tries to close them.

7) There is a natural tendency on the part of users and system managers to perceive little benefit from security investment until a security failure occurs.

8) Security requires regular, even constant, monitoring, and this is difficult in today's short-term, overloaded environment.

9) Many users and even security administrators view strong security as an impediment to efficient and user-friendly operation of an information system or use of information.

1.2 THE OSI SECURITY ARCHITECTURE

- The OSI security architecture is useful to managers as a way of organizing the task of providing security.
- The OSI security architecture focuses on security attacks, mechanisms, and services.

These can be defined briefly as:

1) Security attack: Any action that compromises the security of information owned by an organization.

2) Security mechanism: A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.

3) Security service: A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization.

Threat

A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

Attack

An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

Figure 1.3: Definition of Threat & Attack

1.3 SECURITY ATTACKS

- X.800 and RFC 2828 defines attacks in terms of *passive attacks and active attacks.*
- *A* ***Passive attack*** *attempts to learn or* make use of information from the system but does not affect system resources.

There are two types of passive attacks:

a) The **release of message contents** is easily understood. A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information.

b) **Traffic analysis** is a way of masking the contents of messages or other information traffic. An opponent observe the pattern of messages, determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged.

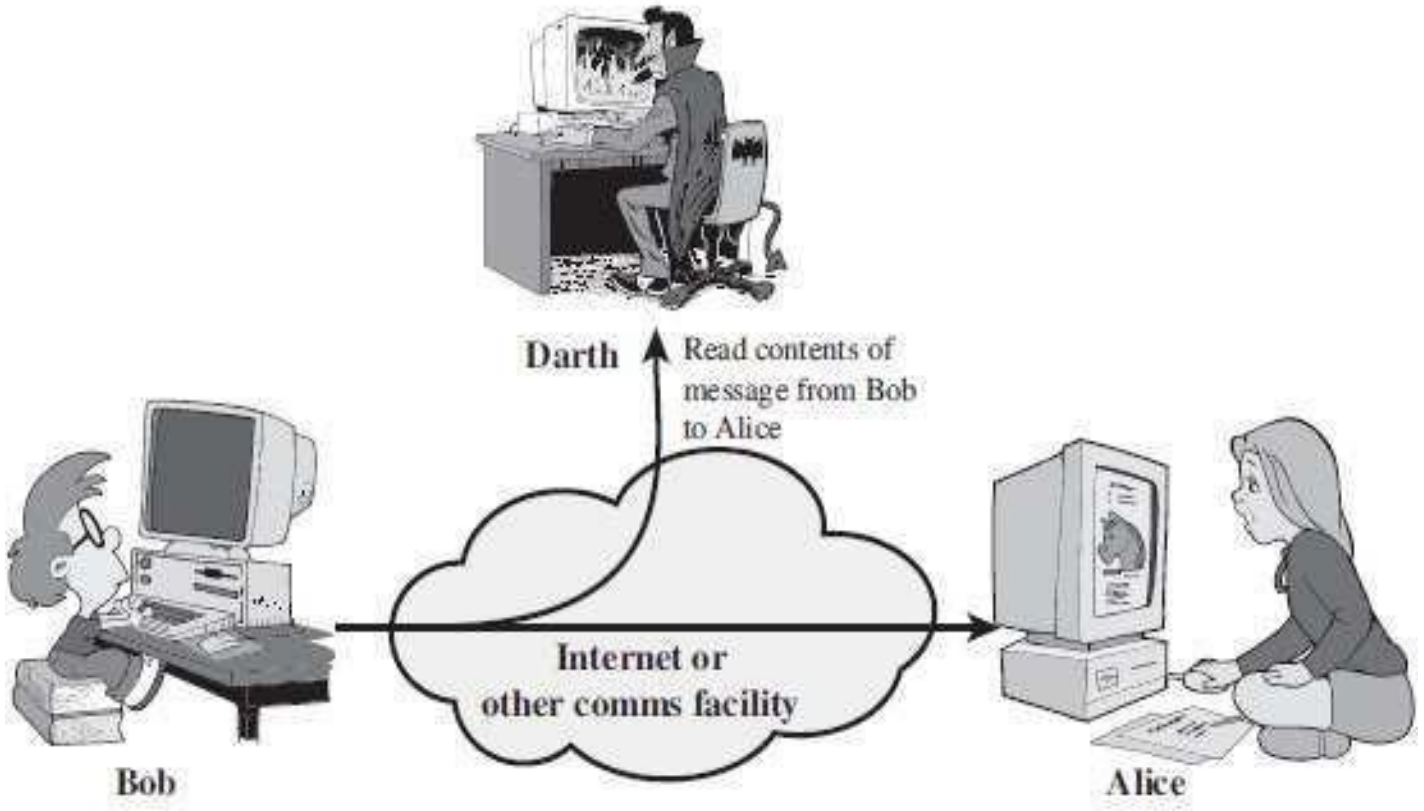


Figure 1.4:Release of Message content

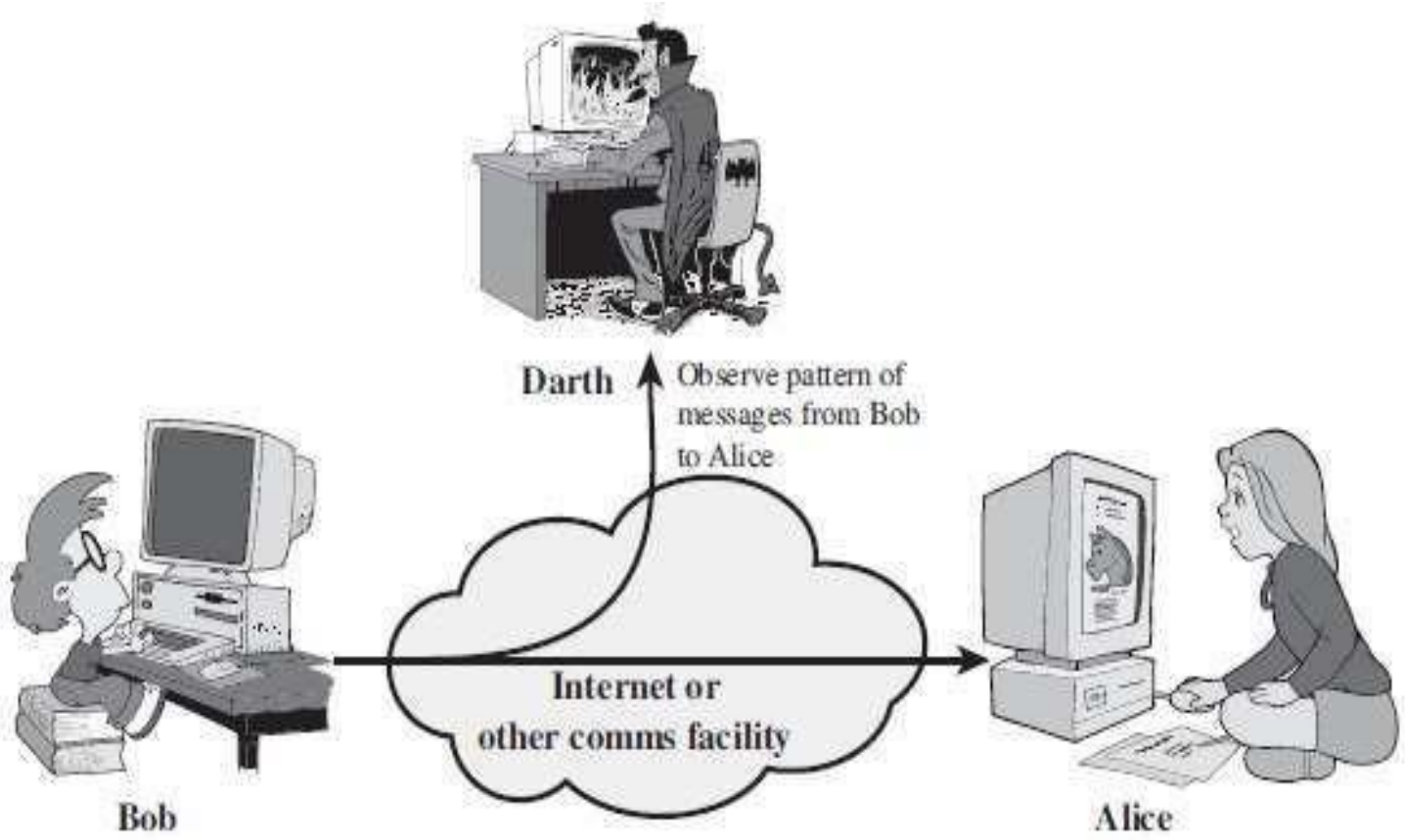


Figure 1.5 :Traffic Analysis

- An ***Active attack*** attempts to alter system resources or affect their operation. There are four types of active attacks:

a) A **masquerade** takes place when one entity pretends to be a different entity**.**

b) **Replay** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

c) **Modification of messages** simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized Effect

For example, "Allow John Smith to read confidential file *accounts" is modified to mean "Allow Fred Brown to read* confidential file *accounts."*

d) The **denial of service** prevents or inhibits the normal use or management of communications facilities. Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

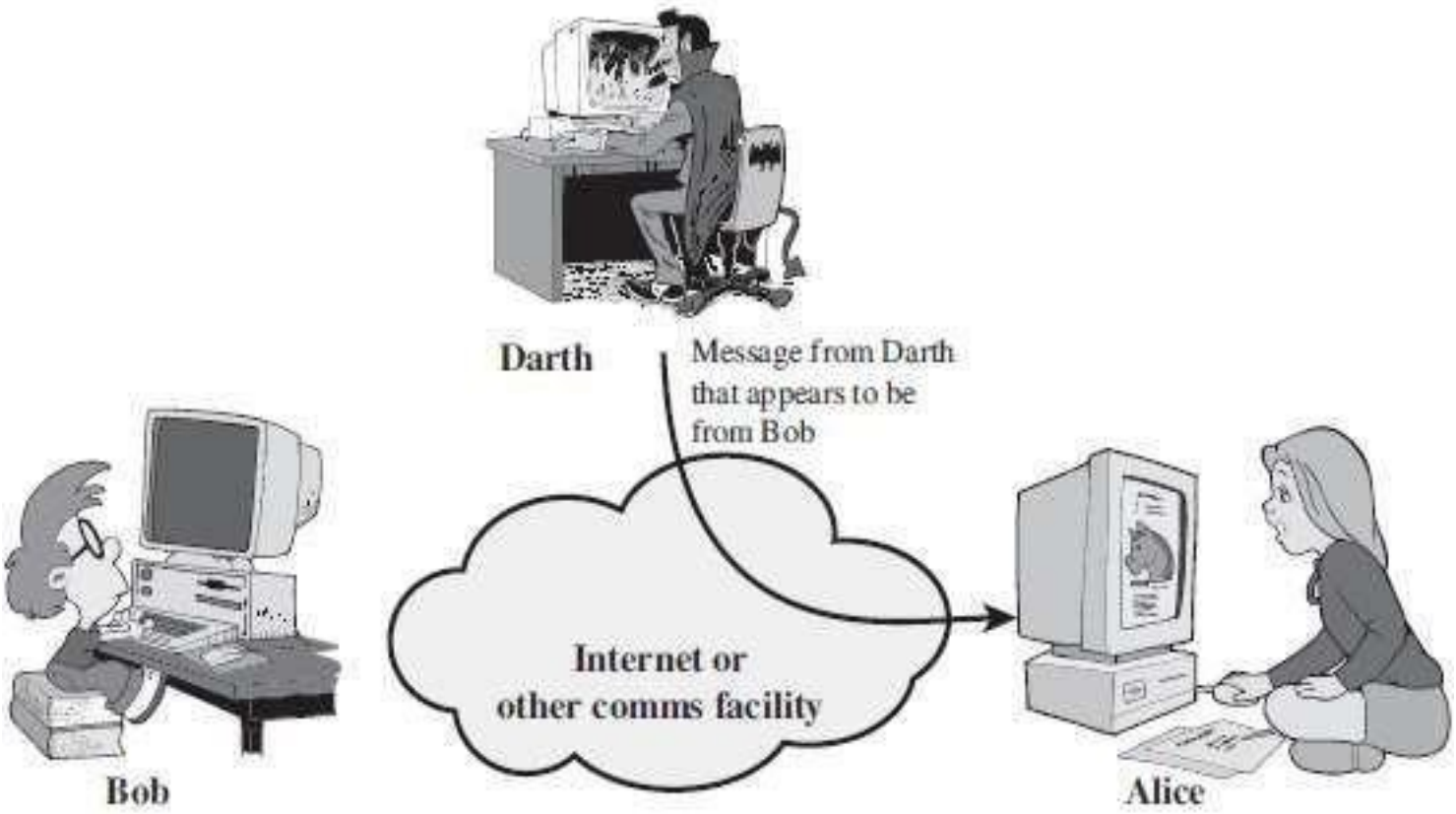


Figure 1.6 : Masquerade

SECURITY

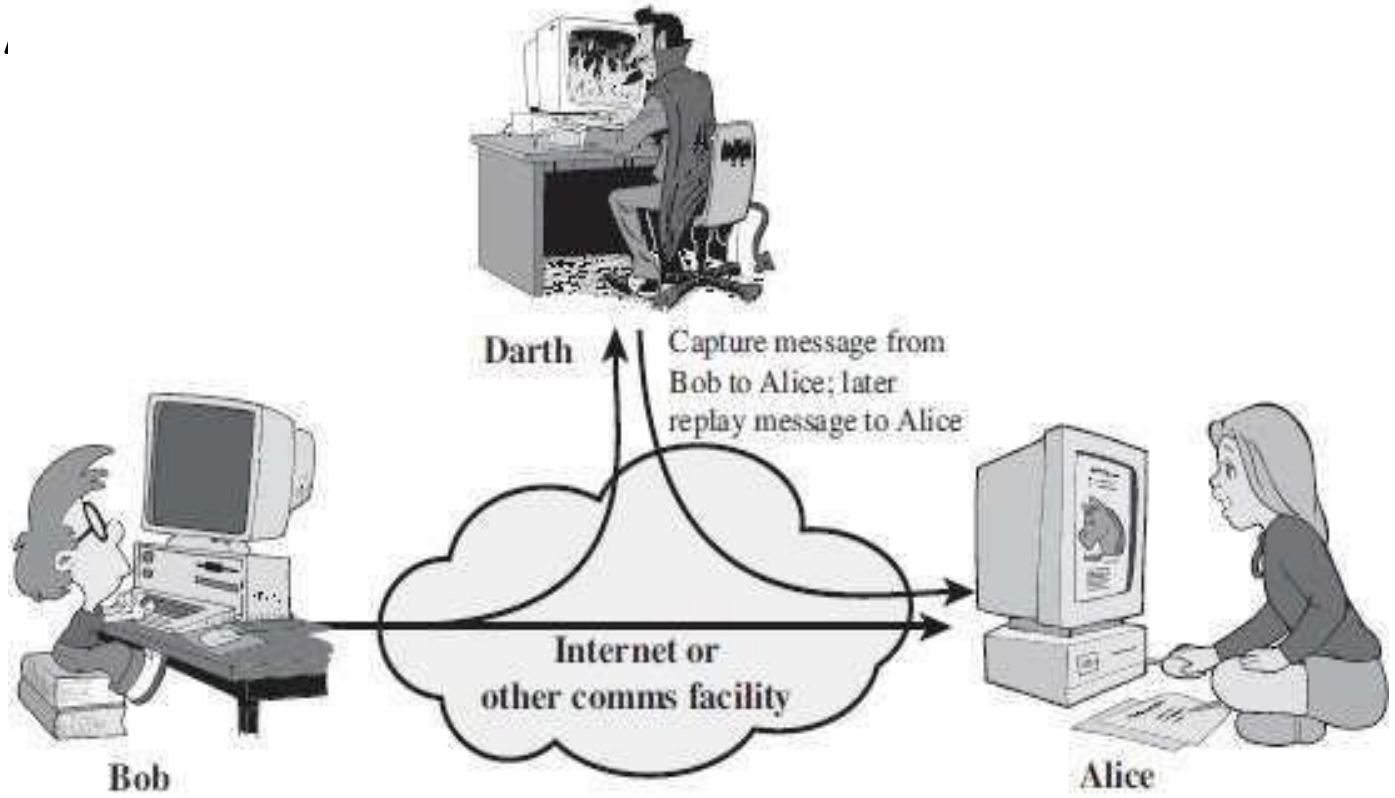


Figure 1.7 :Replay

SECURITY ATTACKS

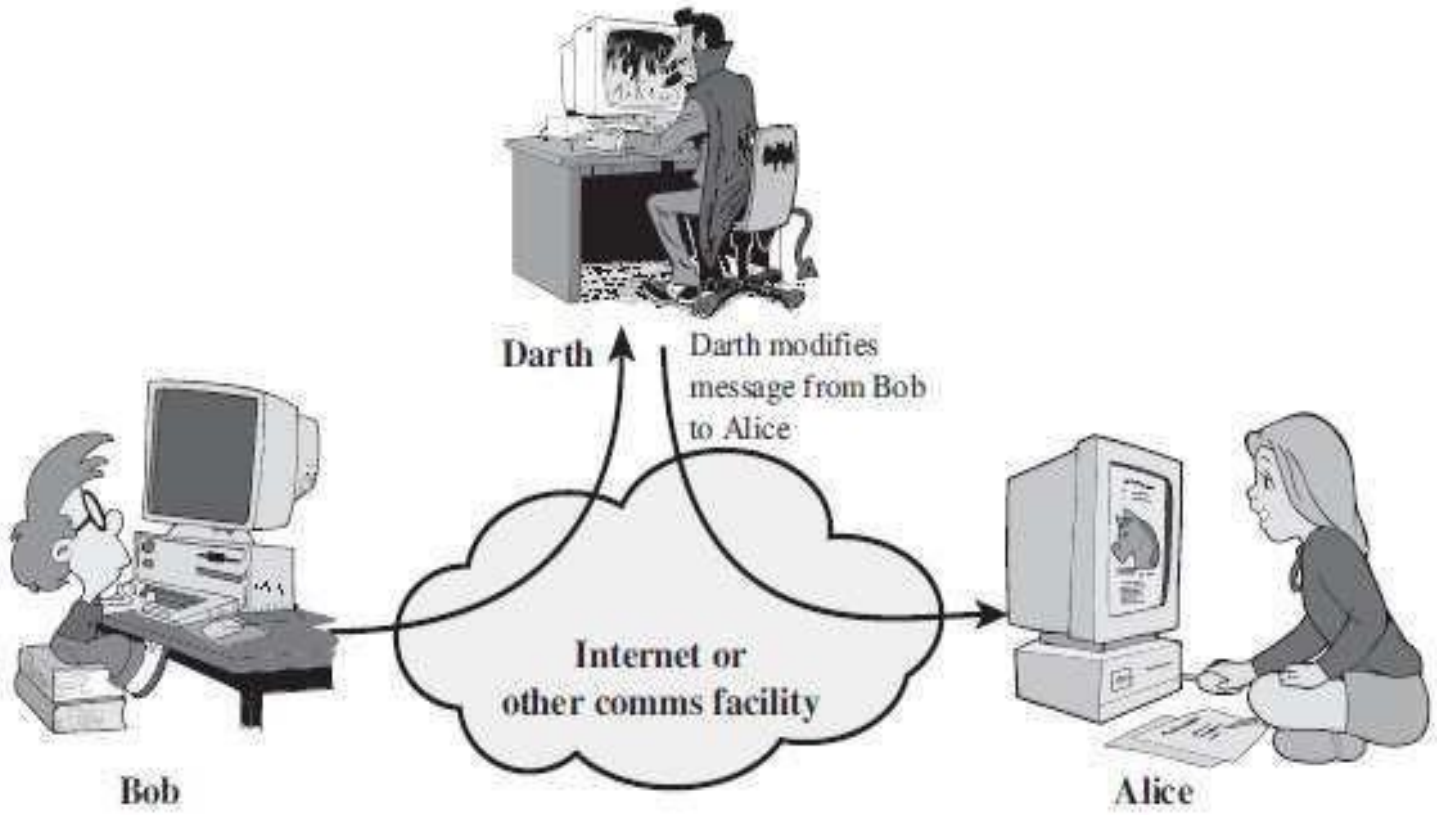


Figure 1.8 :Modification of messages

SECURITY

A

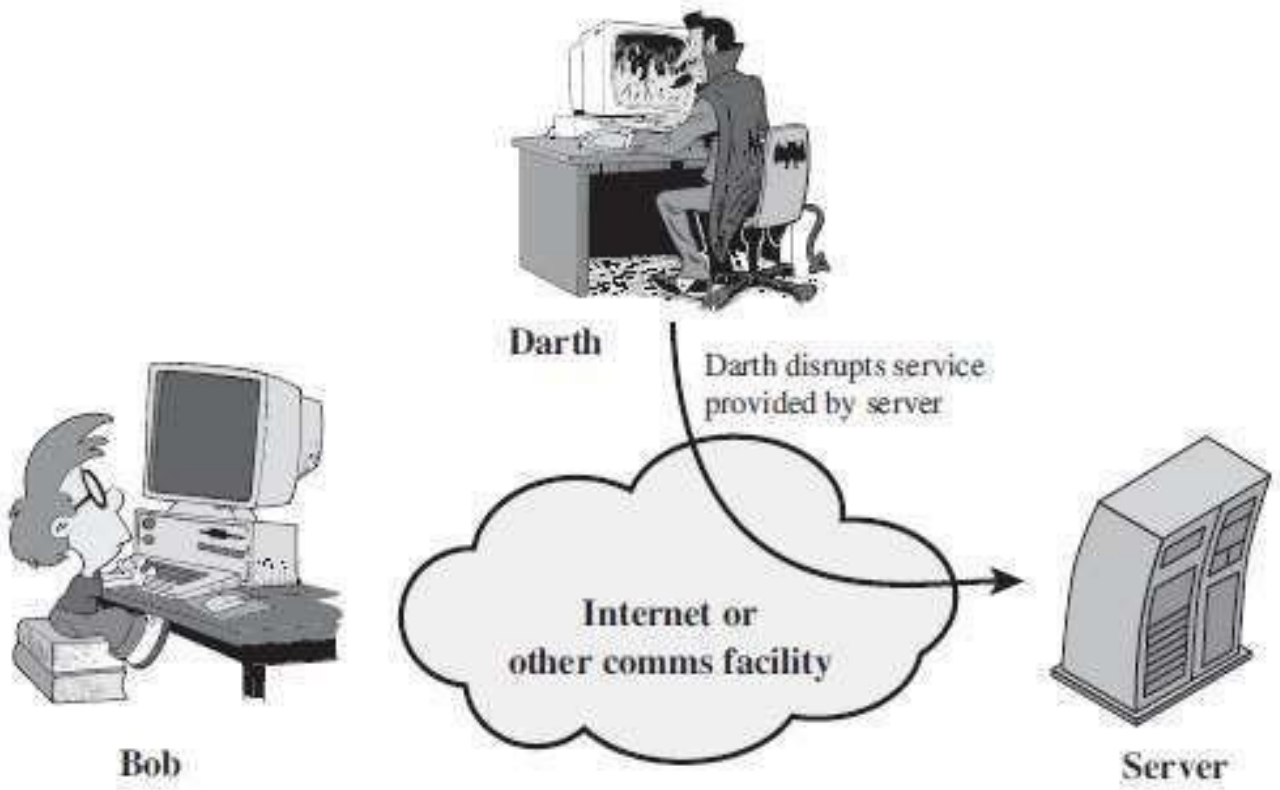


Figure 1.9: Denial of service

SECURITY SERVICES

- Security services are processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies and are implemented by security mechanisms.

- X.800 divides these services into **five** categories and **fourteen** specific services:

    1) Authentication.

    2) Access Control.

    3) Data Confidentiality.

    4) Data Integrity.

    5) Non repudiation.

SECURITY SERVICES

1) Authentication:

- It is concerned with assuring that a communication is authentic.
- First, at the time of connection initiation, the service assures that the two entities are authentic. Second, the service must assure that the connection is not interfered by unauthorized.

Two specific authentication services are defined in X.800:

a) Peer entity authentication: Provides for the corroboration of the identity of a peer entity in an association. Two entities are considered peers if they implement to same protocol in different systems.

b) Data origin authentication: Provides for the corroboration of the source of a data unit. It does not provide protection against the duplication or modification of data units.

2)Access Control: It is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

3)Data Confidentiality: It is the protection of transmitted data from passive attacks and traffic flow from analysis. Four Data Confidentiality services are:

a) Connection Confidentiality: The protection of all user data on a connection.

b) Connectionless Confidentiality: The protection of all user data in a single data block

c) Selective-Field Confidentiality: The confidentiality of selected fields within the user data on a connection or in a single data block.

d) Traffic-Flow Confidentiality: The protection of the information that might be derived from observation of traffic flows.

4) <u>Data Integrity:</u> The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).

Five types of Data Integrity services are:

a) Connection Integrity with Recovery: Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.

b) Connection Integrity without Recovery: As above, but provides only detection without recovery.

c) Selective-Field Connection Integrity: Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.

d) Connectionless Integrity: Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.

e) Selective-Field Connectionless Integrity: Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination

of whether the selected fields have been modified.

5)<u>Non repudiation:</u> Provides protection against denial by one of the entities involved in a communication of having

participated in all or part of the communication.

a) Non repudiation, Origin: Proof that the message was sent by the specified party.

b) Non repudiation, Destination: Proof that the message was received by the specified party

SECURITY MECHANISMS

- Security mechanisms are divided into those that are implemented in a specific protocol layer, such as TCP or an application-layer protocol, and those that are not specific to any particular protocol layer or security service.

- X.800 defines two types of Security mechanisms:

1) **Specific Security Mechanisms:** May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

2) **Pervasive Security Mechanisms:** Mechanisms that are not specific to any particular OSI security service or protocol layer.

SECURITY MECHANISMS

1) Specific Security Mechanisms:

a) Encipherment: The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.

b) Digital Signature: Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).

c) Access Control: A variety of mechanisms that enforce access rights to resources.

d) Data Integrity: A variety of mechanisms used to assure the integrity of a data unit or stream of data units.

e) Authentication Exchange: A mechanism intended to ensure the identity of an entity by means of information exchange.

1.5 SECURITY MECHANISMS

f) Traffic Padding: The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.

g) Routing Control: Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.

h) Notarization: The use of a trusted third party to assure certain properties of a data exchange.

2) Pervasive Security Mechanisms:

a) Trusted Functionality: That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).

1.5 SECURITY MECHANISMS

b) Security Label: The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

c) Event Detection: Detection of security-relevant events.

d) Security Audit Trail: Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.

e) Security Recovery: Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.

1.5 SECURITY MECHANISMS

	Mechanism							
Service	Encipherment	Digital Signature	Access Control	Data Integrity	Authentication Exchange	Traffic Padding	Routing Control	Notarization
Peer Entity Authentication	Y	Y			Y			
Data Origin Authentication	Y	Y						
Access Control			Y					
Confidentiality	Y						Y	
Traffic Flow Confidentiality	Y					Y	Y	
Data Integrity	Y	Y		Y				
Nonrepudiation		Y		Y				Y
Availability				Y	Y			

Table 1.1: Relationship between Security and Mechanisms

NETWORK SECURITY

- A message is to be transferred from one party to another across some sort of Internet service.
- The two parties, who are the *principals in this transaction, must* cooperate

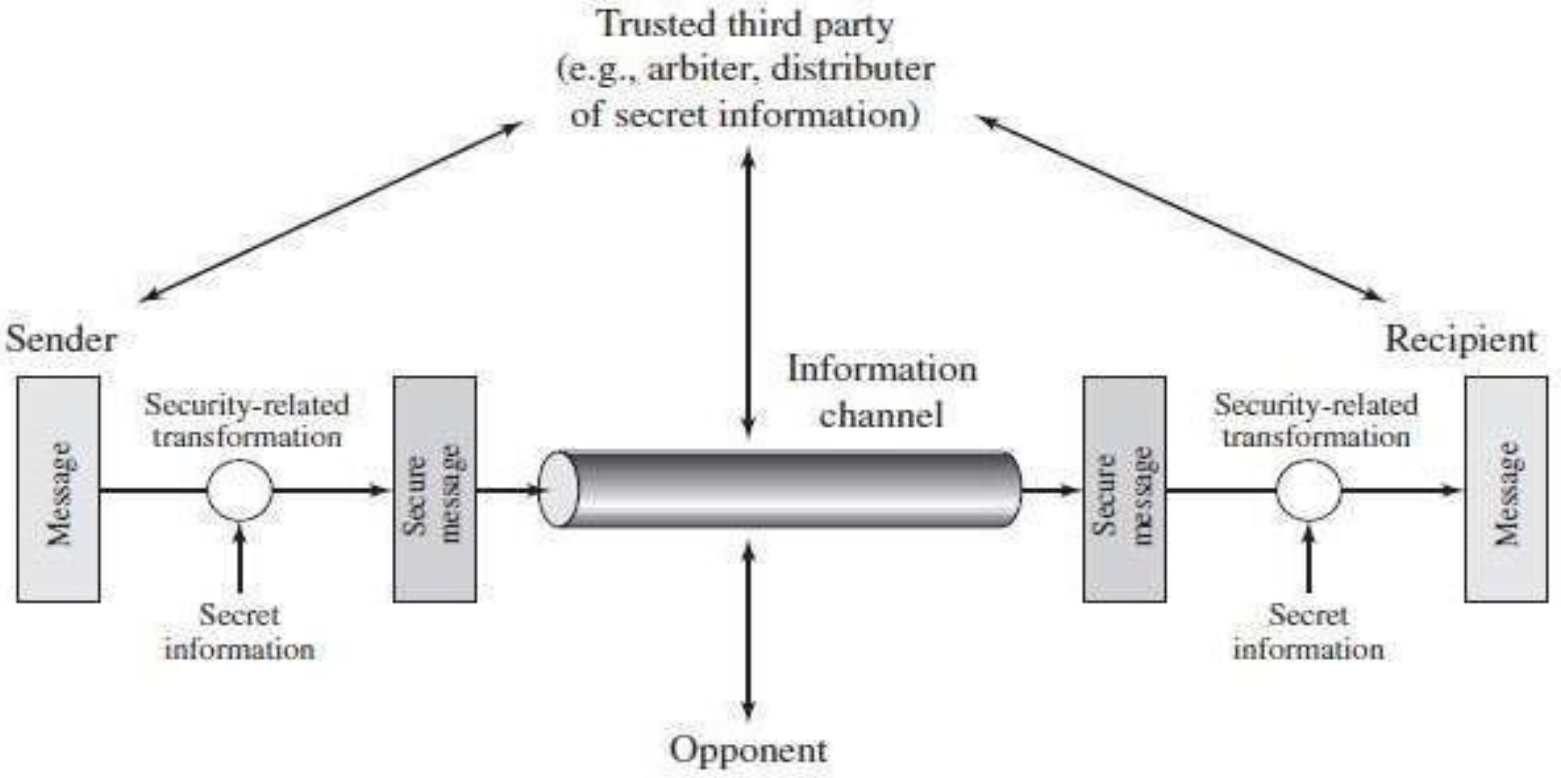


Figure 1.10 :Model for Network Security

NETWORK SECURITY

All the techniques for providing security have two components:

1) A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender.

2) Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception

NETWORK SECURITY

This general model shows that there are four basic tasks in designing a particular security service:

1) Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.

2) Generate the secret information to be used with the algorithm.

3) Develop methods for the distribution and sharing of the secret information.

4) Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

NETWORK SECURITY

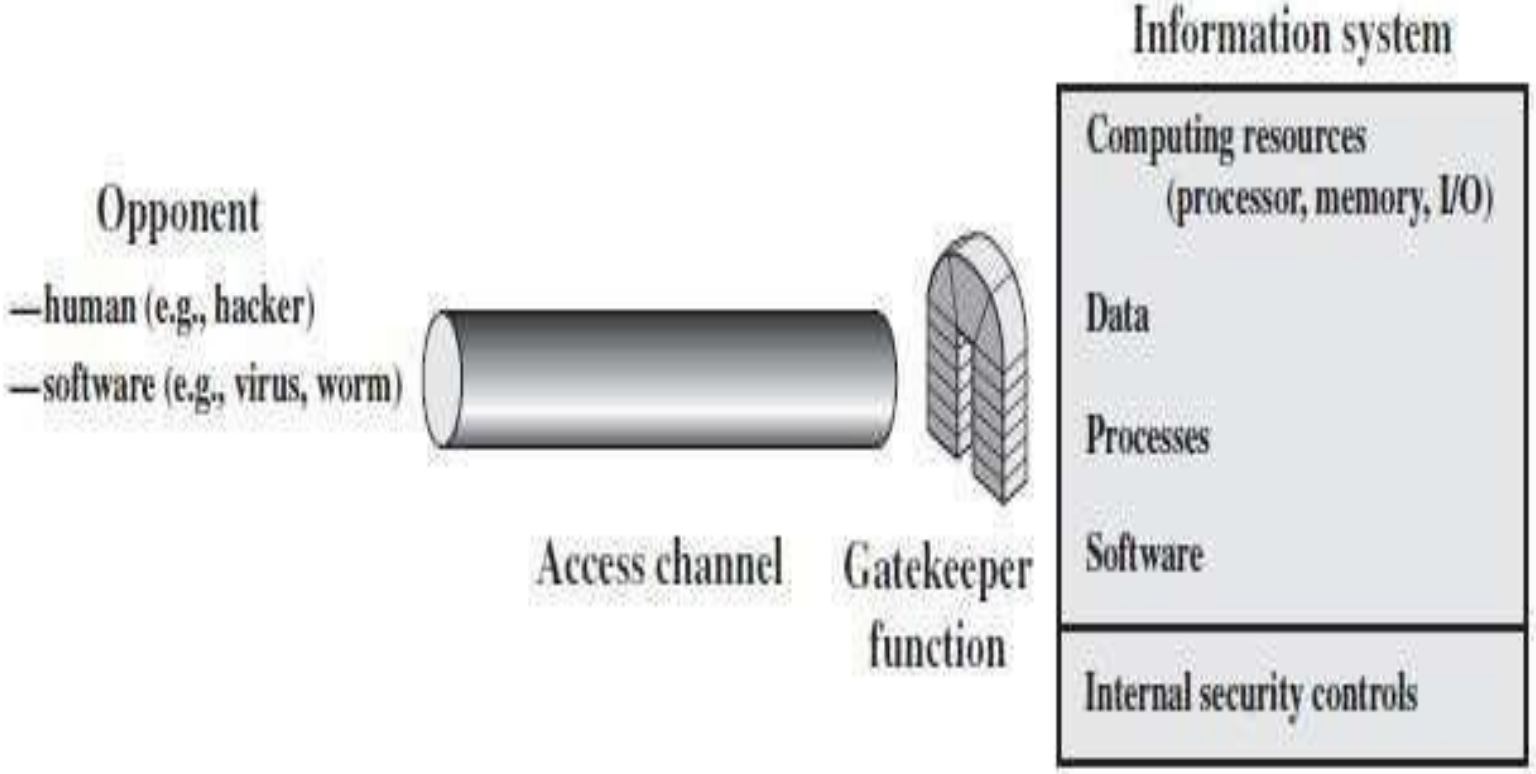


Figure 1.11 :Network Security Components

NETWORK SECURITY

- A gatekeeper function includes password-based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to detect and reject worms, viruses.
- Internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.
- Vulnerabilities affect application programs as well as utility programs, such as editors and compilers.
- Two types of Threats:
- **Information access threats**: Intercept or modify data on behalf of users who should not have access to that data.
- **Service threats**: Exploit service flaws in computers to inhibit use by legitimate users.

ON SECURITY

CHAPTER-2

CLASSICAL ENCRYPTION TECHNIQUES

CIPHER MODEL

A symmetric encryption scheme has five ingredients:

- Plaintext: original message to be encrypted.
- Cipher text: the encrypted message.
- 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 ciphertext and the secret key and produces the original plaintext.
- Secret key: A secret key is the piece of information or parameter that is used to encrypt and decrypt messages in a symmetric, or secret-key, encryption

CIPHER MODEL

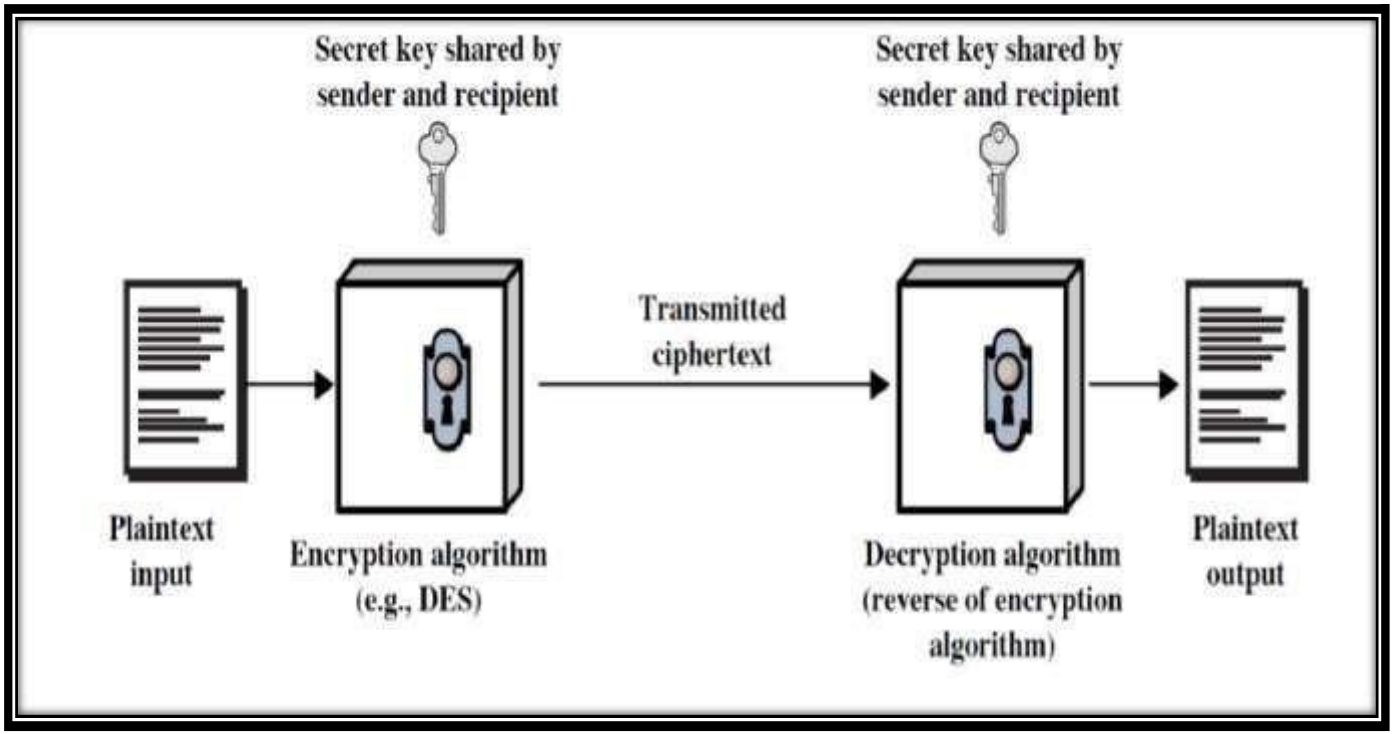


Figure 1.12 :Simplified Model of Symmetric encryption

CIPHER MODEL

There are two requirements for secure use of symmetric encryption:

- A strong encryption algorithm.
- A secret key known only to sender / receiver.

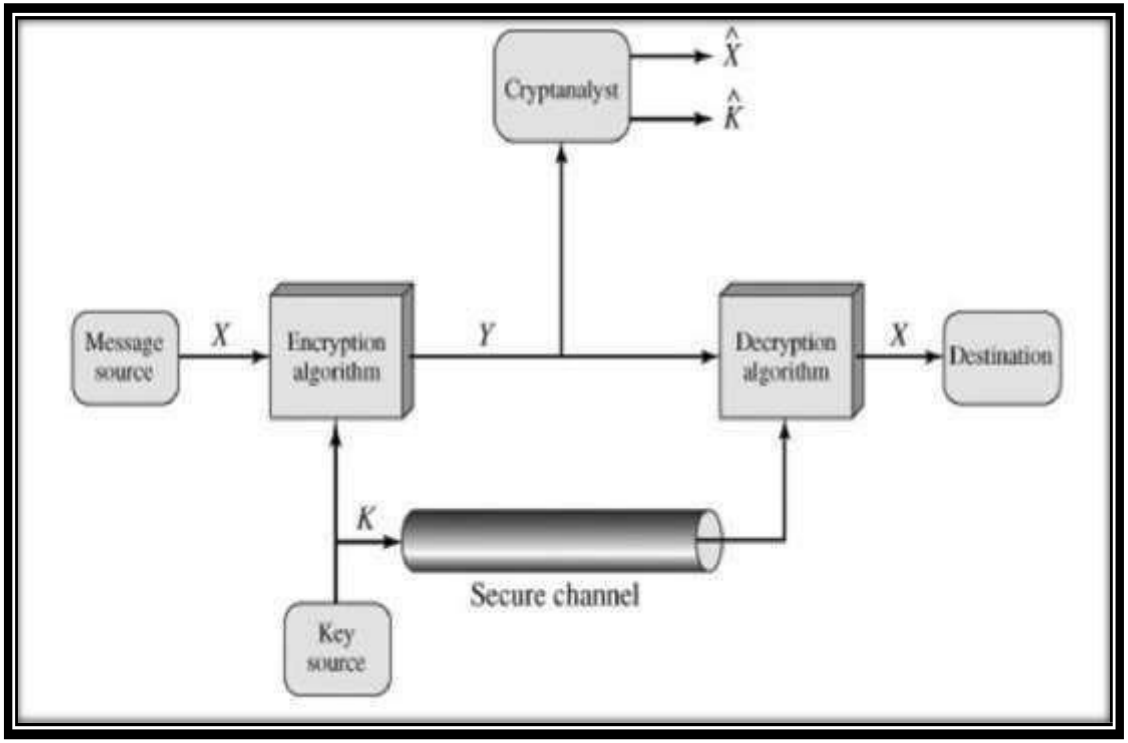


Figure 1.13 :Model of Conventional Cryptosystem

SYMMETRIC ENCRYPTION

Mathematically:

$Y = EK(X)$ or

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

$X = D(K, Y)$

- X = plaintext
- Y = ciphertext
- K = secret key
- E = encryption algorithm
- D = decryption algorithm
- Both E and D are known to public

CRYPTO GRAPHY

Cryptographic systems are characterized along three independent dimensions:

- The type of operations used for transforming plaintext to ciphertext.
- The number of keys used.
- The way in which the plaintext is processed.

CRYPTA NALYSIS

Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–cipher text pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.

Kirchhoff's principle: the adversary knows all details about a cryptosystem except the secret key.

Two general approaches:

- brute-force attack
- non-brute-force attack (cryptanalytic attack)

BRUTE-FORCE ATTACK:

- The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained. on average, half of all possible keys must be tried to achieve success.

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

Table 1.2 :Brute Force Attack

CRYPTANALYTIC ATTACKS

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

Table 1.3 : Cryptanalytic Attacks

UNCONDITIONAL SECURITY

- An encryption scheme is unconditionally secure if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available.
- Therefore, all that the users of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria:

  (1) The cost of breaking the cipher exceeds the value of the encrypted information.

  (2) The time required to break the cipher exceeds the useful lifetime of the information.
- An encryption scheme is said to be computationally secure if either of the foregoing two criteria are met.

ENCRYPTION TECHNIQUES

The two basic building blocks of all encryption techniques are :

1. Substitution techniques: A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

Different types of Substitution techniques are:

1. Caesar Cipher
2. Monoalphabetic Ciphers
3. Playfair Cipher
4. Hill Cipher
5. Polyalphabetic Ciphers
6. One-Time Pad

CAESAR CIPHER

- The Caesar Cipher technique is one of the earliest and simplest method of encryption technique. It's simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter some fixed number of positions down the alphabet. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.

Formula:

Cipher text(C):E(k,p)=(p+k)mod26

Plaintext(p):D(k,C)=(C-k)mod26

- For Example, key=3

  Plain text: hello how are you

  Cipher text: KHOOR KRZ DUH BRX

BETIC CIPHERS

- Better than Caeser Cipher
- For each character of alphabet, assign different-abrupt concerned character

Example:

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

- Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet.
- A countermeasure is to provide multiple substitutes, known as homophones, for a single letter.

For Example:

- Plaintext: goodmorning
- Cipher text: TLLWNLIMRMT

R CIPHER

- The best-known multiple-letter encryption cipher is the Playfair, which treats diagrams in the plaintext as single units and translates these units into ciphertext diagrams.
- The Playfair algorithm is based on the use of a *5 * 5 matrix* of letters constructed using a keyword.

For Example,

Keyword: security

Plaintext: pattern

- In this case, the keyword is security*. The matrix is constructed by filling* in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order.

Note: The letters I and J count as one letter.

R CIPHER

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

Plaintext is encrypted two letters at a time, according to the following rules

1) Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that pattern would be treated as *pa tx te rn*.

2)Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, op is encrypted as PL.

3)Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mv is encrypted as VE.

PLAYFAIR CIPHER

4) Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, pa becomes OB.

Plaintext: pa tx te rn

Ciphertext: OB VA FT CP

- The Playfair cipher is a great advance over simple monoalphabetic ciphers.
- For one thing, whereas there are only 26 letters, there are 26 * 26 = 676 diagrams, so that identification of individual diagrams is more difficult.

CIPHER

- This encryption algorithm takes m successive plaintext letters and substitutes for them m ciphertext letters.
- The substitution is determined by m linear equations in which each character is assigned a numerical value.

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
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

In general terms, the hill system can be expressed as:

$$C=E(P,K)=PK \bmod 26$$

$$P=D(C,K)=CK^{-1} \bmod 26$$

POLYALPHABETIC CIPHERS

- Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message.
- The general name for this approach is polyalphabetic substitution cipher.
- All these techniques have the following features in common:

1. A set of related monoalphabetic substitution rules isused.
2. A key determines which particular rule is chosen for a giventransformation.

For Example, key:

Plaintext:

deceptivewearediscovered

Cipher text:

ZICVTWQNGKZEIIGASXSTSLVVWLA

$$C_i = P_i + k_i \pmod{26}$$

$$\text{or} \quad \bmod m \pmod{26}$$

$$P_i \; C_i - k_i$$

$$\text{or} = \bmod m$$

POLYALPHABETIC CIPHERS

Key	Plaintext																									
	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
a	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
b	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	A
c	C	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
d	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
e	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	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	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

Table 1.4 : Polyalphabetic Ciphers

(VERNAM CIPHER)

ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

We now show two different decryptions using two different keys:

Cipher text: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

key: pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih

Plain text: mr mustard with the candlestick in the hall

Ccipher text: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

key: mfugpmiydgaxgoufhkllmhsqdqogtewbqfgyovuhwt

Plaintext: miss scarlet with the knife in the library

TRANSPOSITIO N TECHNIQUES

- A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher.
- The simplest such cipher is the rail fence technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.

For Example,

Plaintext: meet me after the toga party rail fence of depth: 2

The encrypted message is: MEMATRHTGPRYETEFETEOAAT

TRANSPOSITIO N TECHNIQUES

- A more complex scheme is to write the message in a rectangle, row by row, and read the
- message off, column by column, but permute the order of the columns. The order of the
- columns then becomes the key to the algorithm.

a	t	t	a	c	k	p
o	s	t	p	o	n	e
d	u	n	t	i	l	t
w	o	a	m	x	y	z

Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

- Key: 4 3 1 2 5 6
- Plaintext:

- The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is a more complex permutation that is not easily reconstructed. Thus, if the foregoing message is reencrypted using the same algorithm

TRANSPOSITIO N TECHNIQUES

Key: Plaintext: 4 3 1 2 5 6 7

t	t	n	a	a	p	t
m	t	s	u	o	a	o
d	w	c	o	i	x	k
n	l	y	p	e	t	z

Output: NSCYAUOPTTWLTMDNAOIEPAXTTOKZ

- To visualize the result of this double transposition, designate the letters in the original plaintext message by the numbers designating their position. Thus, with 28 letters in the message, the original sequence of letters is,

01 02 03 04 05 06 07 08 09 10 11 12 13 14

15 16 17 18 19 20 21 22 23 24 25 26 27 28

TRANSPOSITIO N TECHNIQUES

- After the first transposition we have,

03 10 17 24 04 11 18 25 02 09 16 23 01 08

15 22 05 12 19 26 06 13 20 27 07 14 21 28

- which has a somewhat regular structure. But after the second transposition, we have,

17 09 05 27 24 16 12 07 10 02 22 20 03 25

15 13 04 23 19 14 11 01 26 21 18 08 06 28

- This is a much less structured permutation and is much more difficult to cryptanalyze.

OGRAPHY

- A plaintext message may be hidden in one of two ways. The methods of steganography conceal the existence of the message, whereas the methods of cryptography render the message unintelligible to outsiders by various transformations of the text.
- Various other techniques have been used historically;
- some examples are the following :
- Character marking: Selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held at an angle to bright light.
- Invisible ink: A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to the paper.
- Pin punctures: Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light.

OGRAPHY

- Typewriter correction ribbon: Used between lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light.
- Steganography has a number of drawbacks when compared to encryption. It requires a lot of overhead to hide a relatively few bits of information, although using a scheme like that proposed in the preceding paragraph may make it more effective.
- Also, once the system is discovered, it becomes virtually worthless. This problem, too, can be overcome if the insertion method depends on some sort of key.
- Alternatively, a message can be first encrypted and then hidden using steganography.
- The advantage of steganography is that it can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered.
- Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide.

chapter-3

Modern Block Ciphers

BLOCK CIPHERS

- One of the most widely used types of cryptography algorithms
- Provide strong secrecy and/or authentication services
- In particular will introduce DES (Data Encryption Standard)

STREAM CIPHERS

- Block ciphers process messages into blocks, each of which is then encrypted/decrypted
- A substitution on very big characters
- 64-bits or more

- Stream ciphers process messages a bit or byte at a time when encrypted/decrypting
- Many current ciphers are block ciphers

CIPHER PRINCIPLES

- Block ciphers look like an extremely large substitution
- Would need table of 2^{64} entries for a 64-bit block
- Arbitrary reversible substitution cipher for a large block size is not practical
- 64-bit general substitution block cipher, key size 2^{64}!
- Most symmetric block ciphers are based on a **feistel cipher structure**
- Needed since must be able to **decrypt** ciphertext to recover messages efficiently

SUBSTITUTION-PERMUTATION CIPHERS

- In 1949 shannon introduced idea of substitution-permutation (S-P) networks
- Modern substitution-transposition product cipher
- These form the basis of modern block ciphers
- S-P networks are based on the two primitive cryptographic operations we have seen before:
- Substitution (s-box)
- Permutation (p-box) (transposition)
- Provide confusion and diffusion of message

AND CONFUSION

- Introduced by claude shannon to thwart cryptanalysis based on statistical analysis
- Assume the attacker has some knowledge of the statistical characteristics of the plaintext
- Cipher needs to completely obscure statistical properties of original message
- A one-time pad .

AND CONFUSION

- More practically Shannon suggested combining elements to obtain:

- **Diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext

- **Confusion** – makes relationship between ciphertext and key as complex as possible

CIPHER STRUCTURE

- Horst feistel devised the **feistel cipher**
- Implements shannon's substitution-permutation network concept
- Partitions input block into two halves
- Process through multiple rounds
- Perform a substitution on left data half
- Based on round function of right half & subkey
- Permutation swapping halves

CIPHER STRUCT

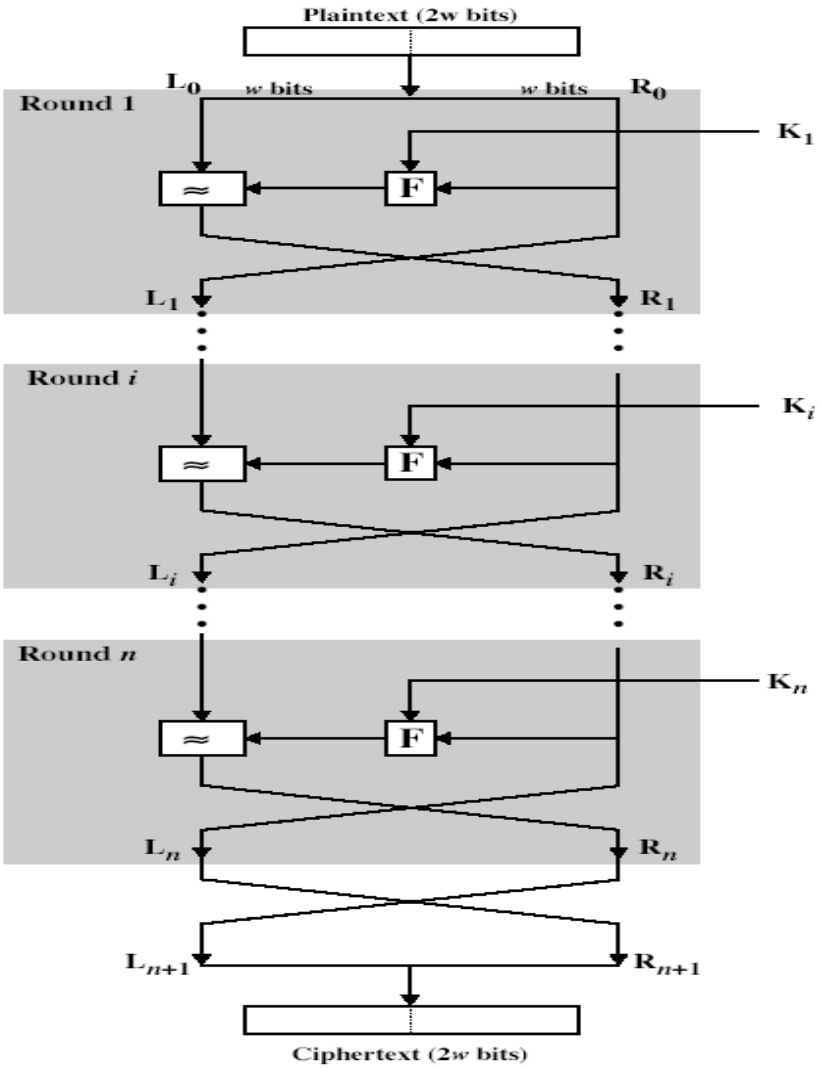


Figure 1.14 : Feistel Cipher Structure

FEISTEL CIPHER

N sequential rounds

A substitution on the left half l_i

1. Apply a round function F to the right half r_i and

2. Take XOR of the output of (1) and l_i

The round function is parameterized by the subkey k_i

K_i are derived from the overall key K

DESIGN PRINCIPLES

- **Block size**
- Increasing size improves security, but slows cipher

Key size

- Increasing size improves security, makes exhaustive key searching harder, but may slow cipher

Number of rounds

- Increasing number improves security, but slows cipher

Sub key generation

- Greater complexity can make analysis harder, but slows cipher

Round function

- Greater complexity can make analysis harder, but slows cipher

Fast software en/decryption & ease of analysis

- Are more recent concerns for practical use and testing

CIPHER
DECF

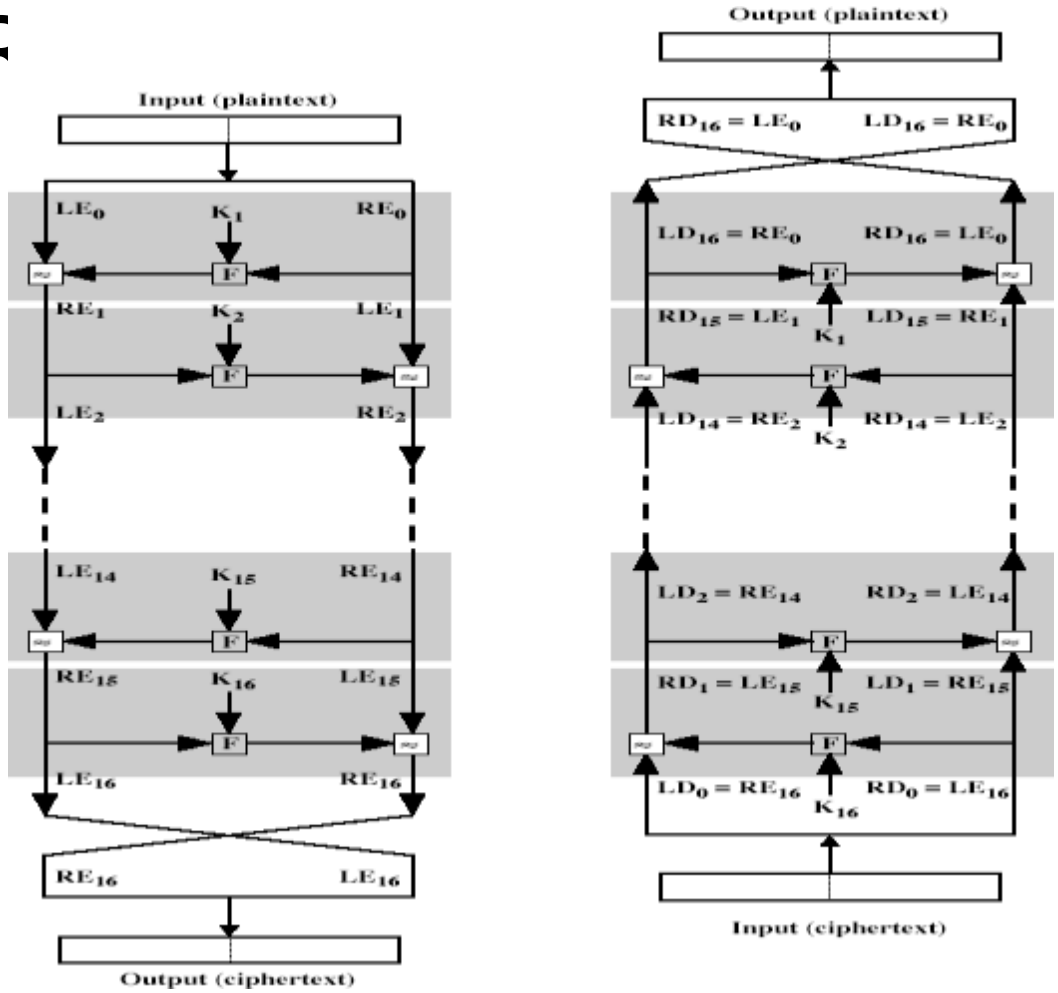


Figure 1.15 :Feistel Cipher Decryption

DATA ENCRYPTION STANDARD (DES)

- Most widely used block cipher in world
- Adopted in 1977 by NBS (now NIST)
- As FIPS PUB 46
- Encrypts 64-bit data using 56-bit key
- Has widespread use

HISTORY

- IBM developed Lucifer cipher by team led by Feistel used 64-bit data blocks with 128-bit key
- Then redeveloped as a commercial cipher with input from NSA and others
- In 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES

CONTROVERSY

- Although DES standard is public
- was considerable controversy over design
- in choice of 56-bit key (vs Lucifer 128-bit)
- subsequent events and public analysis show in fact design was appropriate
- DES has become widely used, especially in financial applications

ENCRYPTION

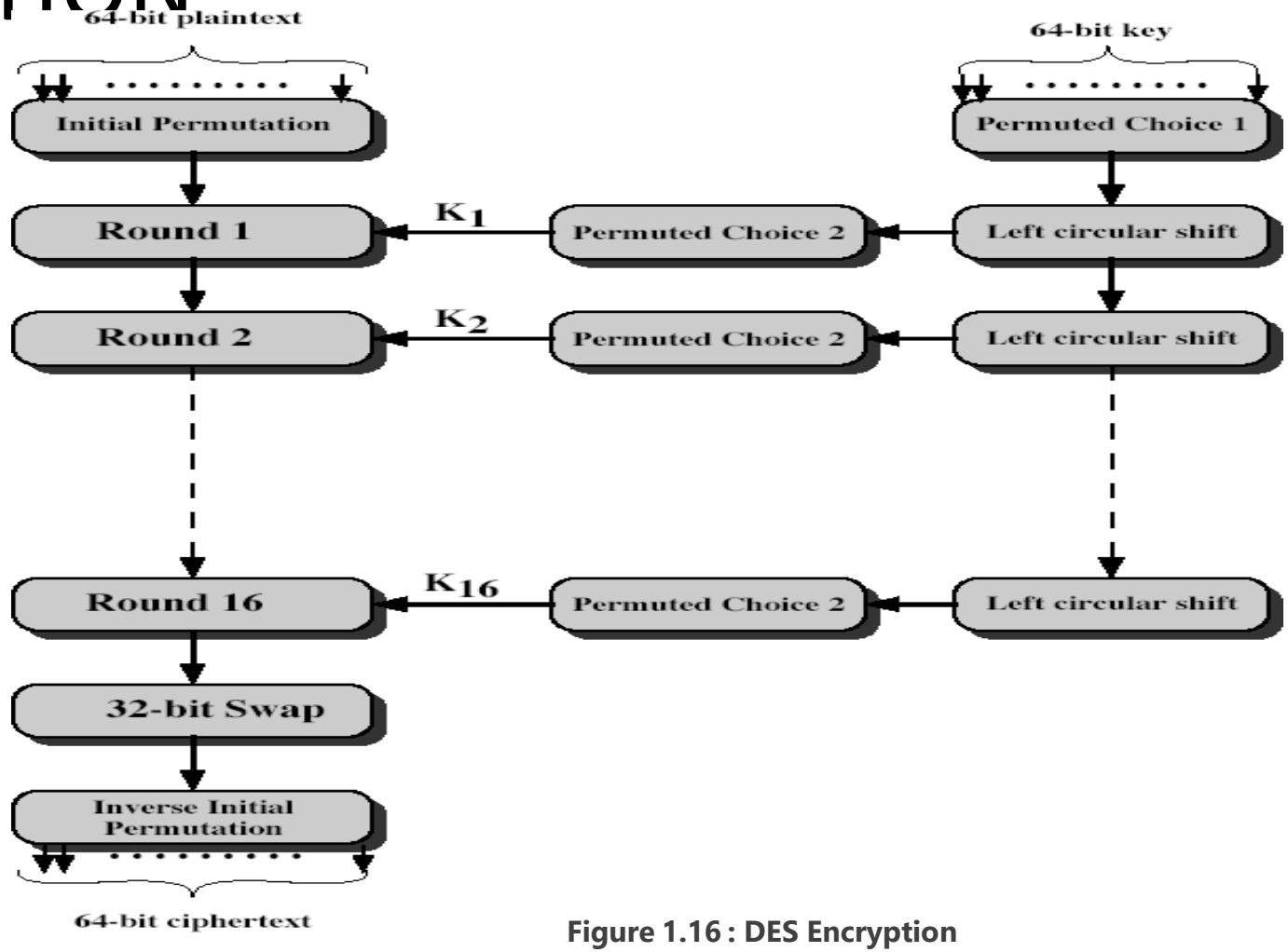


Figure 1.16 : DES Encryption

PERMUTATION IP

- First step of the data computation
- IP reorders the input data bits
- Quite regular in structure
- Example:

  ip(675a6967 5e5a6b5a) = (ffb2194d 004df6fb)

DES ROUND STRUCTURE

- Uses two 32-bit L & R halves
- As for any feistel cipher can describe as:

$L_i = r_{i-1}$

$R_i = l_{i-1}$ xor f(r_{i-1}, k_i)

- Takes 32-bit R half and 48-bit subkey and:
- Expands R to 48-bits using expansion permutation E
- Adds to subkey
- Passes through 8 s-boxes to get 32-bit result
- Finally permutes this using 32-bit permutation function

FUNCTION F(R,K)

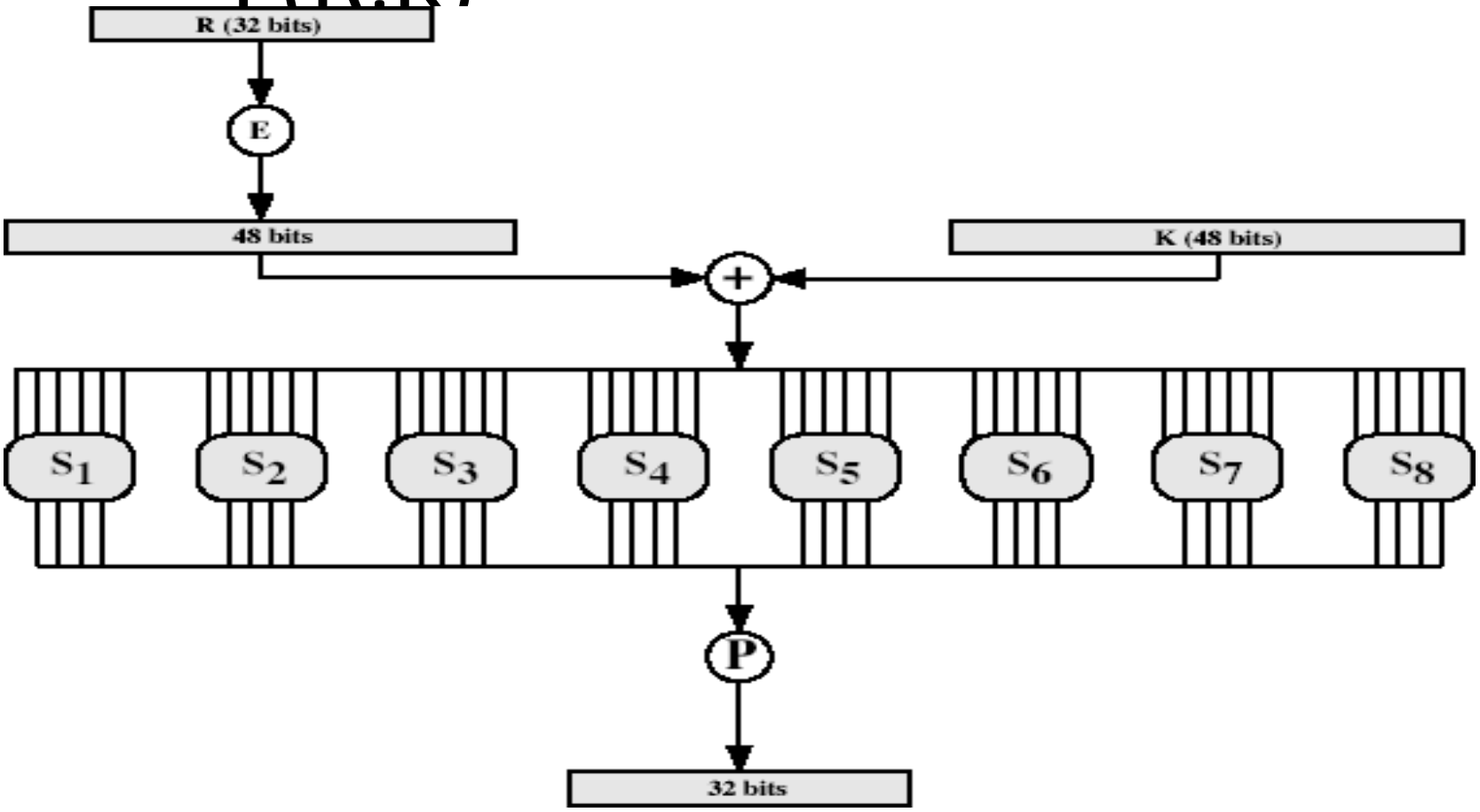


Figure 1.17 : Round Function

SUBSTITUTION BOXES S

8 s-boxes

Each s-box maps 6 to 4 bits

- Outer bits 1 & 6 (**row** bits) select the row
- Inner bits 2-5 (**col** bits) select the column

For example, in S1, for input 011001,

The row is 01 (row 1)

The column is 1100 (column 12).

The value in row 1, column 12 is 9

The output is 1001.

Result is 8 x 4 bits, or 32 bits

SCHEDULE

Forms subkeys used in each round

1. Initial permutation of the key PC1

2. Divide the 56-bits in two 28-bit halves

3. At each round

1. Left shift each half (28bits) separately either 1 or 2 places based on the left shift schedule shifted values will be input for next round

2. Combine two halfs to 56 bits, permuting them by PC2 for use in function f.

Pc2 takes 56-bit input, outputs 48 bits

DECRYPTION

- Decrypt must unwind steps of data computation
- With feistel design, do encryption steps again
- Using sub keys in reverse order (SK16 … SK1)
- Note that IP undoes final FP step of encryption
- 1st round with SK16 undoes 16th encrypt round

  ……
- 16th round with sk1 undoes 1st encrypt round Then final fp undoes initial encryption ip thus recovering original data value

(REVERSE ENCRYPTION)

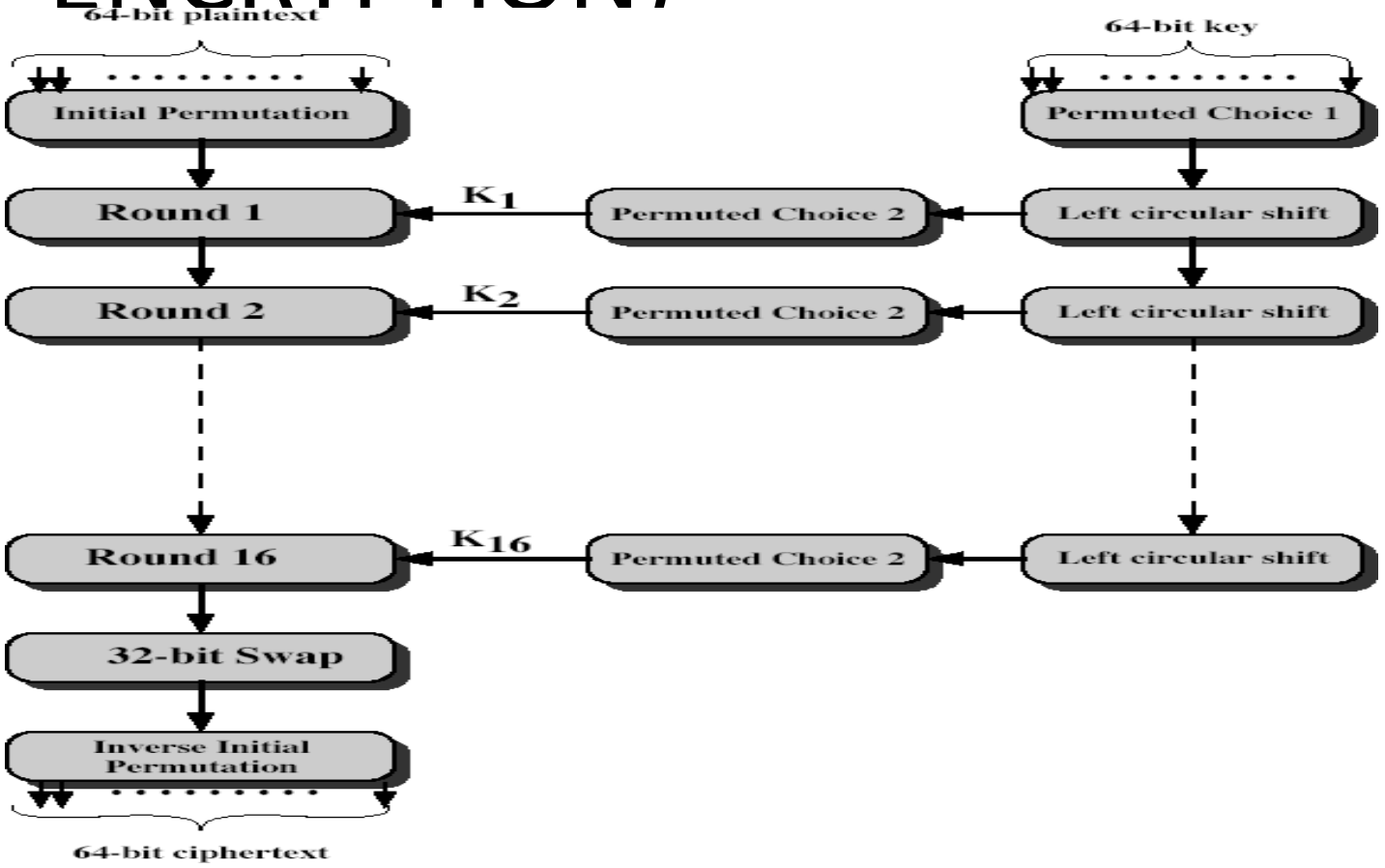


Figure 1.18 :DES Decryption

HE EFFECT

- Key desirable property of encryption alg

- Des exhibits strong avalanche

- Where a change of **one** input or key bit results in changing approximate **half** output bits

STRENGTH OF DES – KEY SIZE

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- Brute force search looks hard
- Recent advances have shown is possible
- In 1997 on internet in a few months
- In 1998 on dedicated hardware (EFF) in a few days
- In 1999 above combined in 22hrs!
- Still must be able to recognize plaintext
- Now considering alternatives to DES

STRENGTH OF DES – TIMING ATTACKS

- Attacks actual implementation of cipher
- Use knowledge of consequences of implementation to derive knowledge of some/all subkey bits
- Specifically use fact that calculations can take varying times depending on the value of the inputs to it

STRENGTH OF DES – ANALYTIC ATTACKS

- In DES have several analytic attacks on DES
- These utilise some deep structure of the cipher
- By gathering information about encryptions can eventually recover some/all of the sub-key bits
- If necessary then exhaustively search for the rest
- Generally these are statistical attacks include
- Differential cryptanalysis
- Linear cryptanalysis
- Related key attacks

DIFFERENTIAL CRYPTANALYSIS

- One of the most significant recent (public) advances in cryptanalysis
- Known in 70's with DES design
- Murphy, biham & shamir published 1990
- Powerful method to analyse block ciphers
- Used to analyse most current block ciphers with varying degrees of success
- DES reasonably resistant to it

DIFFERENTIAL CRYPTANALYSIS

- A statistical attack against feistel ciphers
- Uses cipher structure not previously used
- Design of S-P networks has output of function *f* influenced by both input & key
- Hence cannot trace values back through cipher without knowing values of the key
- Differential cryptanalysis compares two related pairs of encryptions

DIFFERENTIAL CRYPTANALYSIS COMPARES PAIRS OF ENCRYPTIONS

- Differential cryptanalysis is complex
- With a known difference in the input
- Searching for a known difference in output

$$\Delta m_{i+1} = m_{i+1} \oplus m'_{i+1}$$

$$= \left[m_{i-1} \oplus \mathrm{f}\left(m_i, K_i\right)\right] \oplus \left[m'_{i-1} \oplus \mathrm{f}\left(m'_i, K_i\right)\right]$$

$$= \Delta m_{i-1} \oplus \left[\mathrm{f}\left(m_i, K_i\right) \oplus \mathrm{f}\left(m'_i, K_i\right)\right]$$

DIFFERENTIAL CRYPTANALYSIS

- Cryptanalysis some input difference giving some output difference with probability *p*
- If find instances of some higher probability input / output difference pairs occurring can infer sub key that was used in round
- Must iterate process over many rounds

DIFFERENTIAL CRYPTANALYSIS

- Perform attack by repeatedly encrypting plaintext pairs with known input XOR until obtain desired output XOR
- When found
  - If intermediate rounds match required XOR have a **right pair**
  - If not then have a **wrong pair**
- Can then deduce keys values for the rounds
  - Right pairs suggest same key bits
  - Wrong pairs give random values
- Larger numbers of rounds makes it more difficult
- Attack on full DES requires an effort on the order of 2^{47}, requiring 2^{47} chosen plaintexts to be encrypted

CRYPTANALYSIS

- Another recent development
- Also a statistical method
- Based on finding linear approximations to model the transformation of DES
- Can attack DES with 2^{47} known plaintexts, still in practise infeasible

FOR S-BOXES

- No output of any S-Box is too close to a linear function of the input bits
- Each row of an S-Box includes all 16 possible output bit combinations
- If two inputs to an S-box differ in one bit, the output bits differ in at least two bits
- If two inputs differ is the two middle bits, outputs must differ at least two bits
- Defend against differential analysis and provide good confusion properties

DESIGN PRINCIPLES

- Basic principles still like feistel in 1970's
- Number of rounds
  - More is better, makes exhaustive search best attack
  - 16 rounds: brute force 2^{55}
  - Differential analysis: $2^{55.1}$

DESIGN PRINCIPLES

Function F:

- Provides “confusion”, is nonlinear, avalanche
- Strict avalanche criterion (SAC)

  Any output bit i should change with p=1/2 when any single input bit j is inverted, for all i,j

  Applies to both s-boxes and the overall F function

Key schedule

- No general rule has been discovered
- Complex subkey creation, key avalanche

MODES OF OPERATION

- Block ciphers encrypt fixed size blocks

Eg. DES encrypts 64-bit blocks, with 56-bit key

- Need way to use in practise, given usually have arbitrary amount of information to encrypt

- Four were defined for DES in ANSI standard **ANSI X3.106-1983 modes of use**

  DES is the basic building block

- Have **block** and **stream** modes

CODEBOOK BOOK (ECB)

- Message is broken into independent blocks which are encrypted
- Each block is a value which is substituted, like a codebook, hence name
- Each DES is a very complex 64-bit to 64-bit substitution
- Each block is encoded independently of the other blocks
- $C_i = DES_{K1}(p_i)$
- Uses: secure transmission of single values
- Repeated input blocks have same output
- Not secure for long transmission

CODEBOOK BOOK (ECB)

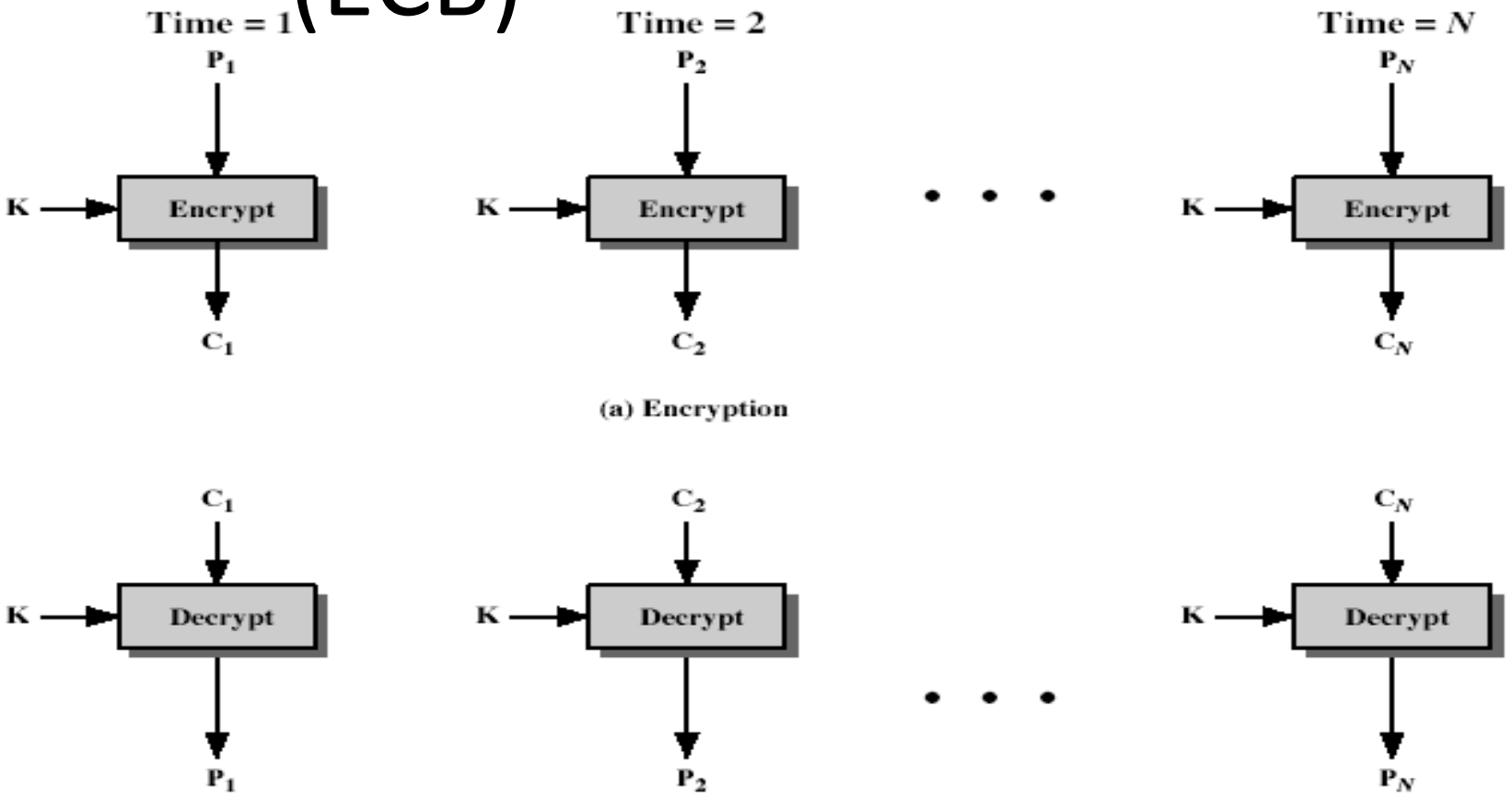


Figure 1.19 :ECE Decryption

ADVANTAGES AND LIMITATIONS OF ECB

- Repetitions in message may show in ciphertext

  If aligned with message block

  Particularly with data such graphics

  Or with messages that change very little, which become a code-book analysis problem

- Weakness due to encrypted message blocks being independent

- Main use is sending a few blocks of data

CHAINING (CBC)

- Message is broken into blocks
- But these are linked together in the encryption operation
- Each previous cipher blocks is chained with current plaintext block, hence name
- Use initial vector (IV) to start process

  $C_i = des_{k1}(p_i \text{ XOR } c_{i-1})$

  $C_{-1} = iv$

- Uses: bulk data encryption, authentication

CHAINING (CBC)

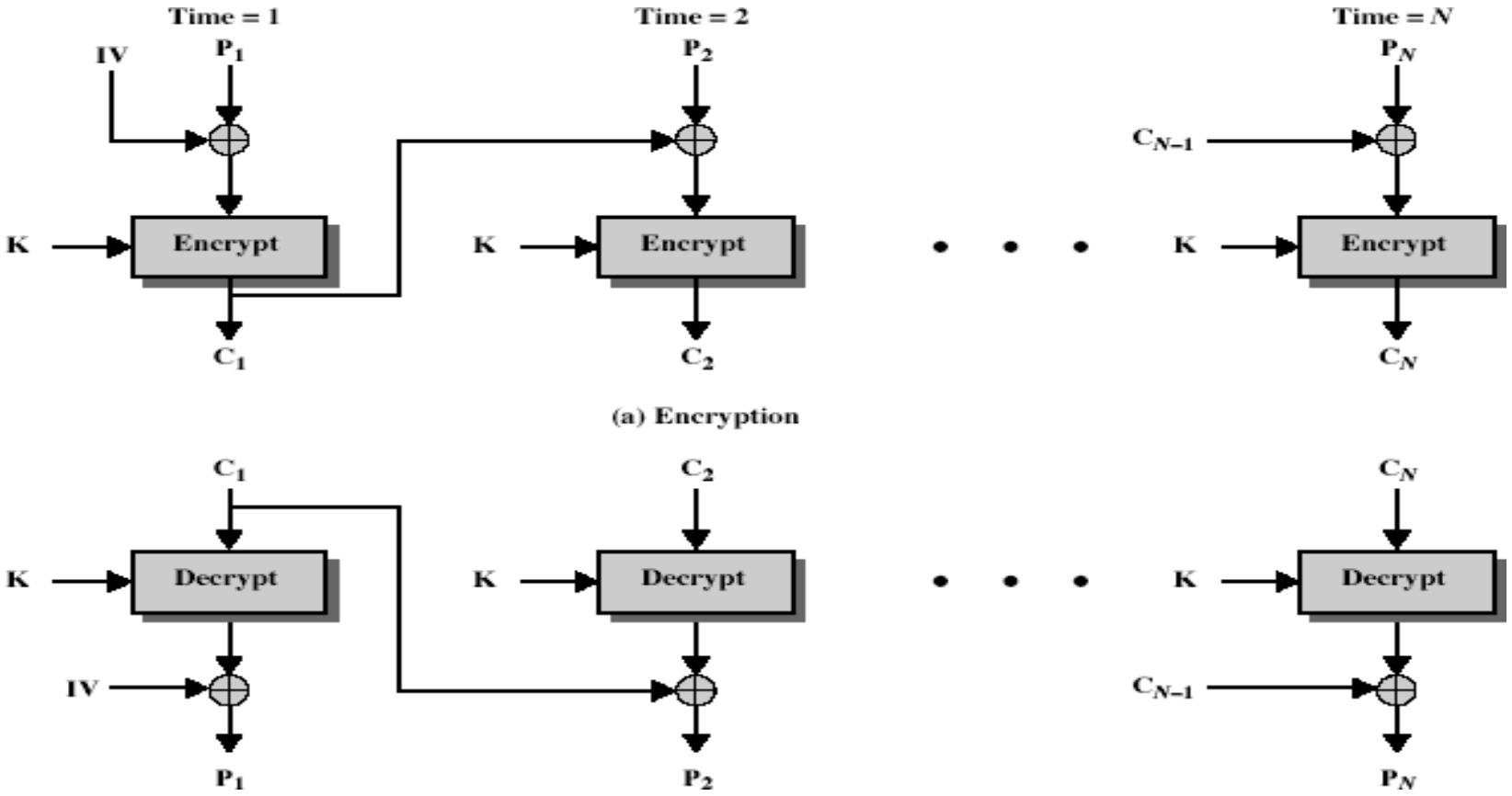


Figure 1.20 : CBC Decryption

ADVANTAGES AND LIMITATIONS OF CBC

- Each ciphertext block depends on **all** message blocks
- Thus a change in the message affects all ciphertext blocks after the change as well as the original block
- Need **initial value** (IV) known to sender & receiver
- However if IV is sent in the clear, an attacker can change bits of the first block, and change IV to compensate
- Hence either IV must be a fixed value (as in EFTPOS) or it must be sent encrypted in ECB mode before rest of message

FEED BACK (CFB)

- Message is treated as a stream of bits
- Added to the output of the block cipher
- Result is feed back for next stage (hence name)
- Standard allows any number of bit (1,8 or 64 or whatever) to be feed back
- Denoted CFB-1, CFB-8, CFB-64 etc
- Is most efficient to use all 64 bits (CFB-64)

$C_i = p_i$ XOR $des_{k1}(c_{i-1})$

$C_{-1} = iv$

Uses: stream data encryption, authentication

FEED BACK (CFB)

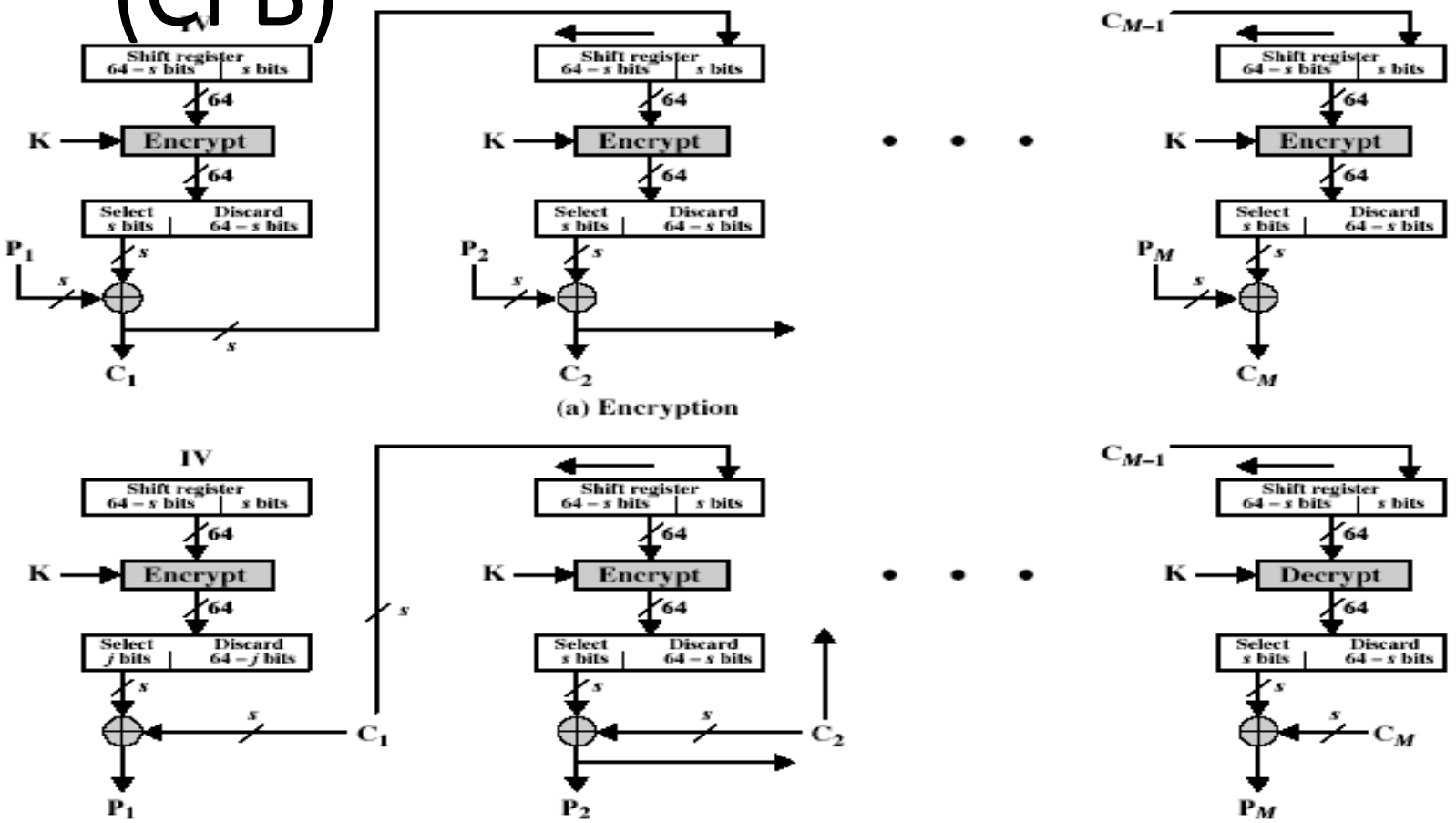


Figure 1.21: CFB Decryption

ADVANTAGES AND LIMITATIONS OF CFB

- Appropriate when data arrives in bits/bytes
- Most common stream mode
- Note that the block cipher is used in **encryption** mode at **both** ends
- Errors propagate for several blocks after the error
- Must use over a reliable network channel

FEED BACK (OFB)

- message is treated as a stream of bits
- output of cipher is added to message
- output is then feed back (hence name)
- feedback is independent of message
- can be computed in advance

$C_i = P_i$ XOR O_i

$O_i = DES_{K1}(O_{i-1})$

$O_{-1} = IV$

uses: stream encryption over noisy channels

FEED BACK (OFB)

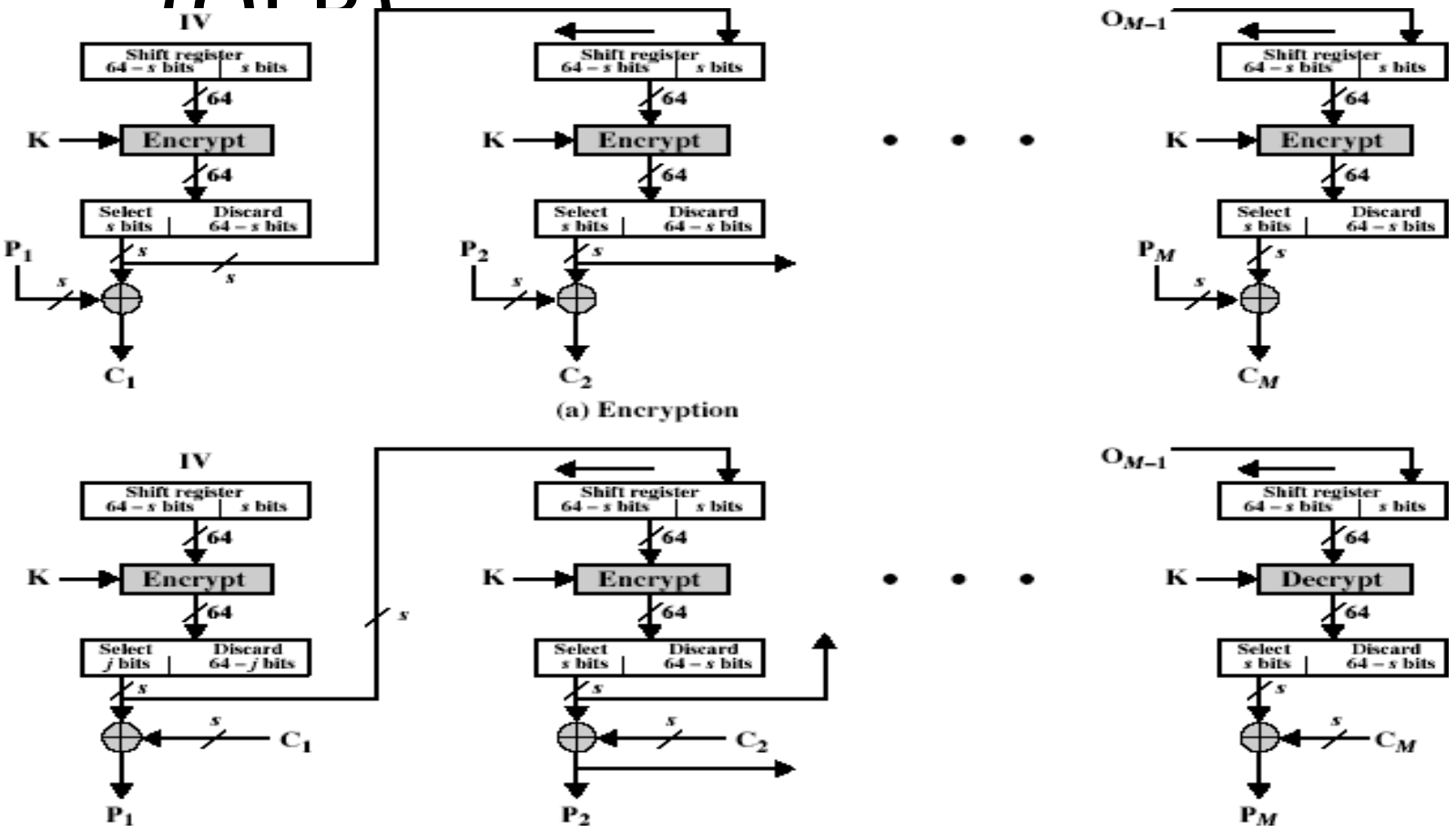


Figure 1.22 : OFB Decryption

LIMITATIONS OF OFB

- Used when error feedback a problem or where need to encryptions before message is available
- Superficially similar to CFB
- But feedback is from the output of cipher and is independent of message
- Errors do not propagate
- Sender and receiver must remain in sync, and some recovery method is needed to ensure this occurs
- Because the "random" bits are independent of the message, they must **never** be used more than once
- Otherwise the 2 ciphertexts can be combined, cancelling these bits)

ER (CTR)

- A “new” mode, though proposed early on
- Encrypts counter value rather than any feedback value
- Must have a different key & counter value for every plaintext block (never reused)

$C_i = p_i$ XOR o_i

$O_i = des_{k1}(i)$

Uses: high-speed network encryptions

ER (CTR)

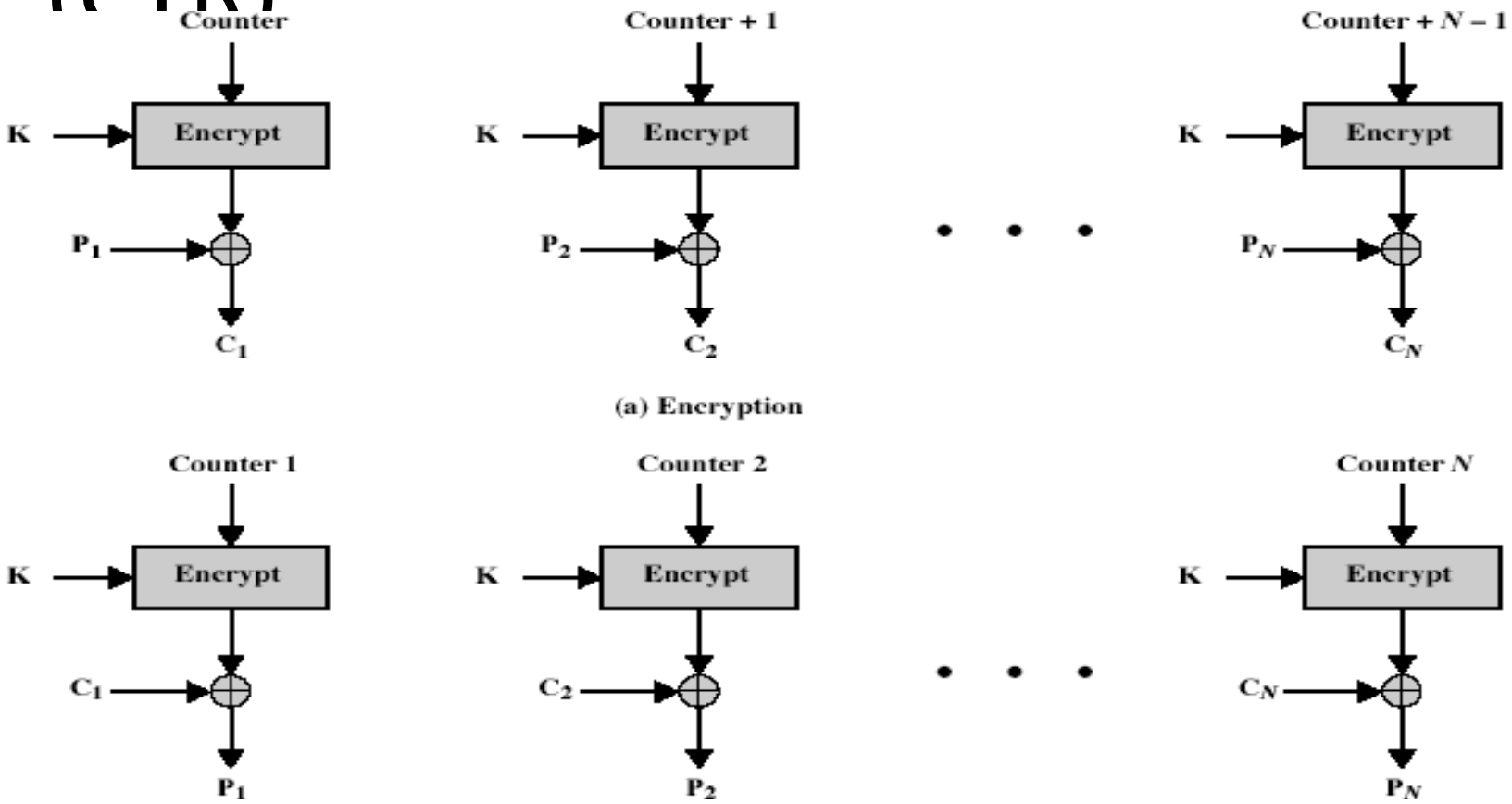


Figure 1.23 :CTR Decryption

ADVANTAGES AND LIMITATIONS OF CTR

- Efficiency
- Can do parallel encryptions
- In advance of need
- Good for bursty high speed links
- Random access to encrypted data blocks
- Do not have to decode from the beginning
- Provable security (good as other modes)
- But must ensure never reuse key/counter values, otherwise could break (cf OFB)

ENCRYPTION STANDARD(AES)

- Replacement for DES was needed
- Theoretical attacks that can break it
- Demonstrated exhaustive key search attacks
- Can use Triple DES – but slow, small block size
- NIST issued a call for a new AES in 1997
- 15 candidates accepted in Jun 1998
- 5 candidates were short-listed in Aug 1999
- Rijndael was selected as the AES in Oct 2000
- Published as FIPS PUB 197 standard in Dec 2001

REQUIREMENTS

- Symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger & faster than triple DES
- Active life of 20-30 years (+ archival use)
- Provide full specification & design details
- Both C & Java implementations
- NIST have released all submissions & unclassified analyses

EVALUATION CRITERIA

- Initial criteria:
- Security – effort for practical cryptanalysis
- Cost – in terms of computational efficiency (speed, memory)
- Algorithm & implementation characteristics
- Flexibility, algorithm simplicity
- Final criteria
- General security
- Ease of software & hardware implementation
- Restricted-space environments
- Attacks on implementations
- Timing attack, power analysis
- Flexibility (in en/decrypt, keying, other factors)

SHORT-LIST

- After testing and evaluation, short-list in Aug 1999:
- MARS (IBM) - complex, fast, high security margin
- RC6 (USA) - very simple, very fast, low security margin
- Rijndael (Belgium) - clean, fast, good security margin
- Serpent (Euro) - clean, slow, very high security margin
- Twofish (USA) - complex, very fast, high security margin
- Then subject to further analysis & comment
- Saw contrast between algorithms with
- Few complex rounds vs. many simple rounds
- Refined existing ciphers vs. new proposals

CIPHER - RIJNDAEL

- Designed by Rijmen-Daemen in Belgium
- Block length: 128 bits
- Key length: 128/192/256 bits
- Number of Rounds: 10/12/14 rounds
- An iterated cipher (rather than Feistel cipher)
- Processes data as block of 4 columns of 4 bytes
- Operates on entire data block in every round
- Designed to be:
- Resistance against all known attacks
- Speed and code compactness on a wide range of platforms
- Design simplicity

AES STRUCTURE

- Data block of 4 columns of 4 bytes is “state”
- Key is expanded to array of words
- Has 9/11/13 rounds in which state undergoes:
- Substitute bytes (1 S-box used on every byte)
- Shift rows (permute bytes between columns)
- Mix columns (substitute using matrix multiplication of columns)
- Add round key (XOR state with key material)
- View as alternating XOR key & scramble data bytes
- Initial XOR key material & incomplete last round
- With fast XOR & table lookup implementation

ENCRYPTION & DECRYPTION

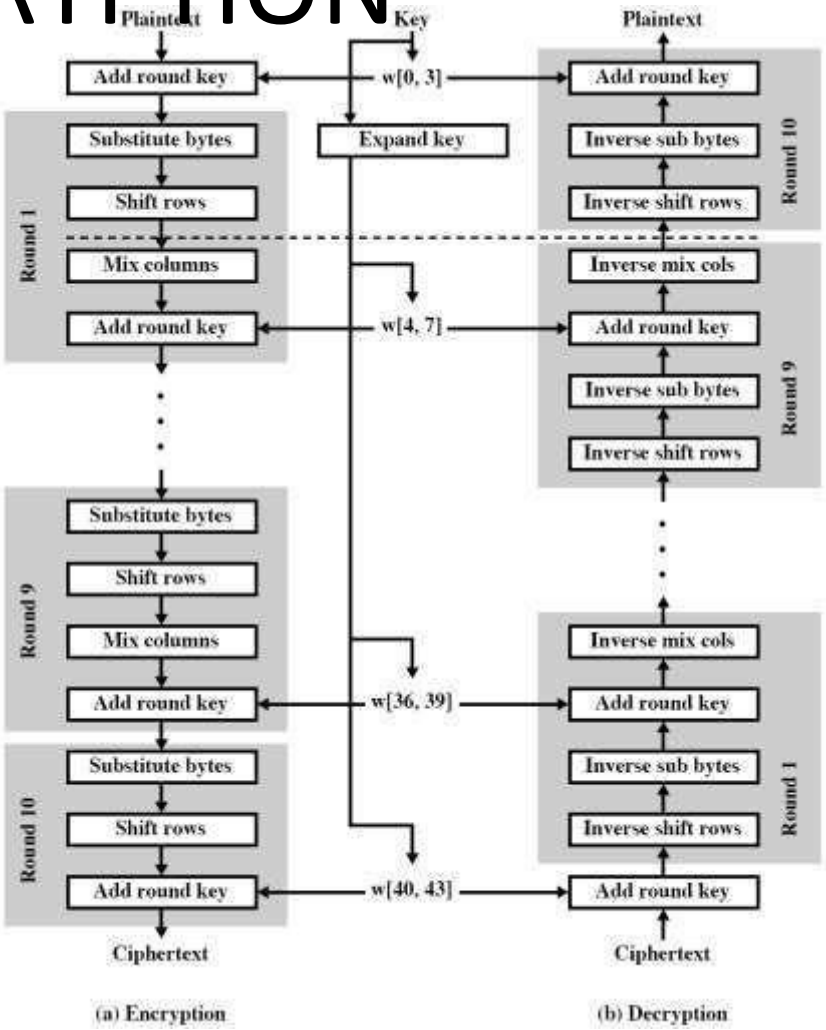
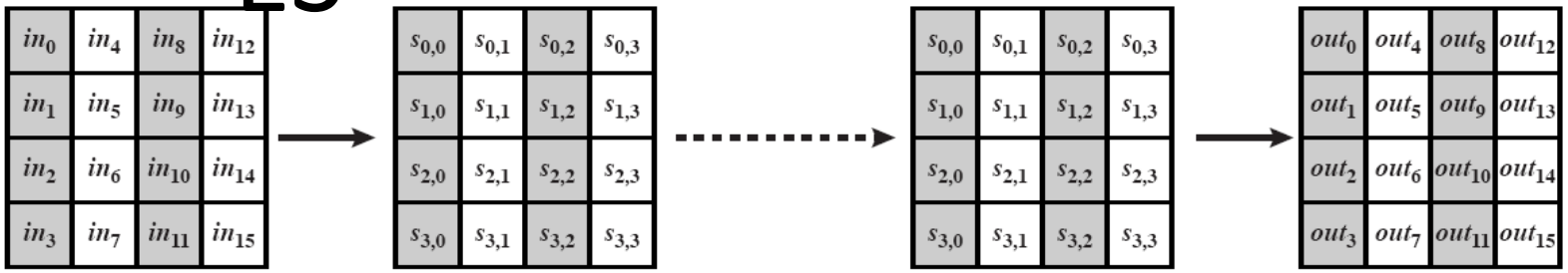


Figure 1.24 :AES Encryption & Decryption

STRUCTURES



(a) Input, state array, and output

(b) Key and expanded key

Figure 1.25 : AES Data Structures

ENCRYPTION

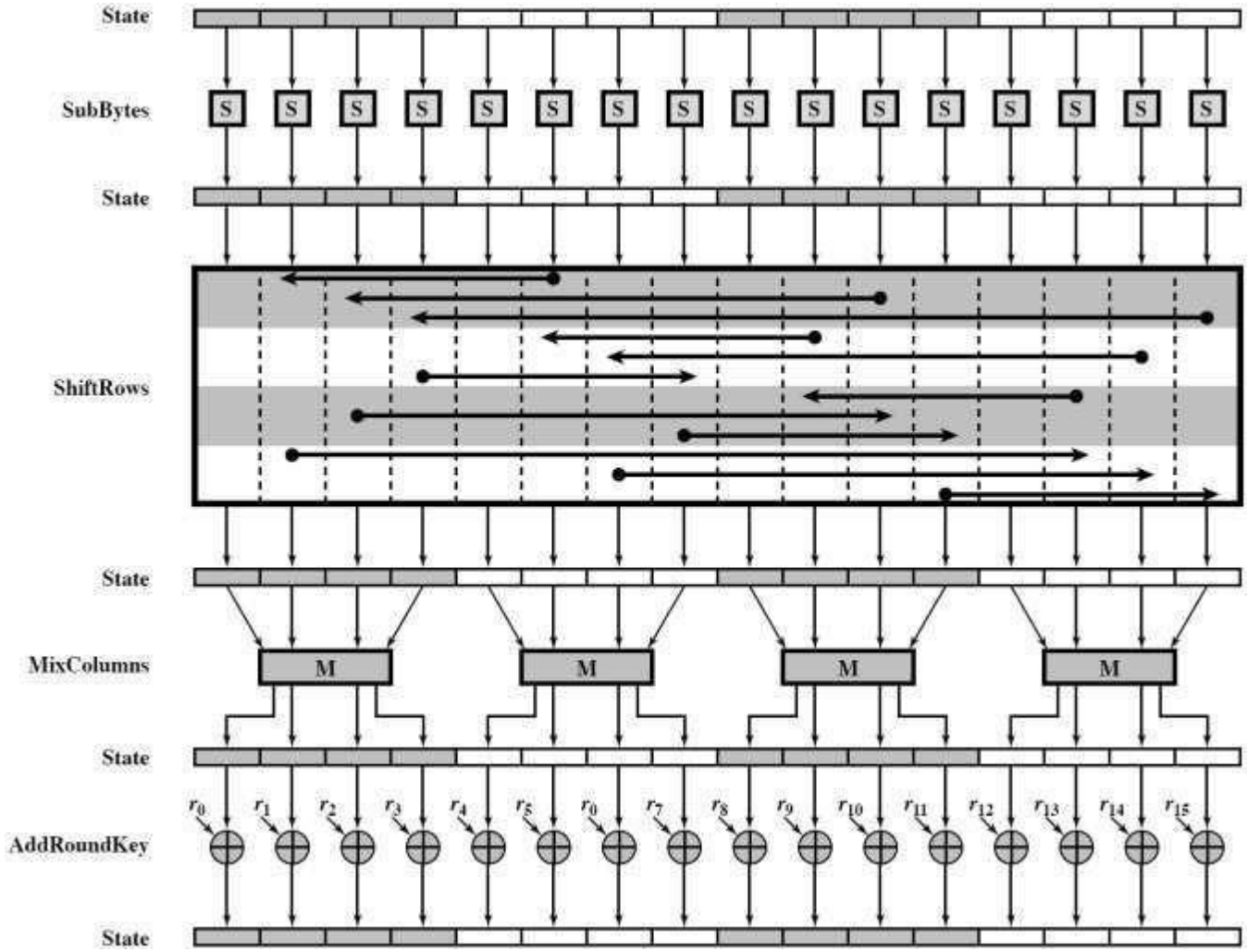


Figure 1.26 : AES Encryption Round

RC4

- A proprietary cipher owned by RSA DSI
- Another ron rivest design, simple but effective
- Variable key size, byte-oriented stream cipher
- Widely used (web SSL/TLS, wireless WEP)
- Key forms random permutation of all 8-bit values
- Uses that permutation to scramble input info processed a byte at a time

SECURITY

- Claimed secure against known attacks
- Have some analyses, none practical
- Result is very non-linear
- Since RC4 is a stream cipher, must never reuse a key
- Have a concern with WEP, but due to key handling rather than RC4 itself

CONTENT BEYOND SYLLABUS

OUT LINE

- ISO 27001
- ISO FEATURES
- INFORMATION ASSET CLASSIFICATION
- PEOPLE ASSETS
- SECURITY INCIDENTS

27001

ISO 27001: This International Standard covers all types of organizations (e.g. commercial enterprises, government agencies, non-profit organizations). This International Standard specifies the requirements for establishing; implementing, operating, monitoring, reviewing, maintaining and improving documented ISMS within the context of the organization's overall business risks. It specifies requirements for the implementation of security controls customized to the needs of individual organizations or parts thereof. The ISMS is designed to ensure the selection of adequate and proportionate security controls that protect information assets and give confidence to interested partie.

Features

Features of ISO 27001

- Plan, Do, Check, Act (PDCA) Process Model
- Process Based Approach

Stress on Continual Process Improvements

- Scope covers Information Security not only IT Security
- Covers People, Process and Technology
- 5600 plus organizations worldwide have been certified
- 11 Domains, 39 Control objectives, 133 controls

ASSET CLASSIFICATION

- **CONFIDENTIAL:** If this information is leaked outside Organization, it will result in major financial and/or image loss.
- Compromise of this information will result in statutory, legal non- compliance. Access to this information must be restricted based on the concept of need-to-know.
- Disclosure requires the information owner's approval.
- In case information needs to be disclosed to third parties a signed confidentiality agreement is also required.
- Examples include Customer contracts, rate tables, process documents and new product development plans.

ASSET CLASSIFICATION

- **INTERNAL USE ONLY:** If this information is leaked outside Organization, it will result in Negligible financial loss and/or embarrassment. Disclosure of this information shall not cause serious harm to Organization, and access is provided freely to all internal users. Examples include circulars, policies, training materials etc.

- **PUBLIC:** Non availability will have no effect. If this information is leaked outside Organization, it will result in no loss. This information must be explicitly approved by the Corporate Communications Department or Marketing Department in case of marketing related information, as suitable for public dissemination. Examples include marketing brochures, press releases.

PEOPLE ASSETS

- Information is accessed or handled by the people from within the organization as well as the people related to organization for business requirements.
- It becomes necessary to identify such people from within the organization as well as outside the organization who handle the organization's information assets.
- The analysis such people, who has access rights to the assets of the organization, is to be done by Business Process Owner i.e. process / function head.
- The people assets shall include roles handled by
    a. Employees
    b. Contract Employees
    c. Contractors & his employees

INCIDENTS

Report Security Incidents (IT and Non-IT) to Helpdesk through

- E-mail to info.sec@organisation.com
- Telephone : xxxx-xxxx-xxxx
- Anonymous Reporting through Drop boxes

e.g.: IT Incidents: Mail Spamming, Virus attack, Hacking, etc.

Non-IT Incidents: Unsupervised visitor movement, Information leakage, Bringing unauthorized Media

- Do not discuss security incidents with any one outside organisation
- Do not attempt to interfere with, obstruct or prevent anyone from reporting incidents

RESOURCES

- **Lecture Notes - Lecture Notes**
- **Video Lectures - Video Lecture**
- **E-Book - Information Security Concepts**
- **Model Papers - JNTUH Question Papers**

DEPT & SEM : CSE-CS & I SEM

SUBJECT NAME: INFORMATION SECURITY

COURSE CODE : IS

UNIT : II

PREPARED BY : Anusha k

OUTLINE

- **Introduction to Number theory**
- **Integer Arithmetic, Modular Arithmetic**
- **Matrices, Linear Congruence**
- **Algebraic Structures, GF(2n) Fields**
- **Primes, Primality Testing**
- **Factorization, Chinese remainder Theorem**
- **Quadratic Congruence, Exponentiation and**
- **Logarithm**

MOTIVATION

- ***Number theory*** is the part of mathematics devoted to the study of the integers and their properties.
- Key ideas in number theory include *divisibility* and the *primality* of integers.
- Representations of integers, including binary and hexadecimal representations, are part of number theory.
- Number theory has long been studied because of the beauty of its ideas, its accessibility, and its wealth of open questions.

ARITHMETIC

- In the modern world, we use *decimal,* or *base* 10, *notation* to represent integers. For example when we write 965, we mean $9 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$.

- We can represent numbers using any base *b*, where *b* is a positive integer greater than 1.

- The bases *b* = 2 (*binary*), *b* = 8 (*octal*) , and *b*= 16 (*hexadecimal*) are important for computing and communications.

ARITHMETIC

- Modular arithmetic is 'clock arithmetic'
- A **congruence** a = b mod n says when divided by *n* that a and b have the same remainder

  100 = 34 mod 11

  Usually have 0<=b<=n-1

  -12mod7 = -5mod7 = 2mod7 = 9mod7

  B is called the **residue** of a mod n

- Can do arithmetic with integers modulo n with all results between 0 and n

ARITHMETIC-OPERATIONS

- **Addition**

  a+b mod n

- **Subtraction**

  a-b mod n = a+(-b) mod n

- **Multiplication**

  a.b mod n

derived from repeated addition can get a.b=0 where neither a,b=0

Example: 2.5 mod 10

ARITHMETIC-OPERATIONS

- **Division**

a/b mod n

is multiplication by inverse of b: $a/b = a.b^{-1} \bmod n$

if n is prime $b^{-1} \bmod n$ exists s.t $b.b^{-1} = 1 \bmod n$

Example: $2.3=1 \bmod 5$ hence $4/2=4.3=2 \bmod 5$

RICE S

Use the matrix to focus measures where they are needed, and to be aware of what measures are being (purposely) neglected.

- Drawing a threat/risk landscape. What areas are most at risk? Acceptable downtime.
- Define future measures, baselines, or project specific security
- Relating security topics.
- Dept. & diversity of defense
- List/audit current measures
- Follow changes in focus over time
- Divide “Computer Equipment” according to your needs

e.g. : OS, DBs, Middleware, Applications

RICE S

- Prevention - Physical, technical, continual re-assessment, resource isolation, ...
- Detection-Audits, looking for unusual behavior Panic? ..
- Reaction -disciplinary action, forensics/detective work

CONGRUENCE

Definition: If a and b are integers and m is a positive integer, then a is *congruent* to b *modulo* m if m divides $a - b$.

- The notation $a \equiv b \pmod{m}$ says that a is congruent to b modulo m.
- We say that $a \equiv b \pmod{m}$ is a *congruence* and that m is its *modulus.*
- Two integers are congruent mod m if and only if they have the same remainder when divided by m.
- If a is not congruent to b modulo m, we write
- $a \not\equiv b \pmod{m}$

Example: Determine whether 17 is congruent to 5 modulo 6 and whether 24 and 14 are congruent modulo 6.

Solution:

$17 \equiv 5 \pmod{6}$ because 6 divides $17 - 5 = 12$.

$24 \not\equiv 14 \pmod{6}$ since 6 divides $24 - 14 = 10$ is not divisible by 6.

MANIPULATION OF CONGRUENCES

- Multiplying both sides of a valid congruence by an integer preserves validity.
  - If $a \equiv b \pmod{m}$ holds then $c \cdot a \equiv c \cdot b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.

- Adding an integer to both sides of a valid congruence preserves validity.
  - If $a \equiv b \pmod{m}$ holds then $c + a \equiv c + b \pmod{m}$, where c is any integer, holds by Theorem 5 with $d = c$.

- However, **dividing** a congruence by an integer **does not** always produce a valid congruence.

Example: The congruence $14 \equiv 8 \pmod{6}$ holds. However, dividing both sides by 2 does not produce a valid congruence since $14/2 = 7$ and $8/2 = 4$, but $7 \not\equiv 4 \pmod{6}$.

STRUCTURES

- Cryptography requires sets of integers and specific operations that are defined for those sets. The combination of the set and the operations that are applied to the elements of the set is called an algebraic structure. In this chapter, we will define three common algebraic structures: groups, rings, and fields.

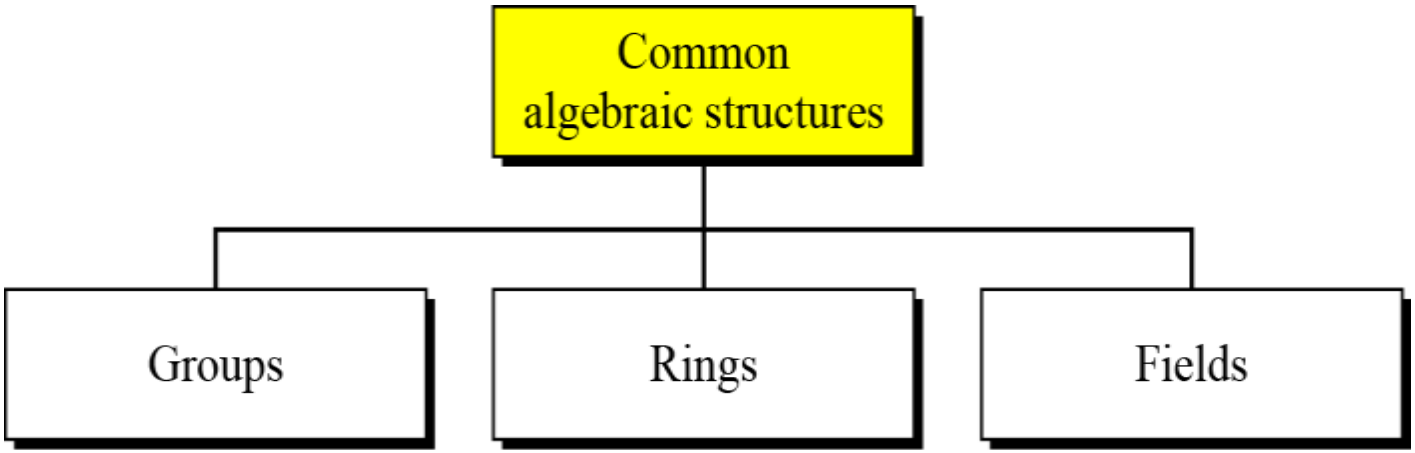


Figure 2.1 : Algebraic Structures

OU
PS

A group (G) is a set of elements with a binary operation (•) that satisfies four properties (or axioms). A commutative group satisfies an extra property, commutativity:

- Closure: If a and b are elements of G, then c=a•b is also an element of G.
- Associativity: a•(b•c)=(a•b)•c
- Commutativity: a•b=b•a
- Existence of identity: For all element a in G, there exists and element e, identity element, s. t. e•a=a•e=a
- Existence of inverse: for each a there exists an a', inverse of a, s.t. a•a'=a'•a=e.

GROUPS

Application

- Although a group involves a single operation, the properties imposed on the operation allow the use of a pair of operations as long as they are inverses of each other.

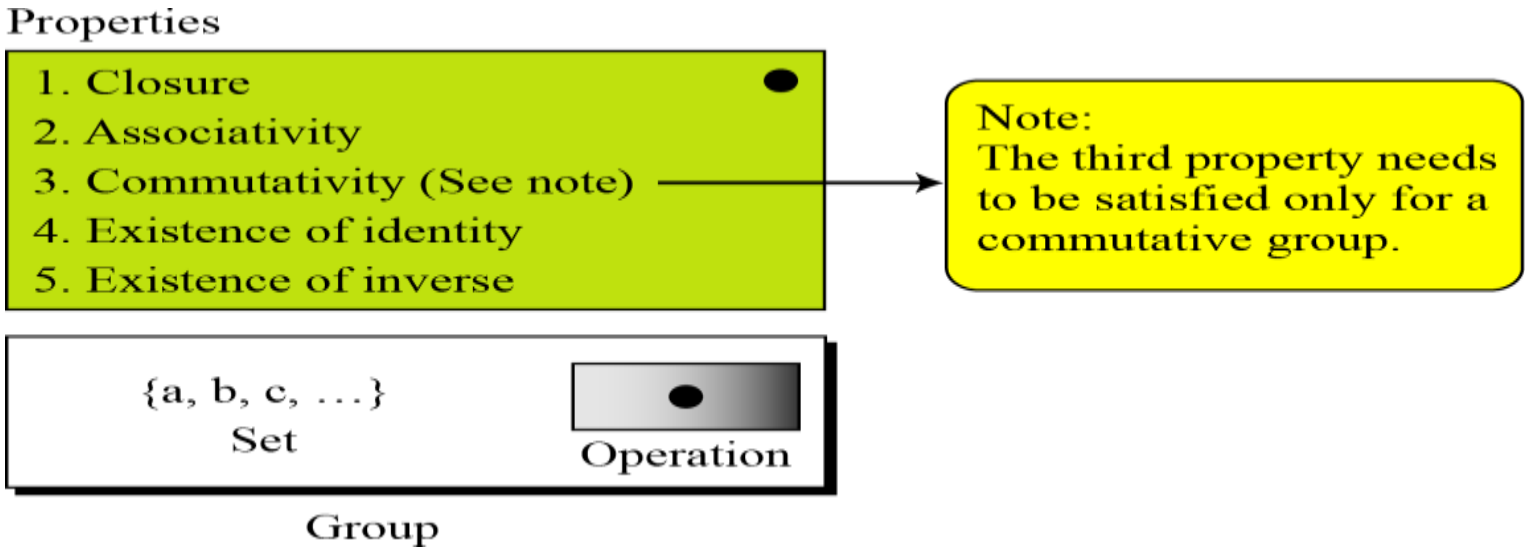


Figure 2.2 :Groups

N

G

- A ring, R = <{…}, •, >, is an algebraic structure with two operations.

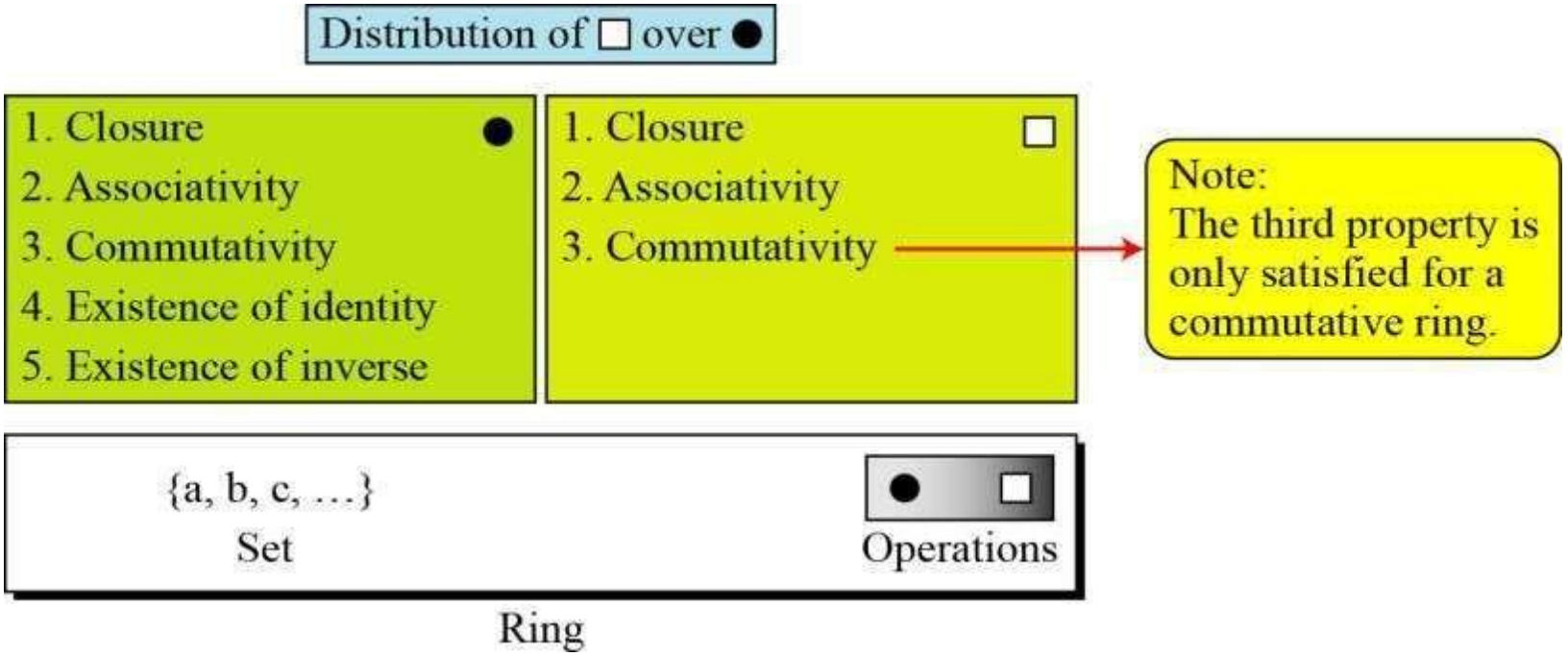


Figure 2.3 : Ring

N

G

- A ring, R = <{…}, •, , is an algebraic structure with two operations. The second operation must be distributed over the s>econd.

- Distributivity: for all a, b, and c elements of R, we have

$$a\ (b \bullet c) = (a\ \ b) \bullet (a\ c) \quad \text{and} \quad (a \bullet b)\ c = (a\ \ c) \bullet (b\ \ c)$$

Ring

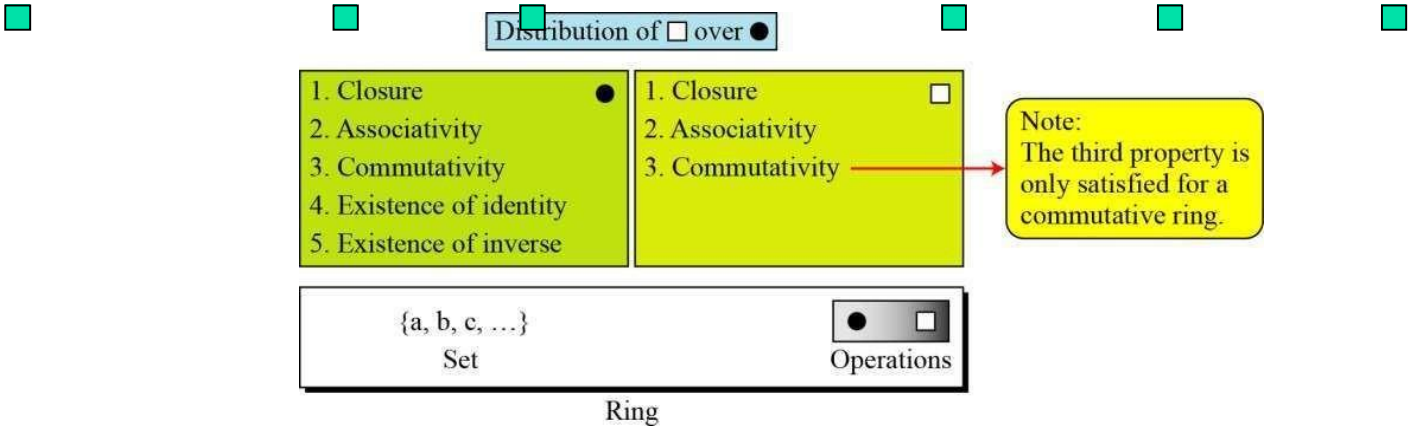


Figure 2.4 : Ring Distributivity

EL D

- A field, denoted by F = <{…}, •, > is a commutative ring in which the second operation satisfies all five properties defined for the first operation except that the identity of the first operation has no inverse.

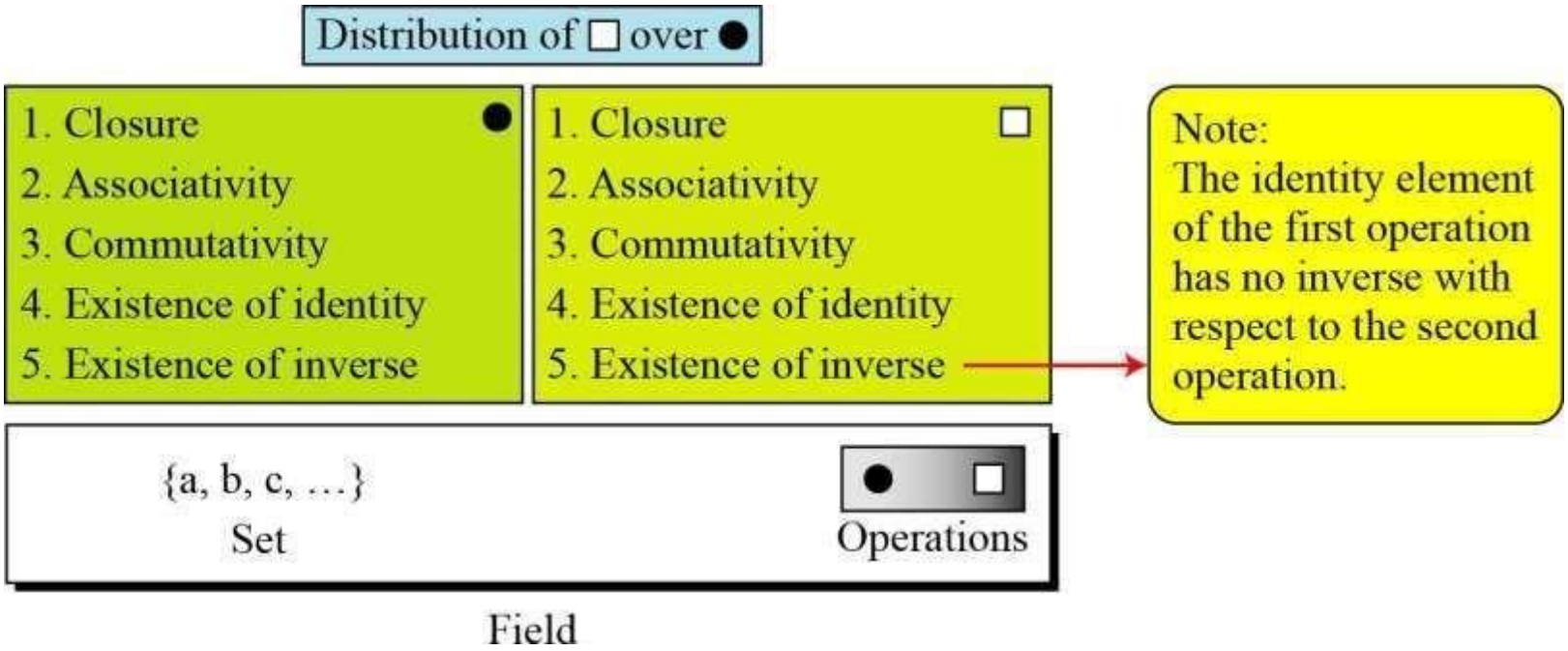


Figure 2.5 : Field

GF(2N) FIELDS

Exponentiation in GF(p)

- many encryption algorithms use exponentiation - raising a number a (base) to some power b (exponent) mod p

$b = a^e \bmod p$

- exponentiation is basically repeated multiplication, which take s O(n) multiples for a number n a better method is the square and multiply algorithm

let base = a, result =1

for each bit ei (LSB to MSB) of exponent

if ei=0 then square base mod p

if ei=1 then

multiply result by base mod p

square base mod p (except for MSB)

required ae is result

only takes $O(\log_2 n)$ multiples for a number n

GF(2N) FIELDS

Discrete Logarithms in GF(p)

- The inverse problem to exponentiation is that of finding the **discrete logarithm** of a number modulo p
  - find x where $a^x = b \bmod p$
- whilst exponentiation is relatively easy, finding discrete logarithms is generally a **hard** problem, with no easy way
- in this problem, we can show that if p is prime, then there always exists an a such that there is always a discrete logarithm for any b!=0
  - successive powers of a "generate" the group mod p
- such an a is called a **primitive root** and these are also relatively hard to find

MES

- **Definition**: A positive integer p greater than 1 is called *prime* if the only positive factors of p are 1 and p. A positive integer that is greater than 1 and is not prime is called *composite*.

Example: The integer 7 is prime because its only positive factors are 1 and 7, but 9 is composite because it is divisible by 3.

The Fundamental Theorem of Arithmetic

- **Theorem**: Every positive integer greater than 1 can be written uniquely as a prime or as the product of two or more primes where the prime factors are written in order of nondecreasing size.

Examples:

$100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$

$641 = 641$

$999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$

$1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$

MES

- Asymmetric-key cryptography uses primes extensively. The topic of primes is a large part of any book on number theory.

Topics discussed in this section:

1. Definition
2. Cardinality of Primes
3. Checking for Primeness
4. Euler's Phi-Function
5. Fermat's Little Theorem
6. Euler's Theorem
7. Generating Primes

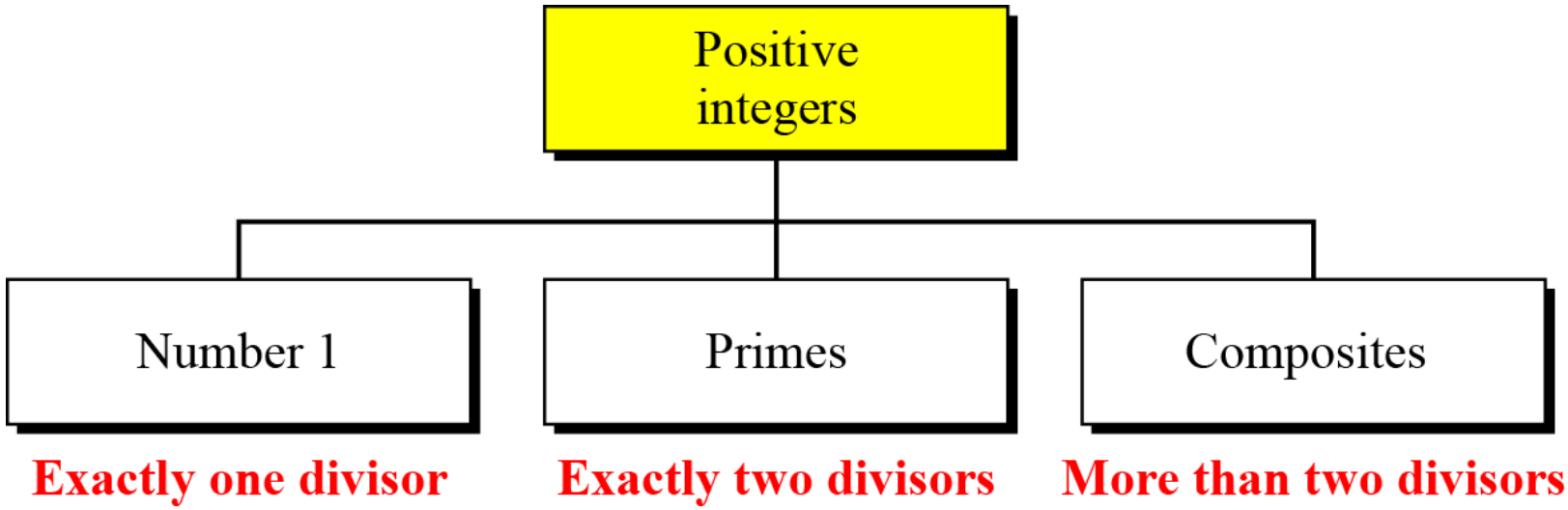


Figure 2. 6 : Three Groups of Positive integers

Note

A prime is divisible only by itself and 1.

INUE
D

Example 1

What is the smallest prime?

Solution

The smallest prime is 2, which is divisible by 2 (itself) and 1.

Example 2

List the primes smaller than 10.

Solution

There are four primes less than 10: 2, 3, 5, and 7. It is interesting to note that the percentage of primes in the range 1 to 10 is 40%. The percentage decreases as the range increases.

CARDINALITY OF PRIMES

Infinite Number of Primes

Note

There is an infinite number of primes.

Number of Primes

$$[n / (\ln n)] < \pi(n) < [n/(\ln n - 1.08366)]$$

INUE

D

Example 1

As a trivial example, assume that the only primes are in the set {2, 3, 5, 7, 11, 13, 17}. Here P = 510510 and P + 1 = 510511. However, 510511 = 19 × 97 × 277; none of these primes were in the original list. Therefore, there are three primes greater than 17.

Example 2

Find the number of primes less than 1,000,000.

Solution

The approximation gives the range 72,383 to 78,543. The actual number of primes is 78,498.

FOR PRIMENESS

Given a number n, how can we determine if n is a prime? The answer is that we need to see if the number is divisible by all primes less than

$$\sqrt{n}$$

We know that this method is inefficient, but it is a good start.

Theorem

If n is composite, then n has a prime divisor less than or equal to $\sqrt{n}$.

Proof.

- Let $n = ab$, $1 < a < n$, $1 < b < n$.
- We can't have both $a > \sqrt{n}$ and $b > \sqrt{n}$ since this would lead to $ab > n$.
- Therefore, n must have a prime divisor less than or equal to $\sqrt{n}$.

INUE

D

Example 1

Is 97 a prime?

Solution

The floor of 97 = 9. The primes less than 9 are 2, 3, 5, and 7. We need to see if 97 is divisible by any of these numbers. It is not, so 97 is a prime.

Example 2

Is 301 a prime?

Solution

The floor of Ö301 = 17. We need to check 2, 3, 5, 7, 11, 13, and 17. The numbers 2, 3, and 5 do not divide 301, but 7 does. Therefore 301 is not a prime.

4 EULER'S PHI-FUNCTION

Euler's phi-function, f (n), which is sometimes called the

Euler's totient function plays a very important role in cryptography**.**

1. $\phi(1) = 0.$
2. $\phi(p) = p - 1$ if p is a prime.
3. $\phi(m \times n) = \phi(m) \times \phi(n)$ if m and n are relatively prime.
4. $\phi(p^e) = p^e - p^{e-1}$ if p is a prime.

We can combine the above four rules to find the value of f(n). For example, if n can be factored as

$$n = p_1^{e1} \times p_2^{e2} \times \ldots \times p_k^{ek}$$

then we combine the third and the fourth rule to find

$$\phi(n) = (p_1^{e_1} - p_1^{e_1-1}) \times (p_2^{e_2} - p_2^{e_2-1}) \times \cdots \times (p_k^{e_k} - p_k^{e_k-1})$$

CONTINUED

Example 1

What is the value of f(13)?

Solution

Because 13 is a prime, f(13) = (13 −1) = 12.

Example 2

What is the value of f(10)?

Solution

We can use the third rule: f(10) = f(2) × f(5) = 1 × 4 = 4, because 2 and 5 are primes.

LITTLE THEOREM

First Version

$$a^{p-1} \equiv 1 \bmod p$$

Second Version

$$a^{p} \equiv a \bmod p$$

Example

Find the result of $6^{10} \bmod 11$.

Solution

We have 610 mod 11 = 1. This is the first version of Fermat's little theorem where p = 11.

INUE

Multiplicative Inverses

$$a^{-1} \bmod p = a^{p-2} \bmod p$$

Example

The answers to multiplicative inverses modulo a prime can be found without using the extended Euclidean algorithm:

a. $8^{-1} \bmod 17 = 8^{17-2} \bmod 17 = 8^{15} \bmod 17 = 15 \bmod 17$

b. $5^{-1} \bmod 23 = 5^{23-2} \bmod 23 = 5^{21} \bmod 23 = 14 \bmod 23$

c. $60^{-1} \bmod 101 = 60^{101-2} \bmod 101 = 60^{99} \bmod 101 = 32 \bmod 101$

d. $22^{-1} \bmod 211 = 22^{211-2} \bmod 211 = 22^{209} \bmod 211 = 48 \bmod 211$

6 EULER'S THEOREM

First Version

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

Second Version

$$a^{k \times \phi(n) + 1} \equiv a \pmod{n}$$

Example

Find the result of 6^{24} mod 35.

Solution

We have 6^{24} mod 35 = $6^{f(35)}$ mod 35 = 1.

INUE D

Multiplicative Inverses

Euler's theorem can be used to find multiplicative inverses modulo a composite.

$$a^{-1} \bmod n = a^{f(n)-1} \bmod n$$

Example

The answers to multiplicative inverses modulo a composite can be found without using the extended Euclidean algorithm if we know the factorization of the composite:

a. $8^{-1} \bmod 77 = 8^{\phi(77)-1} \bmod 77 = 8^{59} \bmod 77 = 29 \bmod 77$

b. $7^{-1} \bmod 15 = 7^{\phi(15)-1} \bmod 15 = 7^{7} \bmod 15 = 13 \bmod 15$

c. $60^{-1} \bmod 187 = 60^{\phi(187)-1} \bmod 187 = 60^{159} \bmod 187 = 53 \bmod 187$

d. $71^{-1} \bmod 100 = 71^{\phi(100)-1} \bmod 100 = 71^{39} \bmod 100 = 31 \bmod 100$

GENERATIN G PRIMES

Mersenne Primes

$$M_p = 2^p - 1$$

$M_2 = 2^2 - 1 = 3$
$M_3 = 2^3 - 1 = 7$
$M_5 = 2^5 - 1 = 31$
$M_7 = 2^7 - 1 = 127$
$M_{11} = 2^{11} - 1 = 2047$ **Not a prime ($2047 = 23 \times 89$)**
$M_{13} = 2^{13} - 1 = 8191$
$M_{17} = 2^{17} - 1 = 131071$

Fermat Primes

$$F_n = 2^{2^n} + 1$$

$F_0 = 3$ $F_1 = 5$ $F_2 = 17$ $F_3 = 257$ $F_4 = 65537$
$F_5 = 4294967297 = 641 \times 6700417$ Not a prime

PRIMALITY TESTING

- Finding an algorithm to correctly and efficiently test a very large integer and output a prime or a composite has always been a challenge in number theory, and consequently in cryptography. However, recent developments look very promising.

Topics discussed in this section:

1. Deterministic Algorithms
2. Probabilistic Algorithms
3. Recommended Primality Test

FACTOR IZATION

- Factorization has been the subject of continuous research in the past; such research is likely to continue in the future

Topics discussed in this section:

1 Fundamental Theorem of Arithmetic

2. Factorization Methods

3. Fermat Method

4. Pollard $p - 1$ Method

5. Pollard rho Method

6. More Efficient Methods

REMAINDER THEOREM

- The Chinese remainder theorem (CRT) is used to solve a set of congruent equations with one variable but different moduli, which are relatively prime, as shown below:

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\cdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

Example

The following is an example of a set of equations with different moduli:

$$\begin{aligned} x &\equiv 2 \pmod 3 \\ x &\equiv 3 \pmod 5 \\ x &\equiv 2 \pmod 7 \end{aligned}$$

- The solution to this set of equations is given in the next section; for the moment, note that the answer to this set of equations is x = 23. This value satisfies all equations: 23 ≡ 2 (mod 3), 23 ≡ 3 (mod 5), and 23 ≡ 2 (mod 7).

CONGRUENCE

- In cryptography, we also need to discuss quadratic congruence¾that is, equations of the form $a_2x^2 + a_1x + a_0 \equiv 0 \pmod{n}$. We limit our discussion to quadratic equations in which $a_2 = 1$ and $a_1 = 0$, that is equations of the form

$$x^2 \equiv a \pmod{n}.$$

Topics discussed in this section:

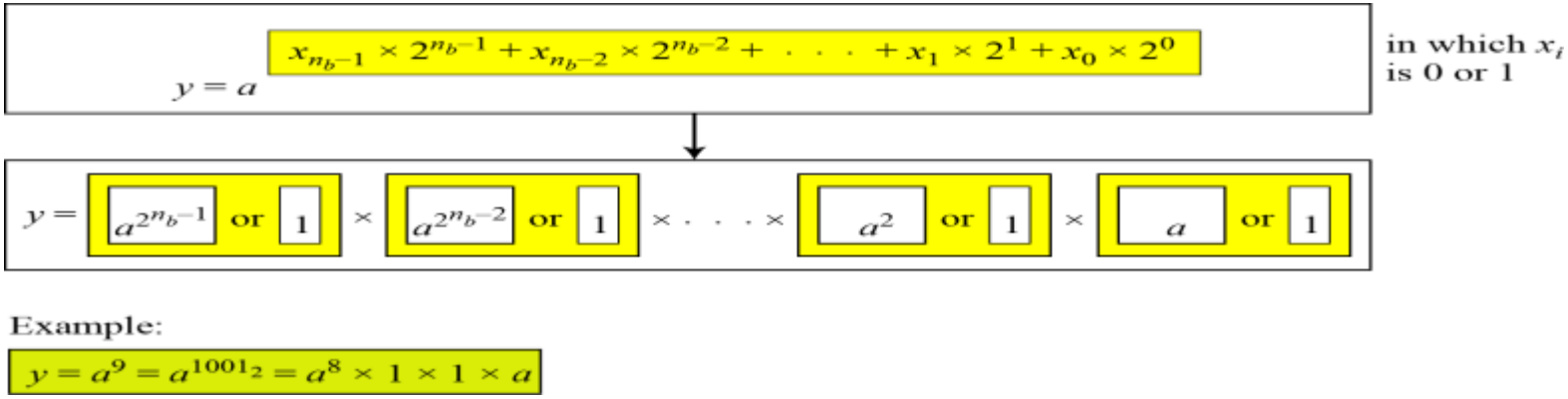
1. Quadratic Congruence Modulo a Prime
2. Quadratic Congruence Modulo a Composite

EXPONENTIATION AND LOGARTHM

Exponentiation: $y = a^x \quad \rightarrow \quad$ **Logarithm:** $x = \log_a y$

$$x^2 \equiv a \pmod{n}.$$

1 Exponentiation



Example:

$y = a^9 = a^{1001_2} = a^8 \times 1 \times 1 \times a$

Figure 2.13 : The idea behind Square and Multiply Method

LOGARITHM

- In cryptography, we also need to discuss modular logarithm•

Exhaustive Search

```
Modular_Logarithm (a, y, n)
{
    for (x = 1 to n −1)                          // k is the number of bits in x
    {
        if (y ≡ a^x mod n) return x
    }
    return failure
}
```

Figure 2.14 : Exhaustive search for modular logarithm

LOGARITHM

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1
6	17	7	4	5	11	9	16	1	6	17	7	4	5	11	9	16	1
7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	7	11	1
8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1
9	5	7	6	16	11	4	17	1	9	5	7	6	16	11	4	17	1
10	5	12	6	3	11	15	17	18	9	14	7	13	16	8	4	2	1
11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	11	7	1
12	11	18	7	8	1	12	11	18	7	8	1	12	11	18	7	8	1
13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1
14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1
15	16	12	9	2	11	13	5	18	4	3	7	10	17	8	6	14	1
16	9	11	5	4	7	17	6	1	16	9	11	5	4	7	17	6	1
17	4	11	16	6	7	5	9	1	17	4	11	16	6	7	5	9	1
18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1

Table 2.1 : Powers of Integers, Modulo 19

ON SECURITY

UNIT-II

CHAPTER-2

OUTLINE

- **Principles of public-key cryptography**
- **RSA Algorithm**
- **Diffie-Hellman Key Exchange**
- **ELGamal cryptographic system**
- **Elliptic Curve Arithmetic**
- **Elliptic curve cryptography**

CRYPTOGRAPHY

public-key (or **two-key**) **cryptography** involves the use of two keys:

- a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
- a **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**

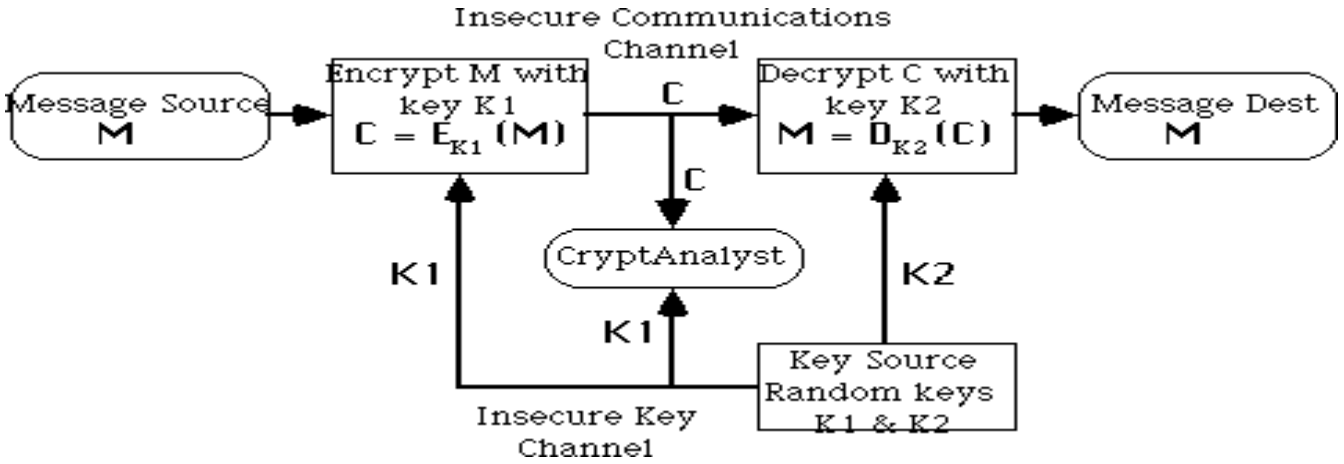


Figure 2.15 : Asymmetric Encryption System

PUBLIC-KEY CRYPTOGRAPHY

- The public-key is easily computed from the private key and other information about the cipher (a polynomial time (p-time) problem)
- However, knowing the public-key and public description of the cipher, it is still computationally infeasible to compute the private key (an np-time problem)
- Thus the public-key may be distributed to anyone wishing to communicate securely with its owner (although secure distribution of the public-key is a non-trivial problem - the **key distribution** problem)

PUBLIC-KEY CRYPTOGRAPHY

- Have three important classes of public-key algorithms:

Public-Key Distribution Schemes (PKDS) - where the scheme is used to securely exchange a single piece of information (whose value depends on the two parties, but cannot be set).

- This value is normally used as a session key for a private-key scheme
- **Signature Schemes** - used to create a digital signature only, where the private-key signs (create) signatures, and the public-key verifies signatures
- **Public Key Schemes (PKS)** - used for encryption, where the public-key encrypts messages, and the private-key decrypts messages.

DIGITAL SIGNATURES AND SIGNATURE SCHEMES

- Where the private-key signs (create) signatures, and the public-key verifies signatures
- Only the owner can create the digital signature, hence it can be used to verify who created a message
- Generally don't sign the whole message (doubling the size of information exchanged), but just a **digest** or **hash** of the message,
- A **hash function** takes the message, and produces a fixed size (typically 64 to 512 bits) value dependent on the message
- It must be hard to create another message with the same hash value (otherwise some forgeries are possible)
- Developing good hash functions is another non-trivial problem

RSA ALGORITHM

- Best known and widely regarded as most practical public-key scheme was proposed by rivest, shamir & adleman in 1977:
- R L rivest, A shamir, L adleman, "on digital signatures and public key cryptosystems", communications of the ACM, vol 21 no 2, pp120-126, feb 1978
- It is a public-key scheme which may be used for encrypting messages, exchanging keys, and creating digital signatures
- Is based on exponentiation in a finite (galois) field over integers modulo a prime

  Nb exponentiation takes $o((\log n)^3)$ operations

ALGORITHM

- Its security relies on the difficulty of calculating factors of large numbers

  Nb factorization takes $o(e^{\log n \log \log n})$ operations

  (same as for discrete logarithms)

- The algorithm is patented in north America (although algorithms cannot be patented elsewhere in the world)

  This is a source of legal difficulties in using the scheme

  RSA is a public key encryption algorithm based on exponentiation using modular arithmetic

ALGORITHM

- To use the scheme, first generate keys:
- Key-generation by each user consists of:
  - Selecting two large primes at random (~100 digit), p, q
  - Calculating the system modulus r=p.Q p, q primes
  - Selecting at random the encryption key e,
  - E < R, gcd(e, F(R)) = 1
  - Solving the congruence to find the decryption key d,
  - E.D [[equivalence]] 1 mod [[phi]](r) 0 <= d <= R
  - Publishing the public encryption key: k1={e,r}
  - Securing the private decryption key: k2={d,p,q}
- Encryption of a message M to obtain cipher text C is:

$C = m^e \bmod R$ 0 <= d <= R

ALGORITHM

- Decryption of a cipher text C to recover the message M is:

$M = c^d = m^{e.D} = m^{1+n.[[Phi]](r)} = M \bmod R$

- The RSA system is based on the following result:

If R = pq where p, q are distinct large primes then

X [[phi]](r) = 1 mod R

For all x not divisible by p or q

And [[phi]](r) = (p-1)(q-1)

EXAM PLE

- Usually the encryption key e is a small number, which must be relatively prime to [[phi]](r) (i.e. gcd(e, [[phi]](r)) = 1)
- Typically e may be the same for all users (provided certain precautions are taken), 3 is suggested
- The decryption key d is found by solving the congruence: e.D [[equivalence]] 1 mod [[phi]](r), 0 <= d <= R,
- An extended Euclid's GCD or binary GCD calculation is done to do this.

EXAM PLE

- Given e=3, r=11*47=517, [[phi]](r)=10*46=460 then d=inverse(3,460) by Euclid's alg:

I	y	g	u	v
0	-	460	1	0
1	-	3	0	1
2	153	1	1	-153
3	3	0	-3	460

I.e.: d = -153, or 307 mod 517

- A sample RSA encryption/decryption calculation is:

  M = 26

  C = 263 mod 517 = 515

  M = 515307 mod 517 = 26

SECURITY OF RSA

- The security of the RSA scheme rests on the difficulty of factoring the modulus of the scheme R Best known factorization algorithm (Brent-pollard) takes:
- Operations on number R whose largest prime factor is p

Decimal digits in R	#bit operations to factor R
20	7200
40	3.11e+06
60	4.63e+08
80	3.72e+10
100	1.97e+12
120	7.69e+13
140	2.35e+15
160	5.92e+16
180	1.26e+18
200	2.36e+19

HELLMAN KEY EXCHANGE

- First public-key type scheme proposed was a PKDS by diffie & hellman in 1976:
- W diffie, M E hellman, "new directions in cryptography", IEEE trans. Information theory, IT-22, pp644-654, nov 1976
- An excellent overview of cryptography at this time is:
- W diffie, M E hellman, "privacy and authentication: an introduction to cryptography", proc. Of the IEEE, vol 67 no 3, pp 397-427, mar 1979
- It is a public-key distribution scheme
  - It cannot be used to exchange an arbitrary message Only a key, whose value depends on the participants (and their private and public key information)
- Is based on exponentiation in a finite (galois) field, either over integers modulo a prime, or a polynomial field
  - Nb exponentiation takes $o((\log n)^3)$ operations
- Its security relies on the difficulty of computing logarithms in these fields
  - Nb discrete logarithms takes $o(e^{\log n \log \log n})$ operations

HELLMAN KEY EXCHANGE

- Diffie-Hellman PKDS works as follows:
  - Two people A & B, who wish to exchange some key over an insecure communications channel can:
  - Select a large prime p (~200 digit), and
  - [[alpha]] a primitive element mod p
  - A has a secret number x_a
  - B has a secret number x_b
  - A and B compute y_a and y_b respectively, which are then made public

    Y_a = [[alpha]]^{xa} mod p y_b = [[alpha]]^{xb} mod p

HELLMAN KEY EXCHANGE

- The key is then calculated as

$$K_{AB} = [[alpha]]^{xa.Xb} \bmod p$$

$$= y_a^{xb} \bmod p \text{ (which B can compute)}$$

$$= y_b^{xa} \bmod p \text{ (which A can compute)}$$

And may then be used in a private-key cipher to secure communications between A and B

Nb: if the same two people subsequently wish to communicate, they will have the **same** key as before, unless they change their public-key (usually not often)

CRYPTOGRAPHIC SYSTE

- A variant of the Diffie-Hellman key distribution scheme, allowing secure exchange of messages Published in 1985 by Elgamal in T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE trans. Information theory, vol IT-31(4), pp469-472, July 1985.
- Like Diffie-Hellman its security depends on the difficulty of factoring logarithms

Key generation

Select a large prime p (~200 digit), and

[[alpha]] a primitive element mod p

A has a secret number x_a

B has a secret number x_b

A and b compute y_a and y_b respectively, which are then made public

$$\mathbf{Y_a = [[alpha]]^{xa} \bmod p}$$

$$\mathbf{Y_b = [[alpha]]^{xb} \bmod p}$$

CRYPTOGRAPHIC SYSTEM

Encryption:

- To **encrypt** a message **M** into cipher text **C**,
  - Selects a random number **k,** 0 <= k <= p-1
  - Computes the message key **K**

    $K = y_b^k \bmod p$
  - Computes the cipher text pair: C = {c1,c2}

    $C_1 = [[alpha]]^k \bmod p$ $C_2 = K.M \bmod p$

Decryption:

To **decrypt** the message

- Extracts the message key **K**

  $K = c_1^{xb} \bmod p = [[alpha]]^{k.Xb} \bmod p$
- Extracts **M** by solving for M in the following equation:

  $C_2 = K.M \bmod p$

CURVESCRYPTO GRAPHY

- Let K be a field.

- An **elliptic curve** E over K is defined by the **Weierstrass equation**:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,\ a_i \in K.$$

The curve should be **smooth** (no singularities).

Special forms

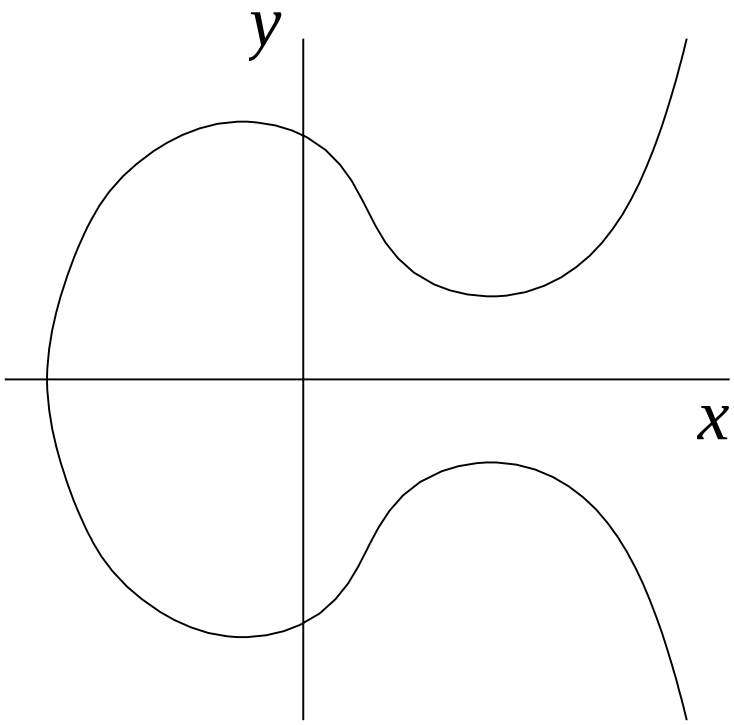
char$K$ ƒ= 2,3: $y^2 = x^3 + ax + b,\ a,b \in K$.

char$K$ = 3: $y^2 = x^3 + b_2x^2 + b_4x + b_6,\ b_i \in K$. char K = 2:

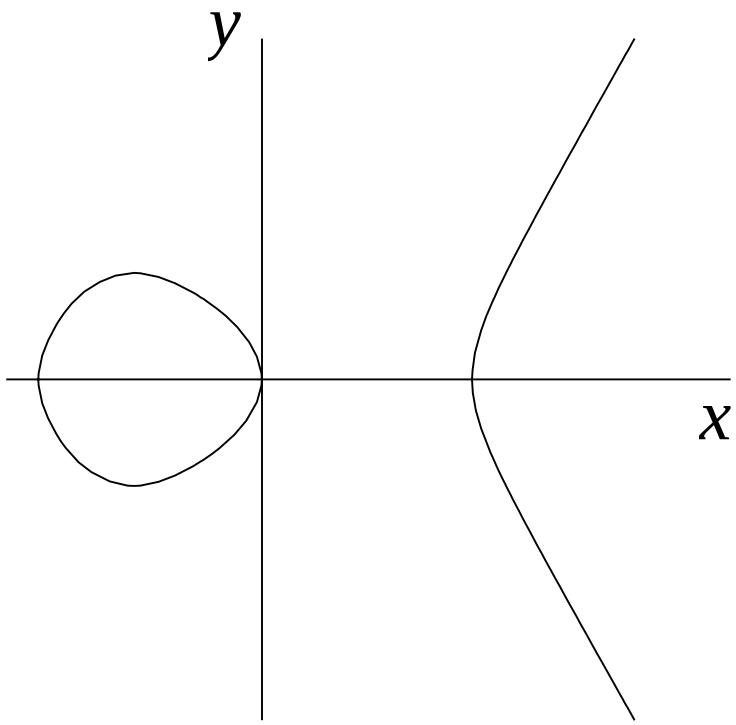
Non-supersingular or ordinary curve: $y^2 + xy = x^3 + ax^2 + b,\ a,b \in K$.

Supersingular curve: $y^2 + ay = x^3 + bx + c,\ a,b,c \in K$.

CURVES: EXAMPLE



(a) $y^2 = x^3 - x + 1$



(b) $y^2 = x^3 - x$

Figure 2.16 : Real Elliptic Curves Example

CURVE GROUP

- Any $(x, y) \in K^2$ satisfying the equation of an elliptic curve E is calleda
- K-**rational point** on E.

- **Point at infinity:**
- There is a single point at infinity on E, denoted by O.
- This point cannot be visualized in the two-dimensional (x, y)plane. The point exists in the projective plane.
- $E(K)$ is the set of all finite K-rational points on E and the point at infinity. An additive group structure can be defined on $E(K)$.

- O acts as the identity of the group.

THE OPPOSITE OF A POINT

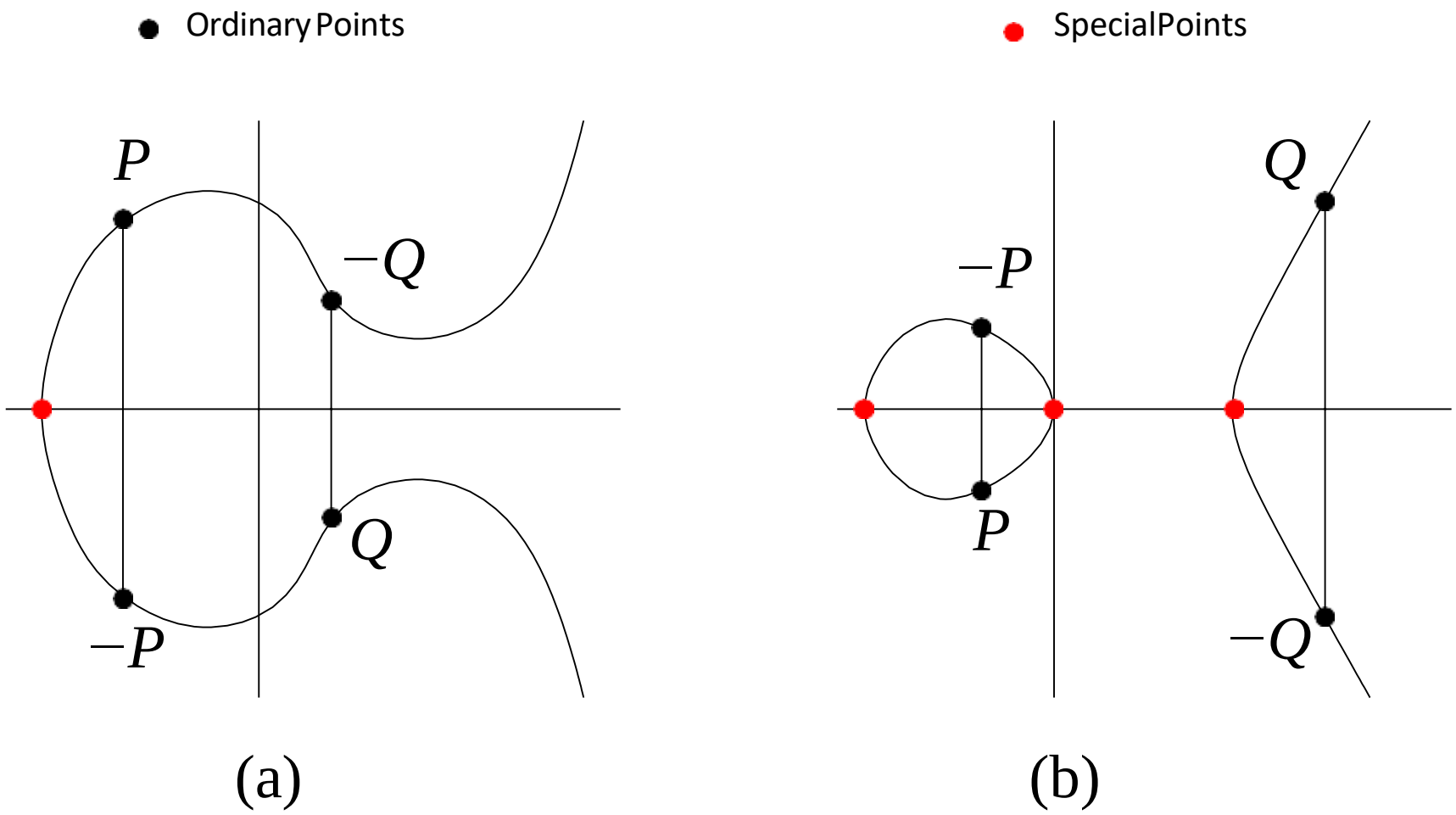


Figure 2.17 : The opposite of a Point

CURVE ARITHMETIC

- **Addition and Doubling Formulas**

Let $P=(h_1,k_1)$ and $Q=(h_2,k_2)$ be finite points. Assume that P $+Q \neq O$ and $2P \neq O$.

Let $P+Q=(h_3,k_3)$ (Note that $P+Q=2P$ if $P=Q$).

$E : y^2 = x^3 + ax + b$

$$-P = (h_1, -k_1)$$

$$h_3 = \lambda^2 - h_1 - h_2$$

$$k_3 = \lambda(h_1 - h_3) - k_1, \text{ where}$$

$$\lambda = \begin{cases} \dfrac{k_2 - k_1}{h_2 - h_1}, & \text{if } P \neq Q, \\ \dfrac{3h_1^2 + a}{2k_1}, & \text{if } P = Q. \end{cases}$$

ELLIPTIC-CURVE ARITHMETIC

$E : y^2 = x^3 - 5x + 1$ defined over F_{17}.

Take the finite points $P = (3,8)$ and $Q = (10,13)$ on E.

Opposite: $-P = (3,9)$, and $-Q = (10,4)$.

Point addition

The line L joining P and Q has slope $? \equiv \frac{13-8}{10-3} \equiv 8 \pmod{17}$.

L has equation $L : y = 8x + c$. Since L passes through P, we have $c = 1$.

Substitute this in the equation for E to get $(8x+1)^2 \equiv x^3 - 5x + 1 \pmod{17}$, that is, $x^3 + 4x^2 + 13x \equiv 0 \pmod{17}$, that is, $x(x-3)(x-10) \equiv 0 \pmod{17}$.

The third point of intersection is $(0,1)$, so $P + Q = -(0,1) = (0,16)$.

ELLIPTIC-CURVE ARITHMETIC

Point doubling

The tangent T to E at P has slope $\frac{3\times 3 - 5^{\ 2}}{2\times 8} \equiv 12 \pmod{17}$.

The equation for T is $y = 12x + 6$.

Substitute T in E to get $x^3 + 9x^2 + 4x + 16 \equiv 0 \pmod{17}$, that is,

$(x-3)^2(x-2) \equiv 0 \pmod{17}$.

The third point of intersection is $(2,13)$, so $2P = -(2,13) = (2,4)$.

CONTENT BEYOND SYLLABUS

OUT LINE

- COMPLEXITY THEORY
- COMPLEXITY THEORY - SOME TERMINOLOGY
- OTHER PUBLIC-KEY SCHEMES
- CHECKSUM
- CHECKSUM APPLICATIONS

COMPLEXI TY THEORY

- Study of how hard a problem is to solve in general
- Allows classification of types of problems
- Some problems intrinsically harder than others, eg
  - Multiplying numbers o(n^(2))
  - Multiplying matrices o(n^(2)(2n-1))
  - Solving crossword o(26^(n))
  - Recognizing primes o(n^(log log n))
- Deal with worst case complexity
  - May on average be easier

COMPLEXITY THEORY - SOME TERMINOLOGY

- An **instance** of a problem is a particular case of a general problem
- The **input length** of a problem is the number **n** of symbols used to characterize a particular instance of it
- The **order** of a function **f(n)** - is some **o(g(n))** of some function **g(n)** s.T.
  - F(n)<=c.|G(n)|, for all n>=0, for some c
- A **polynomial time** algorithm (**P**) is one which solves any instance of a particular problem in a length of time o(p(n)), where p is some polynomial on input length
- An **exponential time** algorithm (**E**) - is one whose solution time is not so bounded

COMPLEXITY THEORY - SOME TERMINOLOGY

- A **non-deterministic polynomial time** algorithm (**NP**) - is one for which any guess at the solution of an instance of the problem may be checked for validity in polynomial time
- **Np-complete** problems - are a subclass of NP problems for which it is known that if any such problem has a polynomial time solution, then **all NP** problems have polynomial solutions. These are thus the **hardest NP** problems
- **Co-np** problems - are the complements of **NP** problems, to prove a guess at a solution of a co-np problem may well require an exhaustive search of the solution space

PUBLIC-KEY SCHEMEs

- A number of other public-key schemes have been proposed, some of the better known being:
  - ALL of these schemes have been broken
  - The only currently known secure public key schemes are those based on exponentiation
    - Knapsack based schemes
    - Mceleice's error correcting code based schems
  - (All of which are patented in north America)
  - It has proved to be very very difficult to develop secure public key schemes
  - This in part is why they have not been adopted faster, as their theorectical advantages might have suggested

KSUM

A checksum or hash sum is a fixed-size data computed from an arbitrary block of digital data for the purpose of detecting accidental errors that may have been introduced during its transmission or storage.

- The integrity of the data can be checked at any later time by recomputing the checksum and comparing it with the stored one.

- If the checksums do not match, the data was almost certainly altered.

APPLICATIONS

- First, checksum value can be used to check data integrity when data is sent through telecommunication networks such as Internet .
- Second, checksum value can be used to check data integrity of stored data to see if the data has been modified or changed in any way over time.
- Third, checksum values can be used to verify data burned to CDROM, CD-R (Compact Disc-Recordable), OR DVD, DVD-R.

RESOURCES

- **Lecture Notes - Lecture Notes**
- **Video Lectures - Video Lecture**
- **E-Book - Information Security Concepts**
- **Model Papers - JNTUA Question Papers**

SVEC TIRUPATI

EDUCATION FOR A BETTER SOCIETY

DEPT & SEM : CSE -CS& I SEM

SUBJECT NAME: INFORMATION SECURITY

COURSE CODE : IS

UNIT : III

PREPARED BY :ANUSHA K

OUTLINE

- **Cryptographic Hash functions**
- **Applications of Cryptographic Hash functions**
- **Requirements and security**
- **Hash functions based on Cipher Block Chaining Secure**
- **Hash Algorithm (SHA)**

CRYPTOGRAPHIC HASH FUNCTIONS

- Condenses arbitrary message to fixed size
  - H = H(M)
- Usually assume hash function is public
- Hash used to detect changes to message
- Want a cryptographic hash function

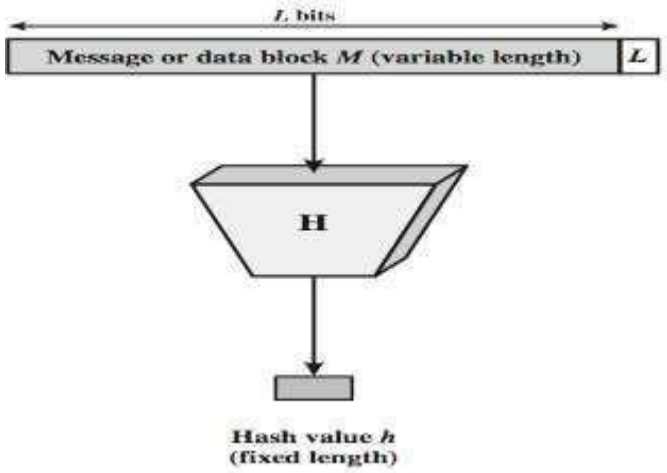


Figure 3.1: Hash Function

- Computationally infeasible to find data mapping to specific hash (one-way property)
- Computationally infeasible to find two data to same hash (collision-free property)

CRYPTOGRAPHIC HASH FUNCTIONS

- There security of hash functions is defined empirically, if the following problems are found to be computationally infeasible:
  - **One way:**

    Given y, find x such that h(x) = y
  - **Second pre-image resistant:**

    Given x, find y≠ x such that h(x) = h(y)
  - **Collision-resistant:**

    Find y, x, with y≠ x such that h(x) = h(y)

FUNCTIONS

- A hash function is a mathematical, efficiently computable function that has fixed size output:

  $F : \{0, 1\}^N \rightarrow \{0,1\}^n$, where $N > n$

  $F: \{0, 1\}^* \rightarrow \{0,1\}^n$

- In cryptography, the first type of hash function is often called a compression function, with the name hash function reserved for the unbounded domain type.

FUNCTION USES

- **Message Integrity Check (MIC)**

  send hash of message (digest)

  MIC always encrypted, message optionally

- **Message Authentication Code (MAC)**

  send keyed hash of message

  MAC, message optionally encrypted

- **Digital Signature (non-repudiation)**

  Encrypt hash with private (signing) key

  Verify with public (verification) key

CRYPTOGRAPHIC HASH FUNCTIONS

- System integrity protection.
- For password verification, eliminating the need to keep passwords
- As building blocks for **message authentication codes** (MACs) and **digital signature** algorithms.

REQUIREMENTS AND SECURITY

Requirement	Description
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical.
Preimage resistant (one-way property)	For any given hash value h, it is computationally infeasible to find y such that H(y) = h.
Second preimage resistant (weak collision resistant)	For any given block x, it is computationally infeasible to find y ! x with H(y) = H(x).
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x, y) such that H(x) = H(y).
Pseudorandomness	Output of H meets standard tests for pseudorandomness

Table 3.1: Requirements and Security

HASH FUNCTIONS

- Since constructing secure hash functions is a difficult problem, the following approach has been taken in practice:
- Construct a good compression function. Since the domain of compression functions are “small” they are easier to test for the desired properties.
- Use the MD construction (next) to turn a one-way, collision-resistant compression function into a hash function with similar properties.

DAMGARD (MD)

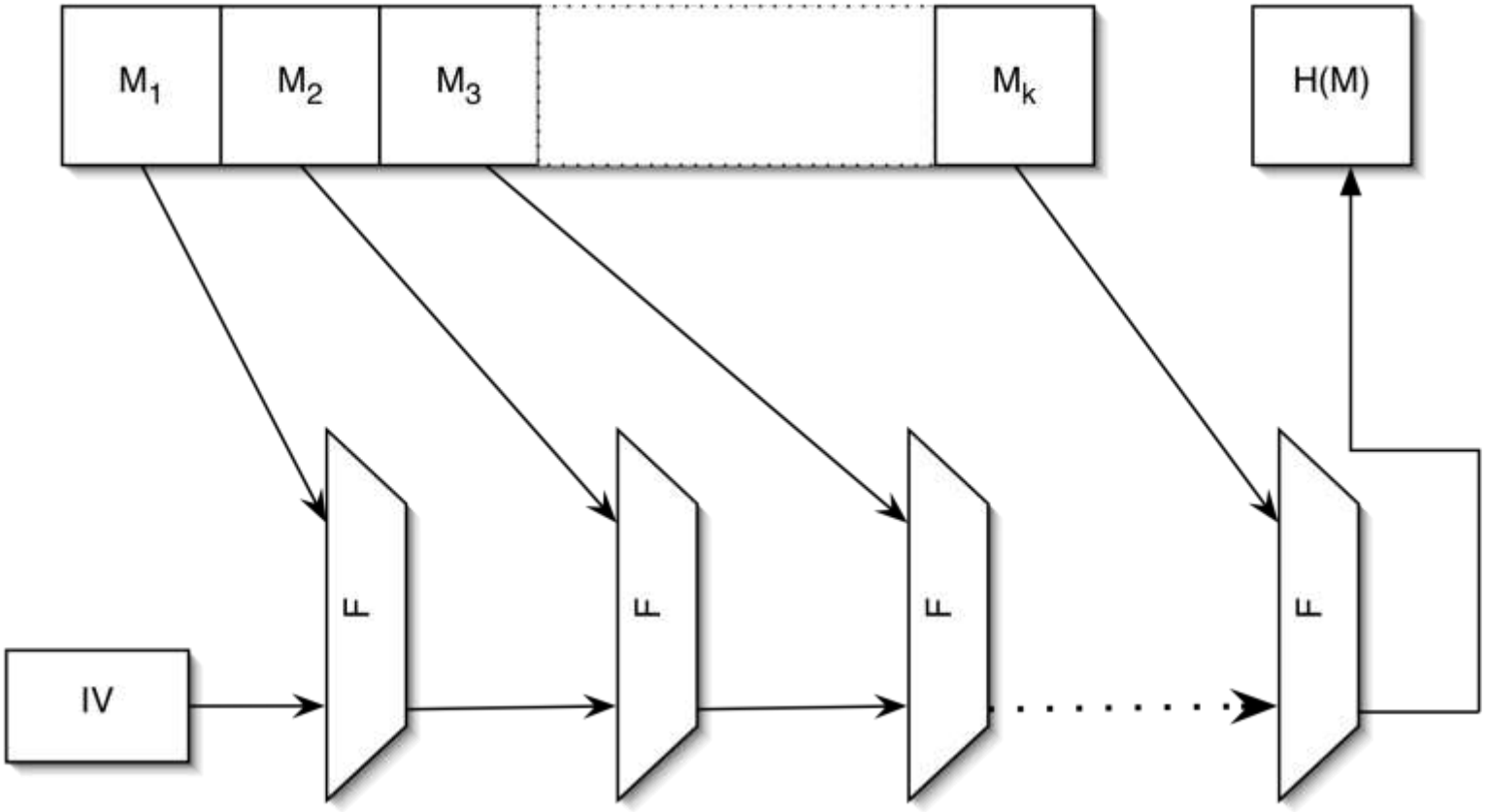


Figure 3.2: Markle-Damgard(MD)

HASH FUNCTIONS BASED ON CIPHER BLOCK CHAINING

- This can use block ciphers as hash functions

  Using H_0=0 and zero-pad of final block

  Compute: $h_i = e_{m_i} [h_{i-1}]$

  And use final block as the hash value

  Similar to CBC but without a key

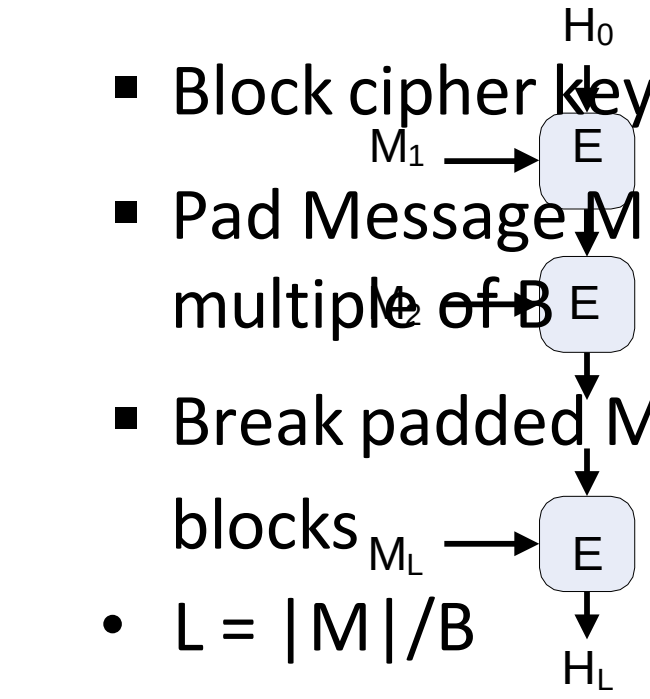
- Resulting hash is too small (64-bit)

  Both due to direct birthday attack

  And to “meet-in-the-middle” attack

- Other variants also susceptible to attack

ON CIPHER BLOCK CHAINING



- Block cipher key length B
- Pad Message M to multiple of B
- Break padded M into L blocks
- L = |M|/B
- $M = M_1, M_2, \ldots M_L$
- Use blocks of M as keys in block cipher, iteratively encrypt state value starting

Figure 3.3: Hash Function-Cipher Block Chain

with constant H resulting in hash

ALGORITHM (SHA)

- SHA originally designed by NIST & NSA in 1993
- was revised in 1995 as SHA-1
- US standard for use with DSA signature scheme

  standard is FIPS 180-1 1995, also Internet RFC3174

  nb. the algorithm is SHA, the standard is SHS

- based on design of MD4 with key differences
- produces 160-bit hash values
- 2005 results on security of SHA-1 raised concerns on its use in future applications

REVISED SECURE HASH STANDARD

- NIST issued revision FIPS 180-2 in 2002
- adds 3 additional versions of SHA

  SHA-256, SHA-384, SHA-512
- designed for compatibility with increased security provided by the AES cipher
- structure & detail is similar to SHA-1
- hence analysis should be similar
- but security levels are rather higher

VERSIONS

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message digest size	160	224	256	384	512
Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block size	512	512	512	1024	1024
Word size	32	32	32	64	64
Number of steps	80	64	64	80	80

Table 3.2: SHA Versions

SHA-512 OVERVIEW

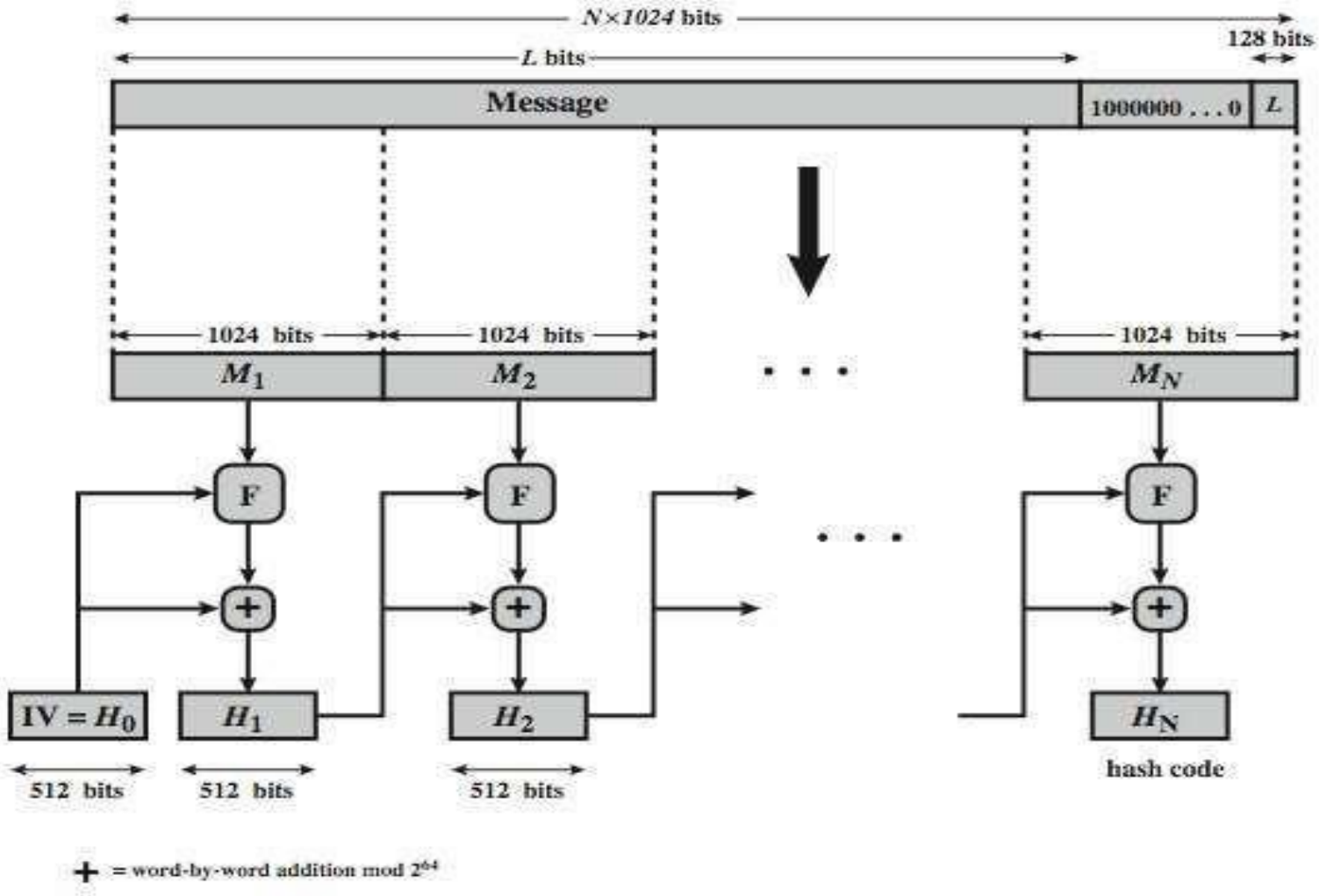


Figure 3.4: SHA-512

COMPRESSION FUNCTION

- Heart of the algorithm
- Processing message in 1024-bit blocks
- Consists of 80 rounds
  - Updating a 512-bit buffer
  - Using a 64-bit value wt derived from the current message block
  - And a round constant based on cube root of first 80 prime numbers

ROUND FUNCTION

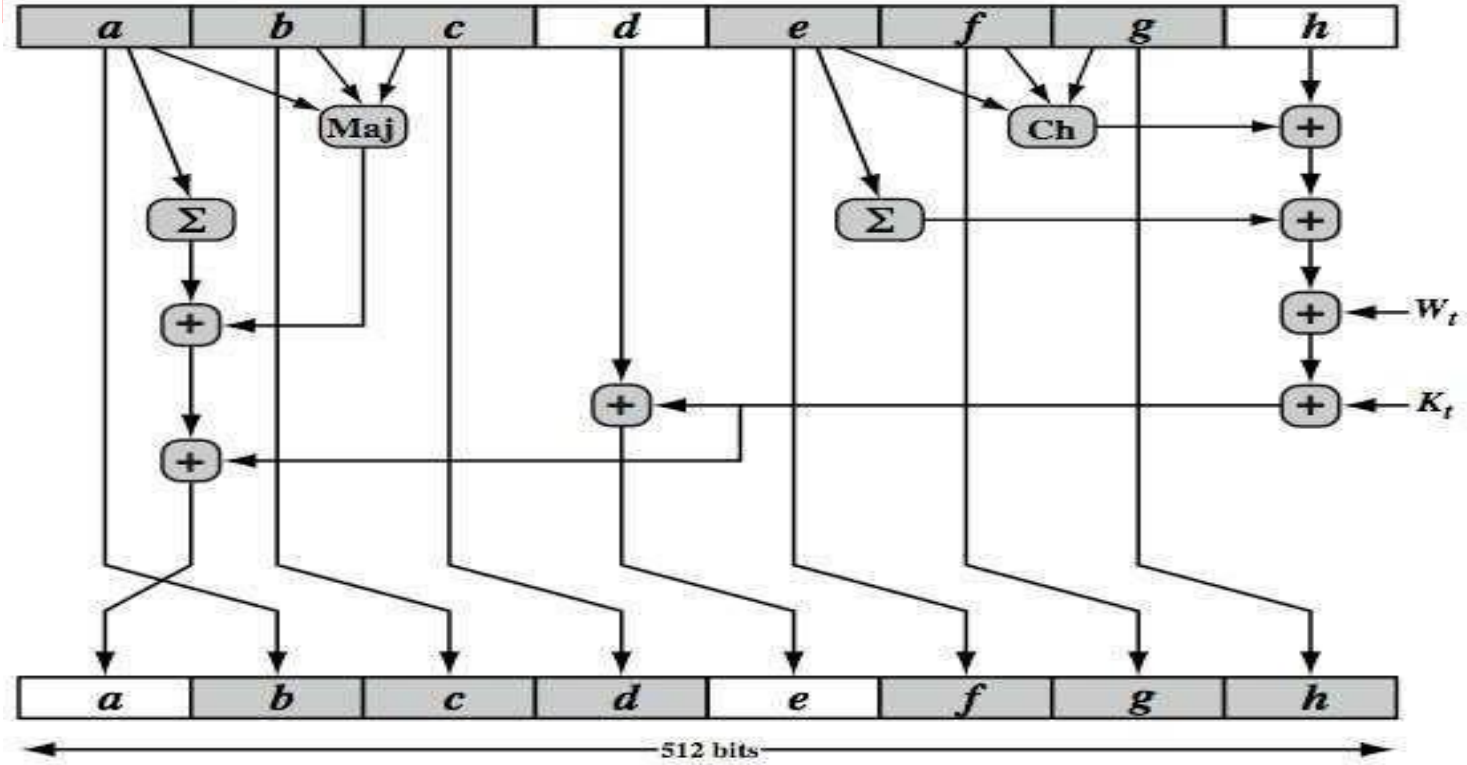


Figure 3.5 : SHA-512 Round Function Encryption

SHA-512
Round Function

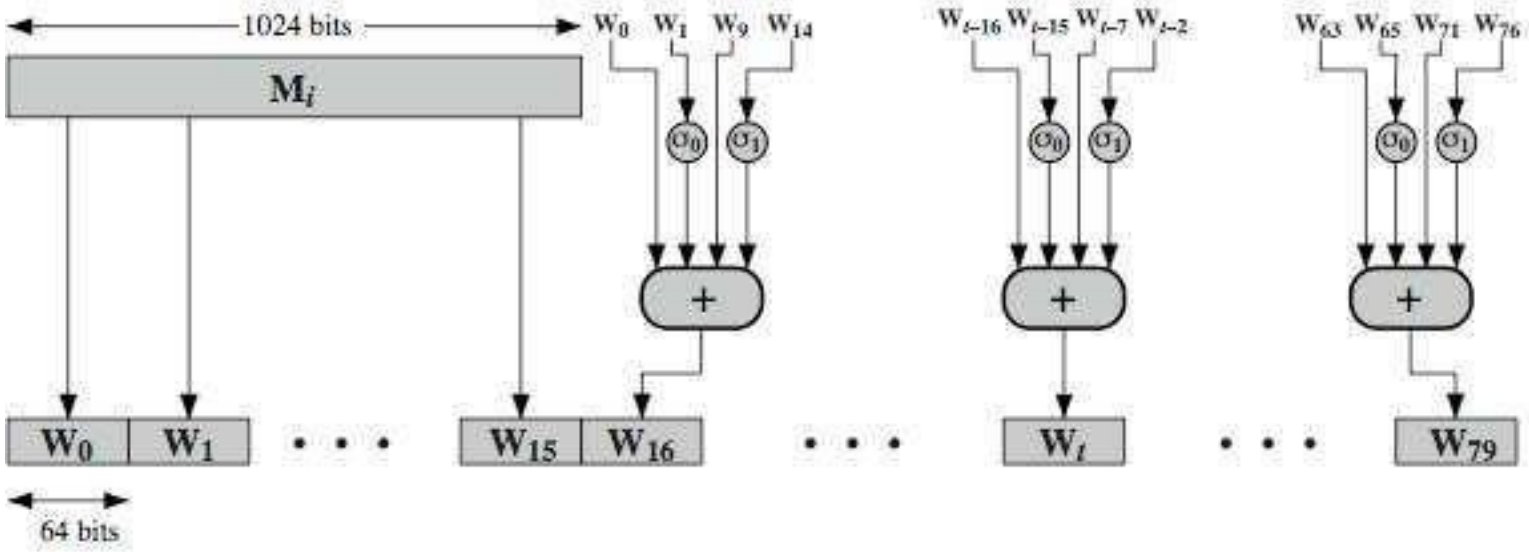


Figure 3.6: SHA-512 Round Function Decryption

A-3

- SHA-1 not yet "broken”
  - But similar to broken MD5 & SHA-0
  - So considered insecure
- SHA-2 (esp. SHA-512) seems secure
  - Shares same structure and mathematical operations as predecessors so have concern
- NIST announced in 2007 a competition for the SHA-3 next gen NIST hash function
- Keccak winner oct 2012 – std in Q2,2014

SHA-3

Requirements

- Replace SHA-2 with SHA-3 in any use
  - So use same hash sizes
- Preserve the online nature of SHA-2
  - So must process small blocks (512 / 1024 bits)
- Evaluation criteria
  - Security close to theoretical max for hash sizes
  - Cost in time & memory
  - Characteristics: such as flexibility & simplicity

TENTS

- **Message authentication codes**
- **Message authentication requirements**
- **Message authentication functions**
- **Requirements for message authentication codes**
- **Security of macs**
- **HMAC**
- **Macs based on block ciphers**
- **Authenticated encryption**
- **Digital signatures-rsa with SHA & DSS**

AUTHENTICATION CODES

- Message authentication codes, like block ciphers, are *symmetrically-keyed* cryptographic primitives.
- Like cryptographic hash functions, MACs take arbitrary-length input and produce a fixed length output.
- Not invertible.

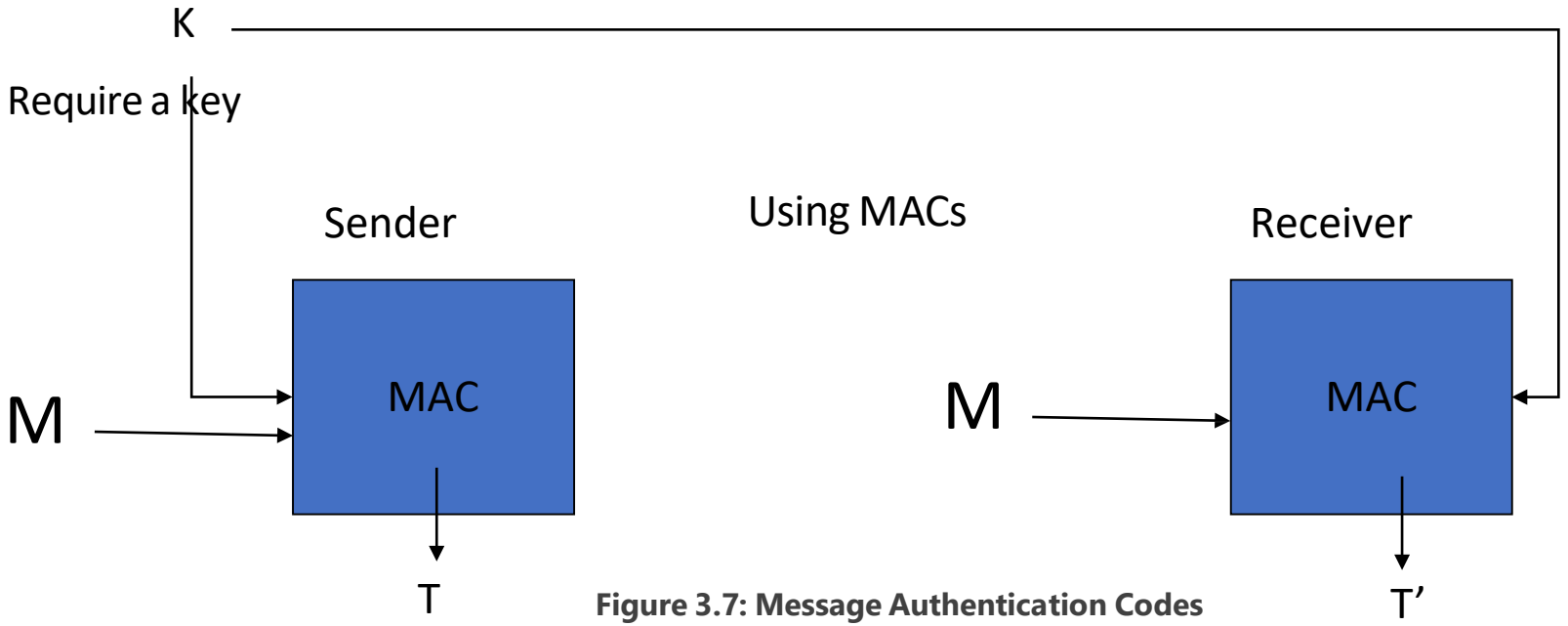


Figure 3.7: Message Authentication Codes

AUTHENTICATION

Why is Message Authentication?

- protecting the integrity ofamessage
- validating identity oforiginator
- non-repudiation of origin (disputeresolution)

MessageAuthentication Functions

- HashFunction
- MessageEncryption
- MessageAuthentication Code(MAC)

AUTHENTICATION CODE (MAC)

- MAC is a small fixed-length code generated using key (K) and message (M).

  - **MAC=C(K,M)**

- The code generated is not reversible.
- MAC is appended to message as a signature.
- At the receiver side a new MAC is calculated which is supposed to match with the original MAC.
- MAC provides assurance that message is unaltered and comes from sender.
- Unlike Hash function, There may be more than one plain text
  - which can generate the same MAC.

Authentication Code (MAC)

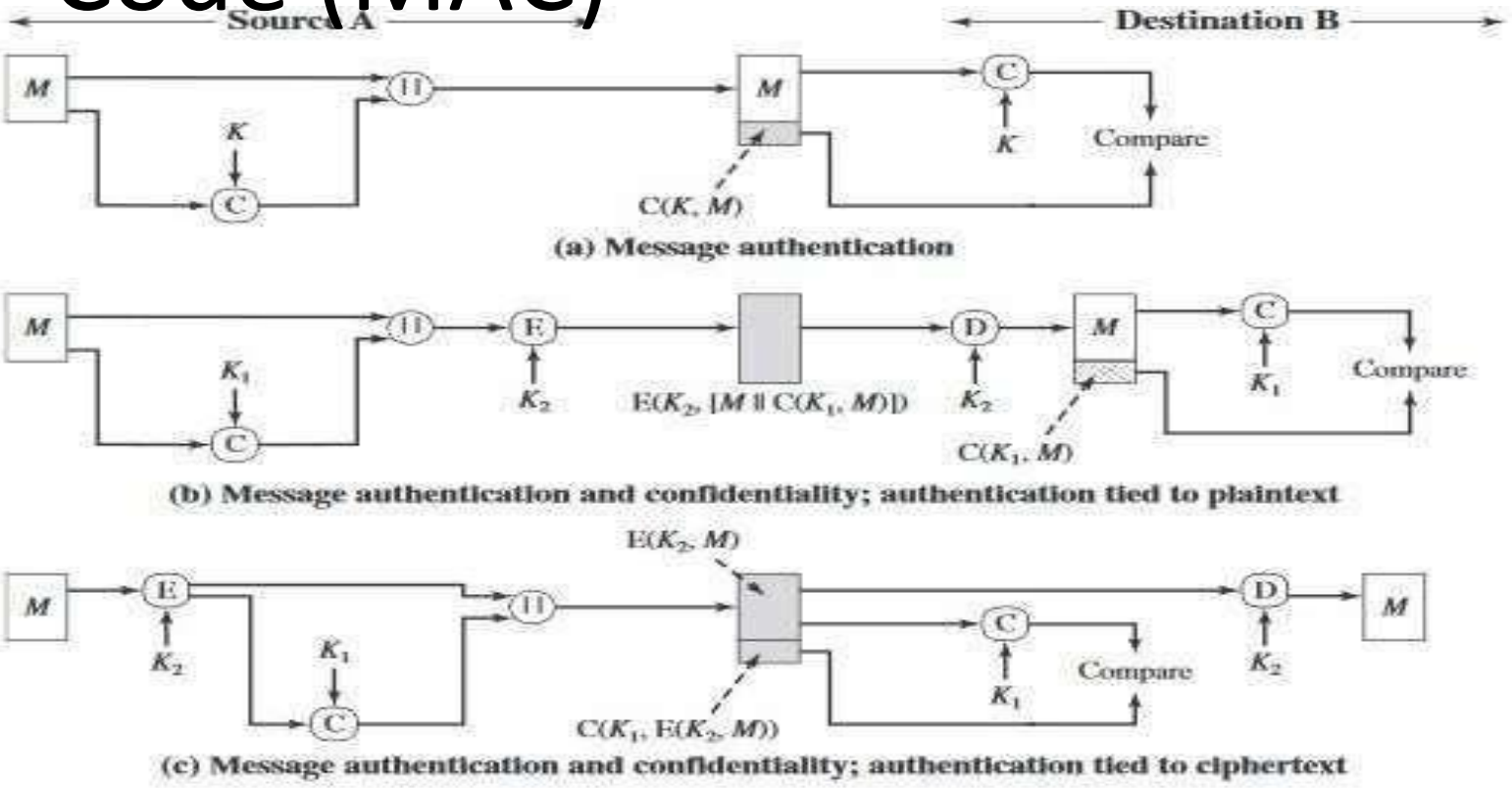


Figure 3.8: MAC operations

AUTHENTICATION REQUIREMENTS

- Disclosure
- Traffic analysais
- Masque rade
- Content modification
- Séquence modification
- Timingmodification
- Sourcerépudiation
- Destination répudiation

MESSAGE AUTHENTICATION FUNCTIONS

- Message authentication or digital signature mechanism can be viewed as having two levels

  –At lower level: there must be some sort of functions producing an authenticator

  – A value to be used to authenticate a message

  –This lower level functions is used as primitive in a higher level authentication protocol.

- Three classes of functions that may be used to produce an authenticator
  - Message encryption: Cipher text itself serves as authenticator
  - Message authentication code (MAC): A public function of the message and a secret key that produces a fixed-length value that serves as the authenticator
  - Hash function: A public function that maps a message of any length into a fixed-length hash value, which serves as the authenticator.

REQUIREMENTS FOR MESSAGE AUTHENTICATION CODES

	Description	Measures
Disclosure	Release of message contents to any person or process not possessing the appropriate cryptographic key.	Measures to deal with the first two attacks are in the realm of message confi-dentiality and are dealt with in Part One.
Traffic analysis	Discovery of the pattern of traffic between parties. In a connection-oriented application, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined.	

Table 3.3: Requirements for MAC

REQUIREMENTS FOR MESSAGE AUTHENTICATION CODES

	Description	Measures
Masquerade	Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or non-receipt by someone other than the message recipient.	Measures to deal with these items are generally regarded as message authentication. Generally, a digital signature technique will also counter some or all of the attacks here.
Content modification	Changes to the contents of a message, including insertion, deletion, transposition, and modification.	

Table 3.3: Requirements for MAC

REQUIREMENTS FOR MESSAGE AUTHENTICATION CODES

	Description	Measures
Sequence modification	Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.	Measures to deal with these items are generally regarded as message authentication.
Timing modification	Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.	Generally, a digital signature technique will also counter some or all of the attacks here.

Table 3.3: Requirements for MAC

REQUIREMENTS FOR MESSAGE AUTHENTICATION CODES

SVEC TIRUPATI
EDUCATION FOR A BETTER SOCIETY

	Description	Measures
Source repudiation	Denial of transmission of message by source.	Mechanisms for dealing specifically with this comes under the heading of digital signatures.
Destination repudiation	Denial of receipt of message by destination.	Dealing with this item may require a combination of the use of digital signatures and a protocol designed to counter this attack.

Table 3.3: Requirements for MAC

SECURITY OF MACS

- In cryptography, a message authentication code (**MAC**), sometimes known as a tag, is a short piece of information used to authenticate a message.
- In other words, to confirm that the message came from the stated sender (its authenticity) and has not been changed.
- MAC algorithm is a symmetric key cryptographic technique to provide message authentication.
- For establishing MAC process, the sender and receiver share a symmetric key K.
- Essentially, a MAC is an encrypted checksum generated on the underlying message that is sent along with a message to ensure message authentication.

HMAC

- A FIPS standard for constructing MAC from a hash function *h*.

Conceptually,

$$HMAC_k(m) = h(k_2 \quad h(k_1 \quad m))$$

where k_1 and k_2 are two keys generated from *k*.

Various hash functions (e.g., SHA-1, MD5) may be used for *h*. If we use SHA-1, then HMAC is as follows: = ⊕ ⊕

$HMAC_k(m)$ SHA-1 *k* *opad* SHA-1(*k* *ipad* *m*)

Where *k* is padded with 0's to 512 bits

ipad = 3636 36 (x036 repeated 64 times)

opad 5c5c 5c (x05c repeated 64 times)

MACS BASED ON BLOCK CIPHERS

SVEC TIRUPATI
EDUCATION FOR A BETTER SOCIETY

- A FIPS and ISO standard.
- One of the most popular MACs in use.
- Use a block cipher in CBC mode with a fixed, public IV. Called DES CBC-MAC if the block cipher is DES.

Let E :$\{0,1\}^n$ $\{0,1\}^n$ be a block cipher. CBC-MAC(m,k)

$M=m_1,m_2,m_l$, where$|m_i|=n$.

$C \leftarrow$ IV (typically 0^n)

i$\leftarrow$1 to l do

return(c_l)

$c_i \leftarrow E_k(c_{i-1} \oplus m_i)$

CIPHER BLOCK CHAINING (CBC)

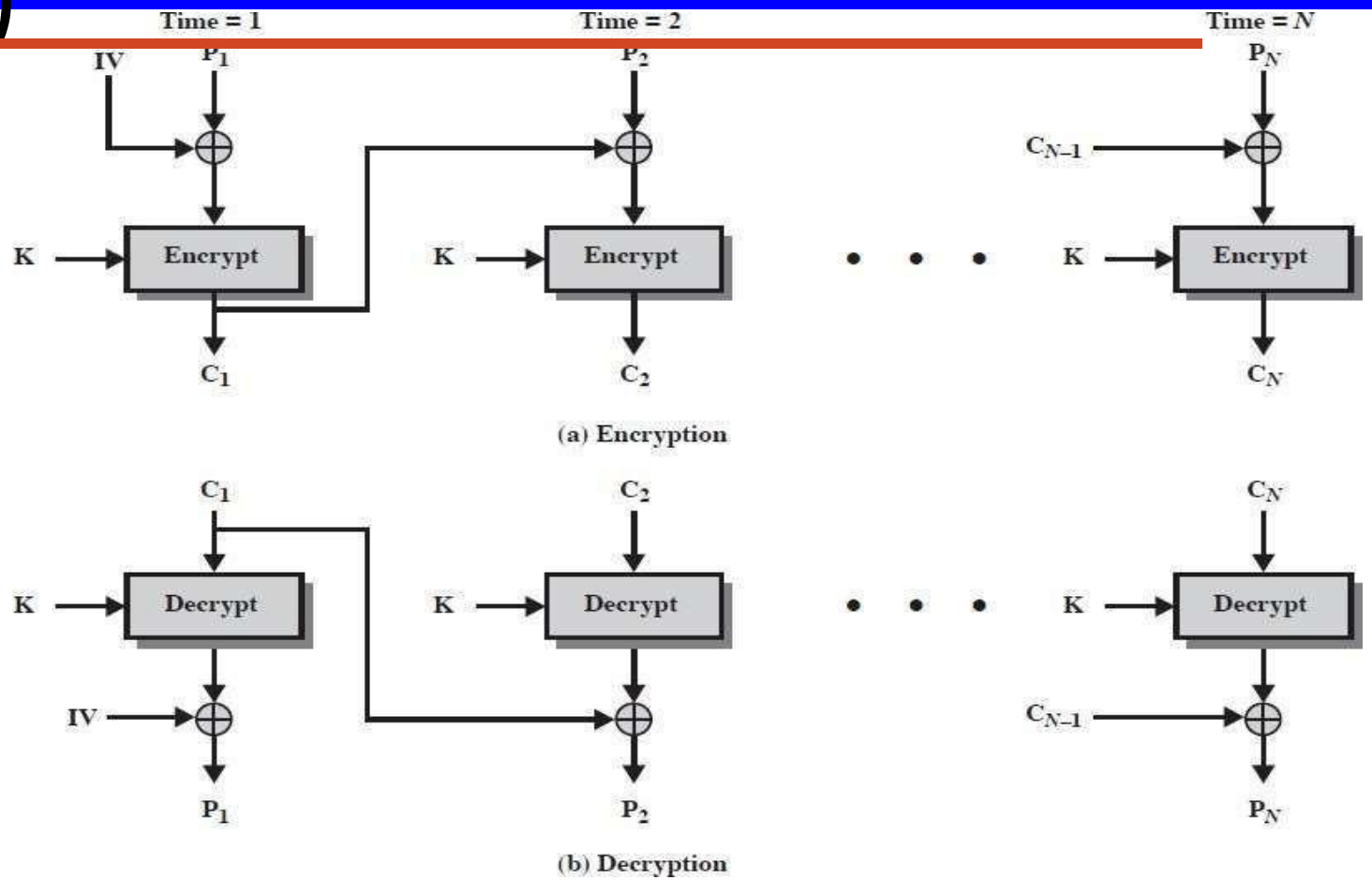


Figure 3.9:Cipher Block Chaining(CBC)

CMAC (CIPHER-BASED MAC)

- A refined version of CBC-MAC.
- Adopted by NIST for use with AES and 3DES.
- Use two keys: k, k' (assuming $|m|$ is a multiple of n).
- Let $E : \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher. CMAC(m,
- k)

$m = m_1 \| m_2 \| \ldots \| m_l$, where $|m_i| = n$.

$c_0 \leftarrow$ IV (typically 0^n)

for $i \leftarrow 1$ to $l - 1$ do

$c_i \leftarrow E_k(c_{i-1} \oplus m_i)$

$c_l \leftarrow E_k(c_{l-1} \quad m_l$ [illegible] $k')$

return(c_l)

AUTHENTICATED ENCRYPTION

- An *authenticated encryption* system (E,D) is a cipher where

  As usual: $E: K \times M \times N \longrightarrow C$

  but $D: K \times C \times N \longrightarrow M \cup \{\perp\}$

reject ciphertext as invalid

- Security: the system must provide

  Semantic security under CPA attack, ***and***

  - *ciphertext integrity*. The attacker cannot create a new ciphertext that decrypts properly.

TEXT INTEGRITY

For b ={0,1}, define EXP(0) and EXP(1) as:

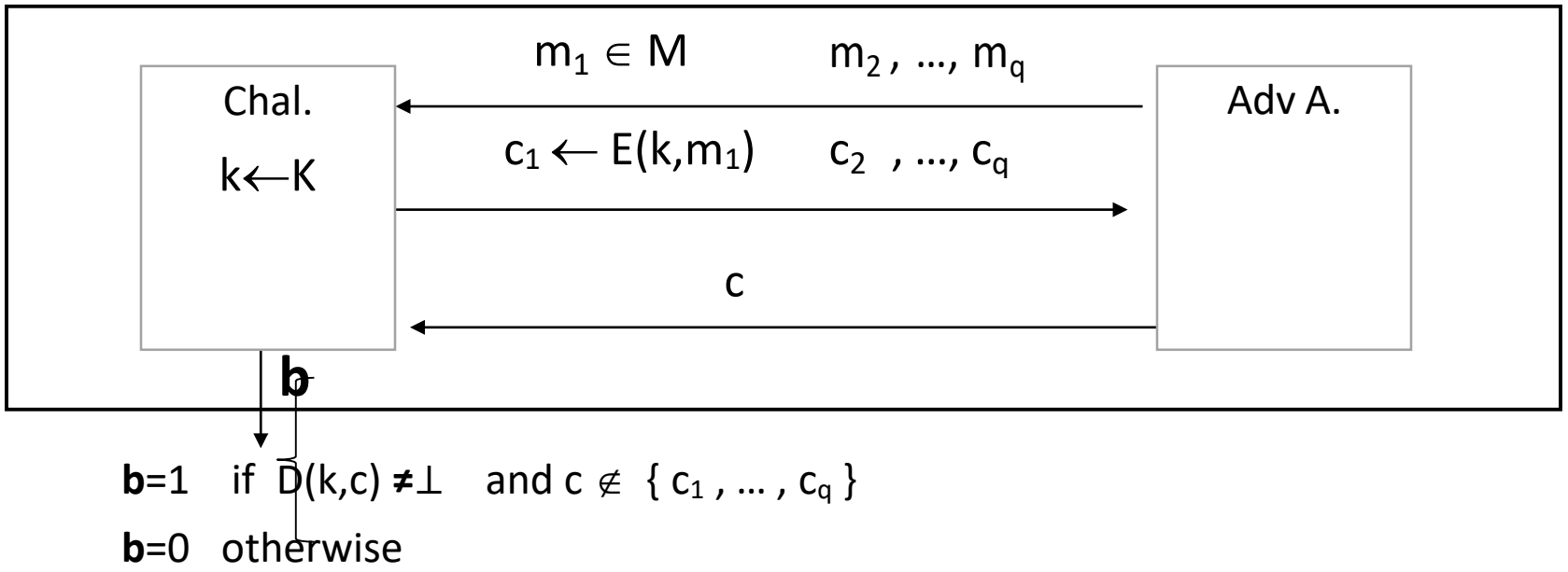


Figure 3.10 : Cipher Text Integrity

Def: (E,D) has *ciphertext integrity* iff for all "efficient" A:

$Adv_{CI}[A,I] = Pr$ [Chal. outputs 1] < ε

AUTHENTICATED ENCRYPTION

Def: cipher (E,D) provides *authenticated encryption (AE)* if it is

(1) semantically secure under CPA, and

(2) has ciphertext integrity

Counter-example: CBC with rand. IV does not provide AE

D(k, ·) never outputs ⊥, hence adv. always wins ciphertext integrity game

IMPLICATION 1: AUTHENTICITY

Attacker cannot fool Bob into thinking a message was sent from Alice

⇒ if D(k,c) ≠⊥ Bob *guaranteed* message is from someone who knows k (but could be a replay)

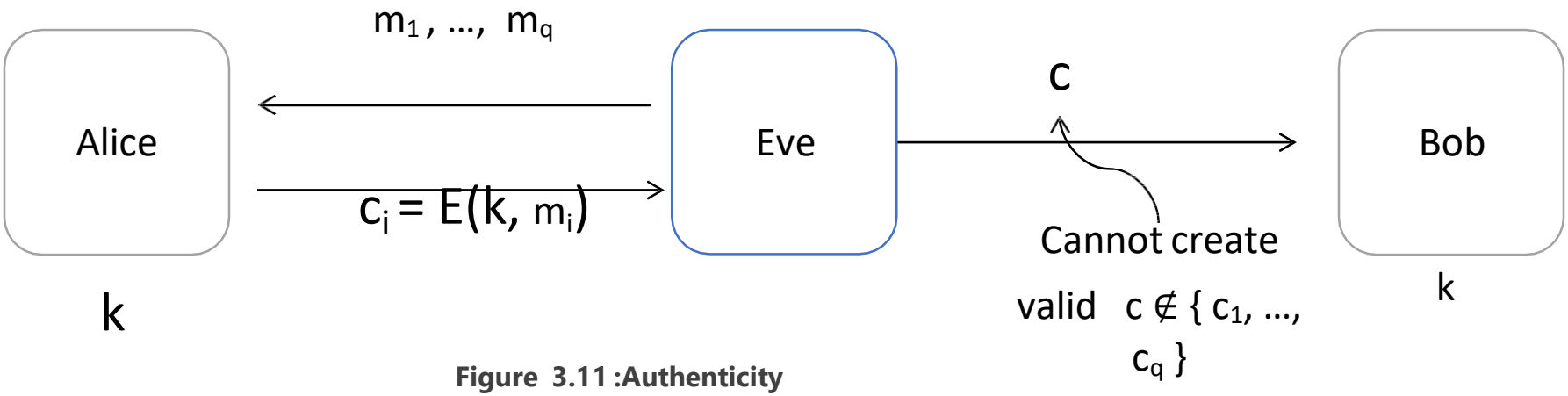


Figure 3.11 :Authenticity

IMPLICATION 2

Authenticated encryption ⇒

Security against *chosen ciphertext attack*

DIGITAL SIGNATURES- RSA WITH SHA & DSS

- RSA can be used for digital signatures.
- A digital signature is the same as a MAC except that the tag (signature) is produced using a public-key cryptosystem.
- Digital signatures are used to provide message authentication and non-repudiation.

Message m $MAC_k(m)$

Message m $Sig_{pr}(m)$

DIGITAL SIGNATURES- RSA WITH SHA & DSS

- Digital signature protocol:
- Bob has a key pair (*pr*,*pu*).
- Bob sends m=Sig $_{pr}$ (m) toAlice.
- Alice verifies the received m s
  - by checking if s=Verify$_{pu}$ (m).
- Sig $_{pr}$ (m) is called a signature for*m*.

Security requirement: infeasible to forge a valid pair (m, Sig $_{pr}$ (m)) without knowing *pr*.

ENCRYPTION (USING RSA):

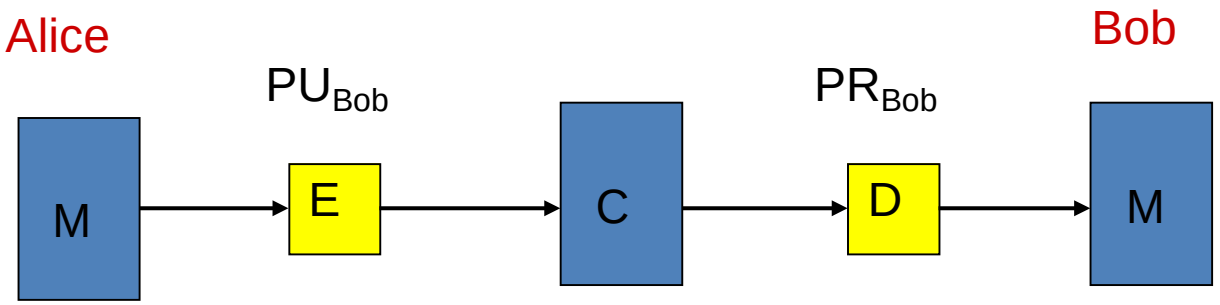


Figure 3.12 : Encryption

Digital signature (using RSA^{-1}):

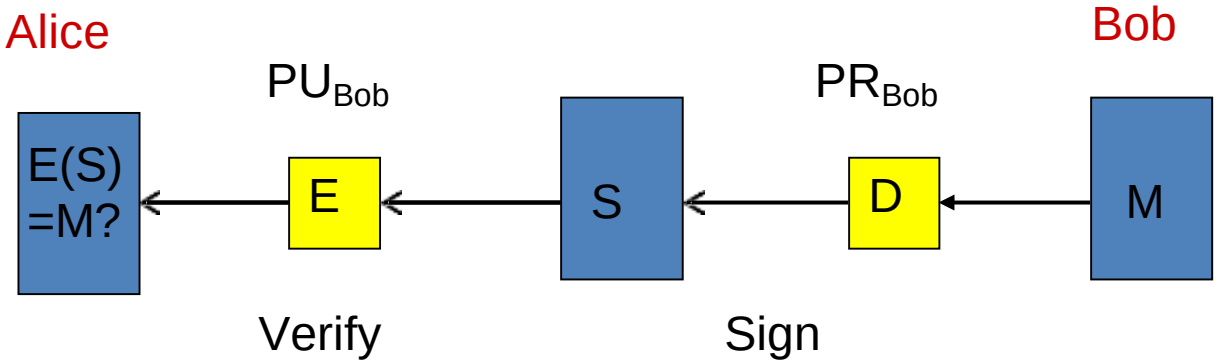


Figure 3.13: Digital Signature

- Keys are generated as for RSA encryption:

  Public key: *PU* $= (n,e)$. Private key: *PR* $= (n,d)$.

- Signing a message *m* $\in Z_n^*$: $\sigma = D_{PR_1}(m) = m^d \bmod n$.

  That is, $\sigma = \text{RSA}(m)$.

- Verifying a signature (m, σ):

  check if $m = E_{PU}(\sigma) = \sigma^e \bmod n$, or $m = \text{RSA}(\sigma)$.

- Only the key's owner can sign, but anybody can verify.

- Existential forgeries:

1. Every message m its ciphertext c: $\in Z_n^*$ is a valid signature for RSA(m).

Encryption (using Bob's public key): $m \xrightarrow{\text{RSA}} c$

Sign (if using Bob's private key): $m \xleftarrow{\text{RSA}^{-1}} c$

2. If Bob signed m_1 and m_2, then the signature for $m_1 m_2$ can be easily forged: $\sigma(m_1 m_2) = \sigma(m_1)\,\sigma(m_2)$.

Countermeasure:

- hash and sign:

$\sigma =$ Sign$_{PR}(h(m))$, using some collision resistant hash function h.

CONTENT BEYOND SYLLABUS

OUT LINE

- COLLISION-RESISTANT HASH FUNCTION
- BIRTHDAY ATTACK'S
- TOSS A COIN BY EMAIL

RESISTANT HASH FUNCTION

- Let $\Pi = (Gen, H)$ be a hash function.
- Collision-finding experiment Hash-coll$_{A,\Pi}(n)$: A key is generated, $s \leftarrow Gen(1^n)$.
  - The adversary A is given s and outputs $x, x' \in \{0,1\}^*$ (or $x, x \in \{0,1\}^{l(n)}$ if Π is fixed-length).
  - The output of the experiment is 1 if and only if $x \neq x'$ and $H^s(x) = H^s(x')$. //A finds a collision//
- Definition: A hash function $\Pi = (Gen, H)$ is collision-resistant if for all PPT adversaries A, there is a $negl(n)$ such that

$$\Pr[\text{Hash-coll}_{A,\Pi}(n) = 1] \leq negl(n).$$

BIRTHDAY ATTACK'S

Birthday attack's success rate

- If k objects are each assigned a random value in $1,2, \ldots, N$, the probability of a collision is

$$p = 1 - 1 \cdot \frac{N-1}{N} \cdot \frac{N-2}{N} \cdot \ldots \cdot \frac{N-k+1}{N} \quad \text{(i.e., } 1 - \Pr[\text{no collision}]\text{)}$$

$$= 1 - \prod_{i=1}^{k-1} \left(1 - \frac{i}{N}\right) \quad \text{(note: } 1 - x \le e^{-x} \text{ if } 0 < x < 1\text{)}$$

$$\ge 1 - \prod_{i=1}^{k-1} e^{-i/N} = 1 - e^{-\sum_{1 \le i \le k-1} i/N} = 1 - e^{-k(k-1)/2N}$$

- $p \ge 1/2 \quad \text{if} \quad k \ge 1.17\sqrt{N}$.

- Birthday paradox: with $N = 365$, $p \ge 1/2$ for k as small as 23.

TOSS A COIN BY EMAIL

- Problem: Alice and Bob want to toss a coin by email to decide who is going to pay for dinner.
- A proposed solution:
  - Use a collision resistant hash function h.
  - Alice chooses a string x_1 and compute $y_1 := h(x_1)$. Bob chooses a string x_2 and compute $y_2 := h(x_2)$.
  - Alice and Bob exchange y_1 and y_2. Alice and Bob exchange x_1 and x_2.

    //commit but hide x_1 and x_2 //

    //reveal x_1 and x_2 //
  - Alice and Bob check if $y_2 := h(x_2)$, $y_1 := h(x_1)$, respectively. Alice and Bob compute a boolean value from x_1 and x_2 (e.g., take the XOR of the last bits).
- Is the proposed scheme "secure/fair"?

RESOURCES

- Lecture Notes - Lecture Notes
- Video Lectures - Video Lecture
- E-Book - Information Security Concepts
- Model Papers - JNTUA Question Papers

DEPT & SEM : CSE-cs & ISEM

SUBJECT NAME: INFORMATION SECURITY

COURSE CODE : IS

UNIT : IV

PREPARED BY : Anusha K

OUT LINE

- **Key Management and distribution**
- **Symmetric key distribution using Symmetric**
- **Encryption**
- **Symmetric key distribution using Asymmetric**
- **Distribution of Public keys**
- **X.509 Certificates**
- **Public key Infrastructure**

KEY MANAGEMENT AND DISTRIBUTION

- Topics of cryptographic key management / key distribution are complex

  Cryptographic, protocol, & management issues

- Symmetric schemes require both parties to share a common secret key
- Public key schemes require parties to acquire valid public keys
- Have concerns with doing both

DISTRIBUTION

- Symmetric schemes require both parties to share a common secret key
- Issue is how to securely distribute this key
- Whilst protecting it from others
- Frequent key changes can be desirable
- Often secure system failure due to a break in the key distribution scheme

DISTRIBU TION

- Given parties A and B have various **key distribution** alternatives:
  - A can select key and physically deliver to B
  - Third party can select & deliver key to A & B
  - If A & B have communicated previously can use previous key to encrypt a
    - new key

DISTRIBUTION TASK

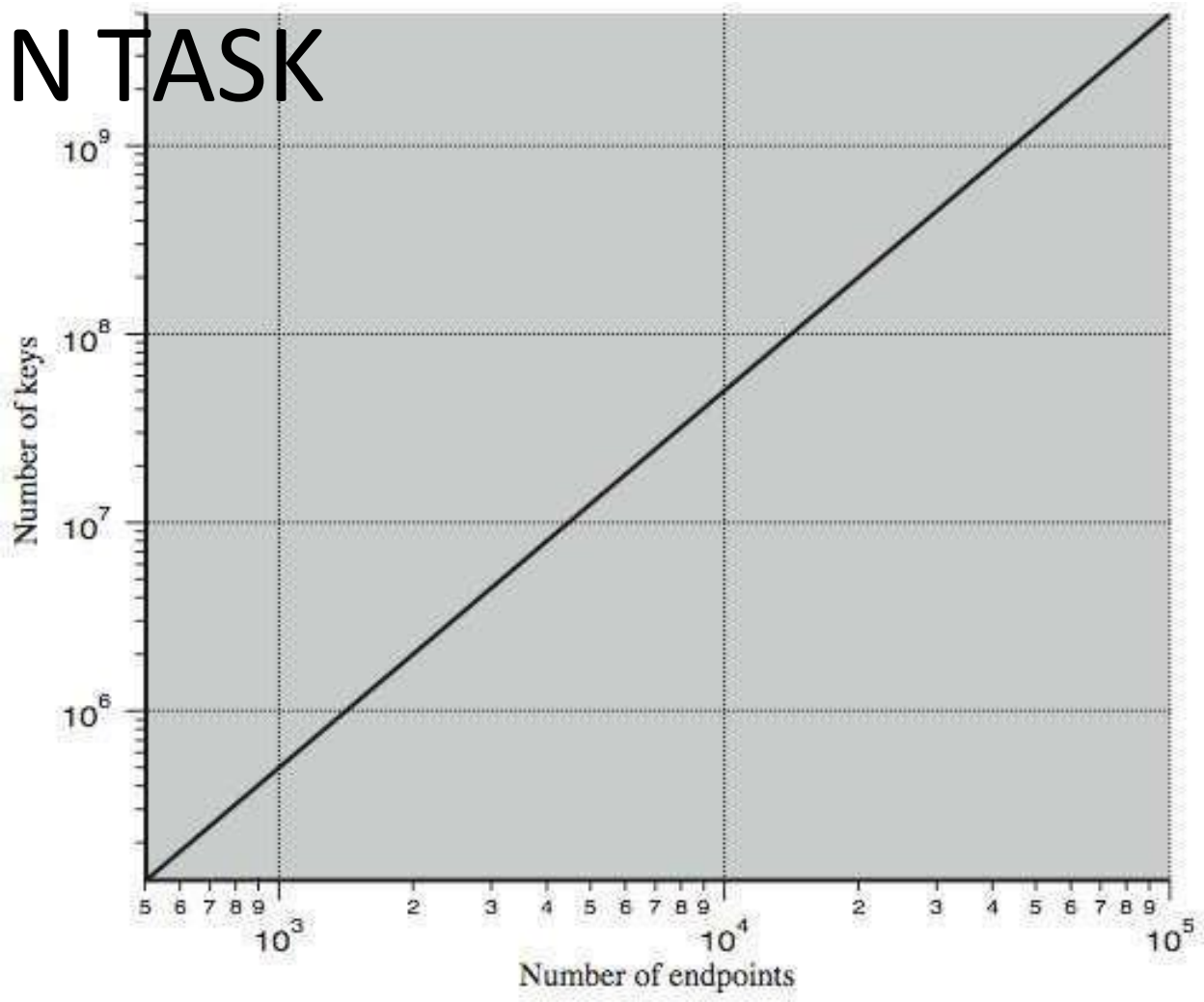


Fig . 4.1

HIERARCHY

- Typically have a hierarchy of keys
- Session key
  - Temporary key
  - Used for encryption of data between users
  - For one logical session then discarded
- Master key
  - Used to encrypt session keys
  - Shared by user & key distribution center

HIERAR

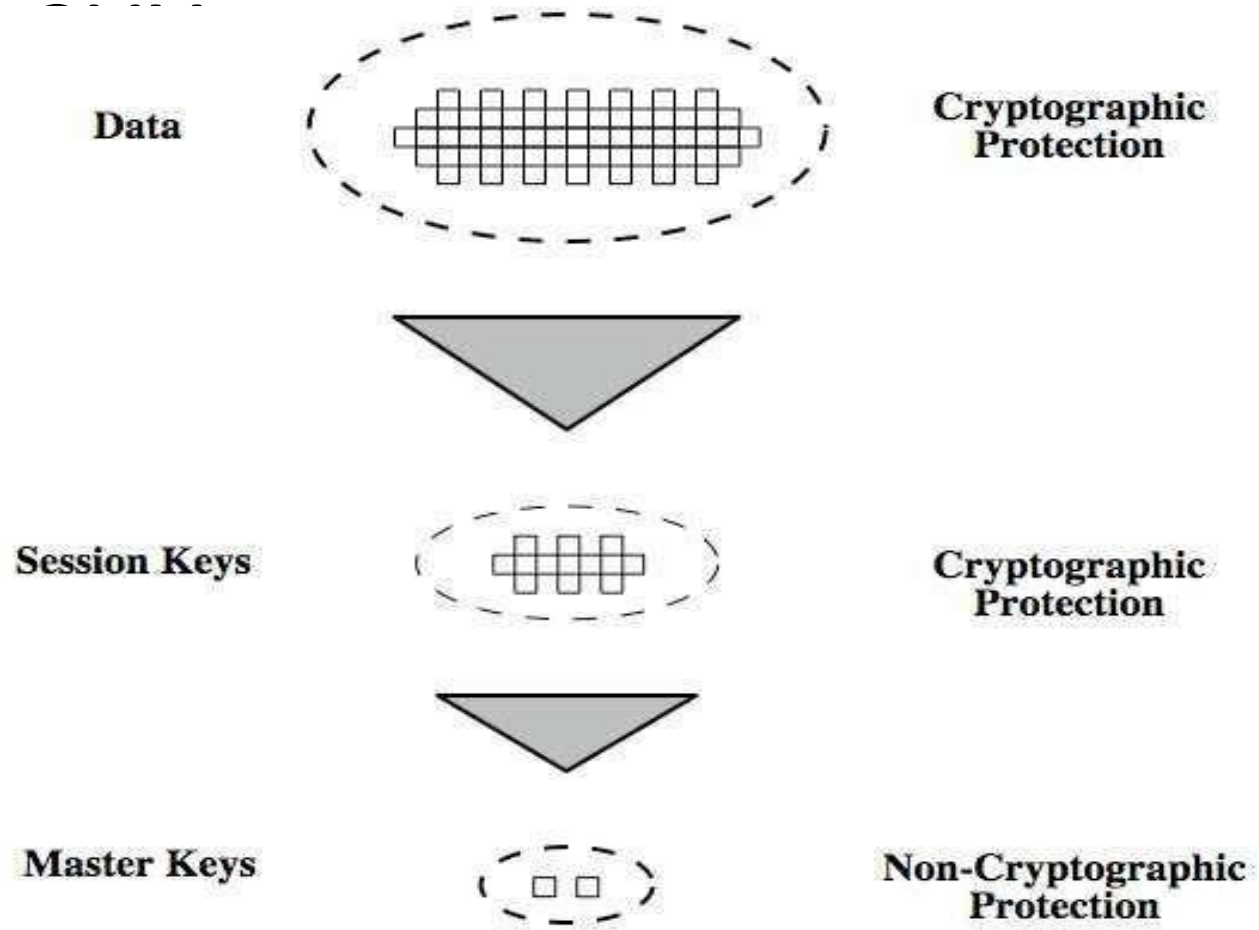


Fig . 4.2

DISTRIBUTION SCENARIO

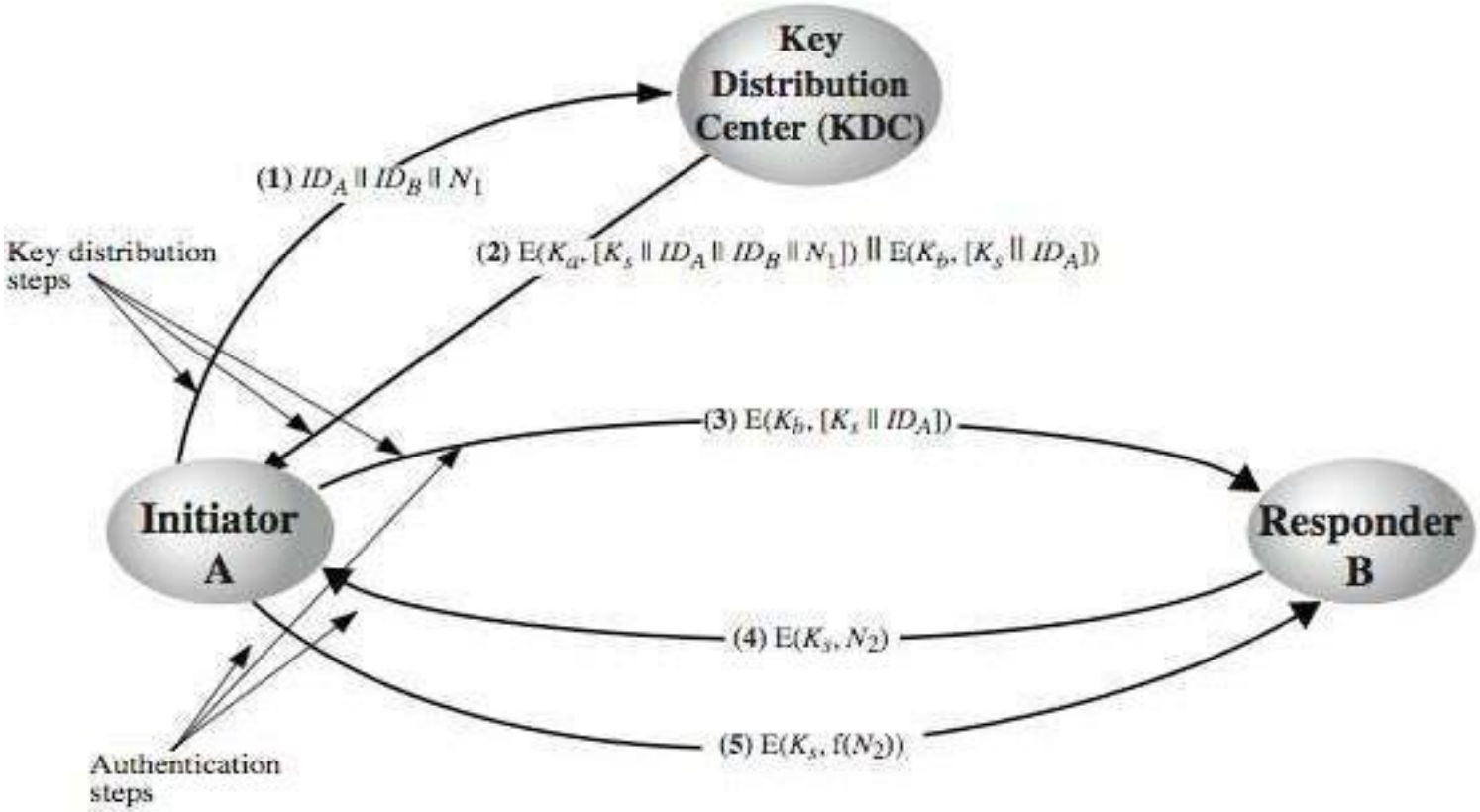


Fig . 4.3

DISTRIBUTION ISSUES

- Hierarchies of kdc's required for large networks, but must trust each other
- Session key lifetimes should be limited for greater security
- Use of automatic key distribution on behalf of users, but must trust system
- Use of decentralized key distribution
- Controlling key usage

DISTRIBUTION USING PUBLIC KEYS

- Public key cryptosystems are inefficient
  - So almost never use for direct data encryption
  - Rather use to encrypt secret keys for distribution

SIMPLE SECRET KEY DISTRIBUTION

- Merkle proposed this very simple scheme
- Allows secure communications
- No keys before/after exist

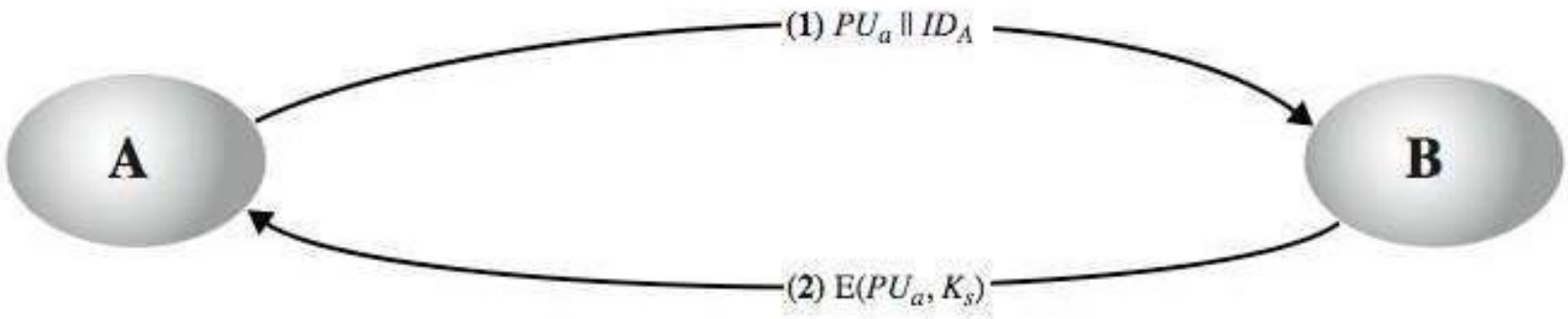


Fig . 4.4

MIDDLE ATTACK

- This very simple scheme is vulnerable to an active man-in-the-middle attack

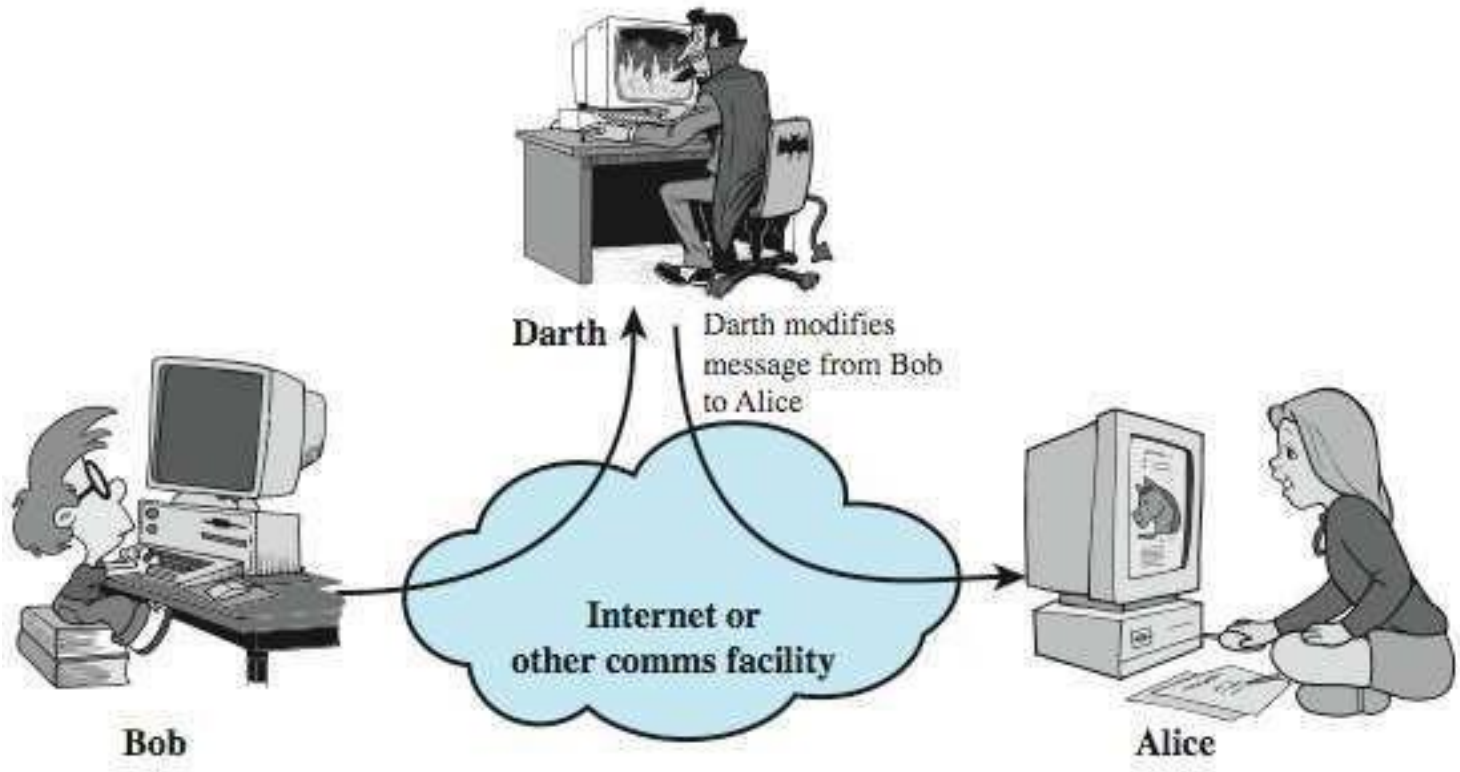


Fig . 4.5

Confidentiality and Authentication

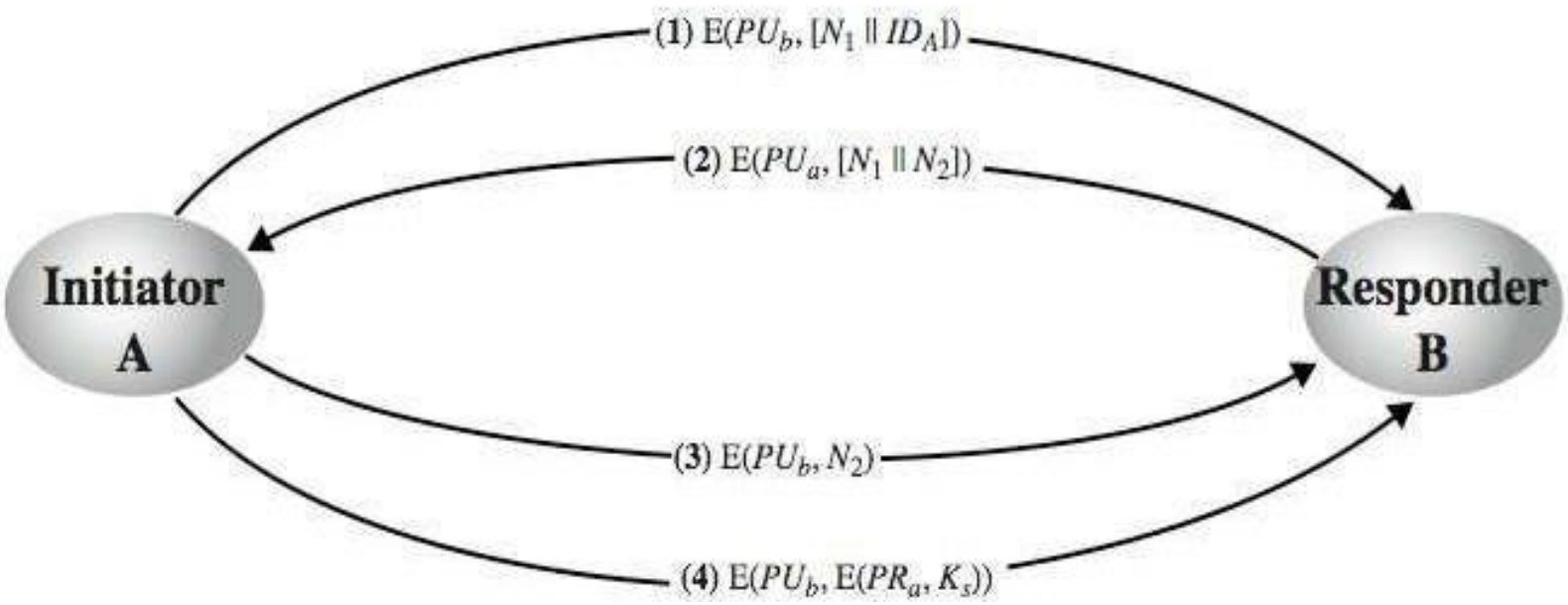


Fig . 4.6

DISTRIBUTION

- Retain use of private-key KDC
- Shares secret master key with each user
- Distributes session key using master key
- Public-key used to distribute master keys
  - Especially useful with widely distributed users
- Rationale
  - Performance
  - Backward compatibility

DISTRIBUTION OF PUBLIC KEYS

Can be considered as using one of:

- Public announcement
- Publicly available directory
- Public-key authority
- Public-key certificates

ANNOUNCE MENT

- Users distribute public keys to recipients or broadcast to community at large

  Eg. Append PGP keys to email messages or post to news groups or email list

- Major weakness is forgery
  - Anyone can create a key claiming to be someone else and broadcast it
  - Until forgery is discovered can masquerade as claimed user

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 register securely with directory
  - Participants can replace key at any time
  - Directory is periodically published
  - Directory can be accessed electronically
- Still vulnerable to tampering or forgery

PUBLIC-KEY AUTHORITY

- Improve security by tightening control over distribution of keys from directory
- Has properties of directory
- And requires users to know public key for the directory
- Then users interact with directory to obtain any desired public key securely
  - Does require real-time access to directory when keys are needed
  - May be vulnerable to tampering

PUBLIC-KEY AUTHORITY

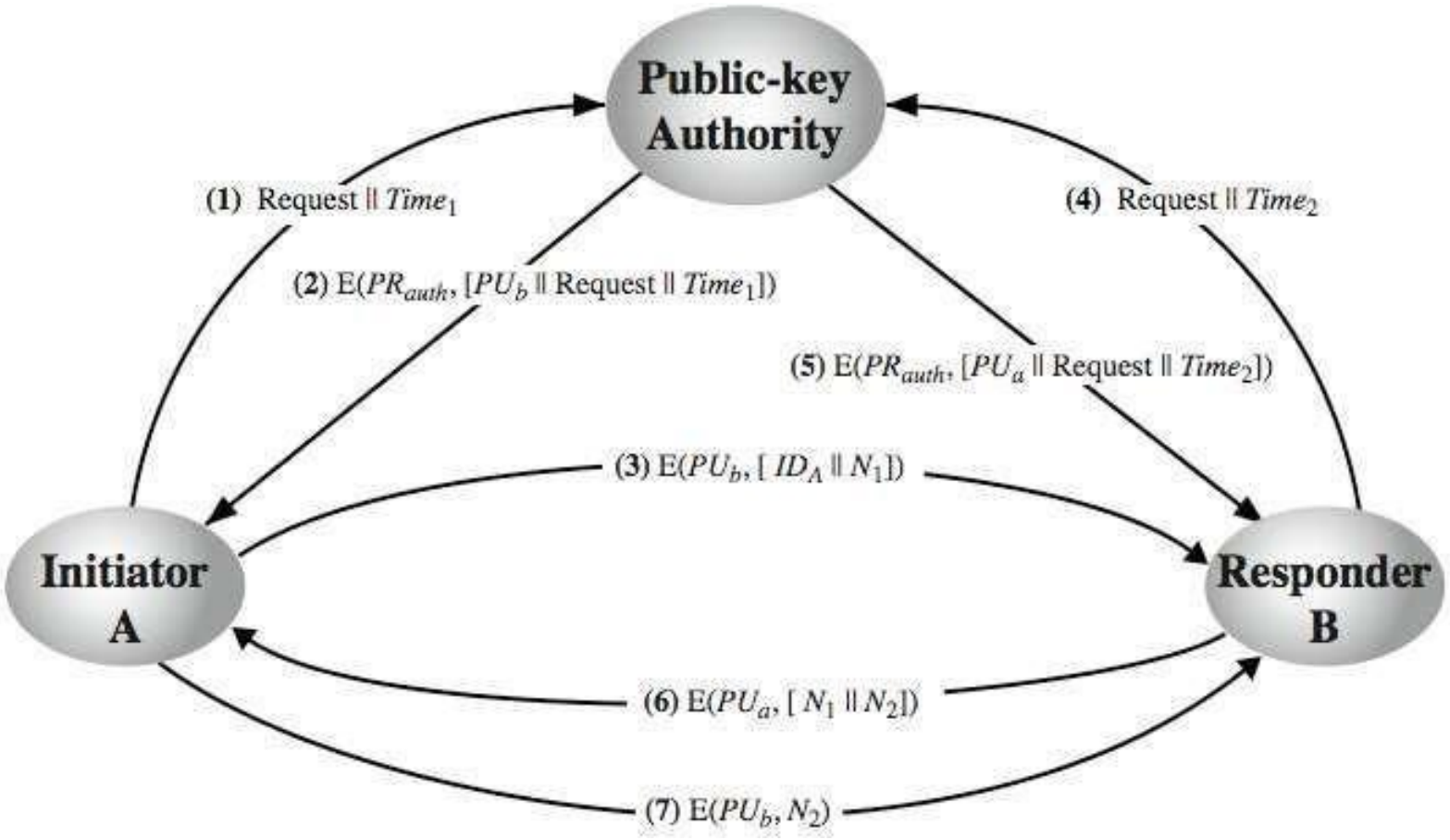


Fig . 4.7

CERTIFICATES

- Certificates allow key exchange without real-time access to public-key authority
- A certificate binds **identity** to **public key**
  - Usually with other info such as period of validity, rights of use etc
- With all contents **signed** by a trusted public-key or certificate authority (CA)
- Can be verified by anyone who knows the public-key authorities public-key

CERTIFICATES

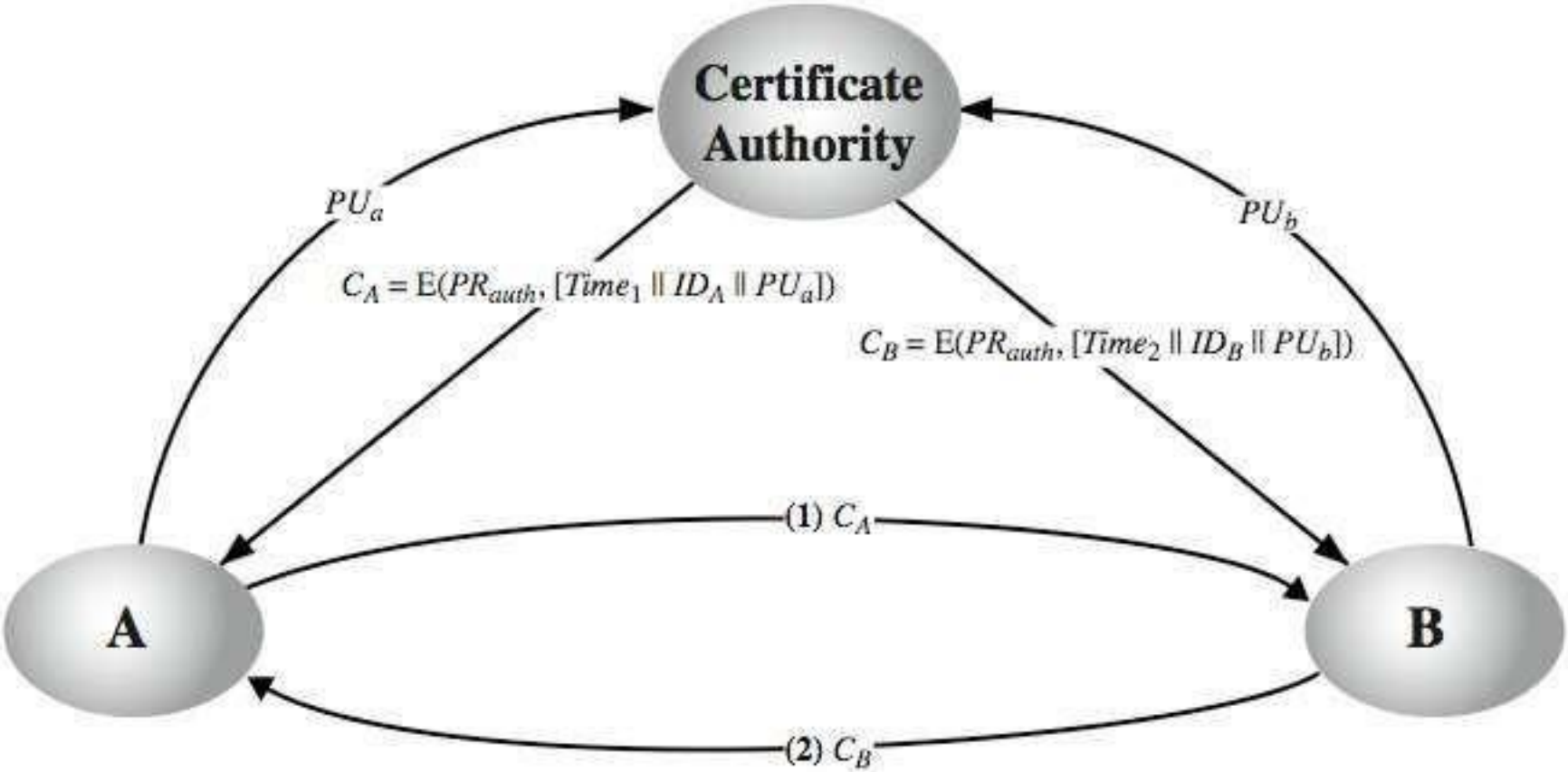


Fig . 4.8

AUTHENTICATION SERVICE

- Part of ITU-T X.500 directory service standards
  - Distributed servers maintaining user info database
- Defines framework for authentication services
  - Directory may store public-key certificates
  - With public key of user signed by certification authority
- Also defines authentication protocols
- Uses public-key crypto & digital signatures
  - Algorithms not standardised, but RSA recommended
- X.509 certificates are widely used
  - Have 3 versions

X.509 CERTIFICATE USE

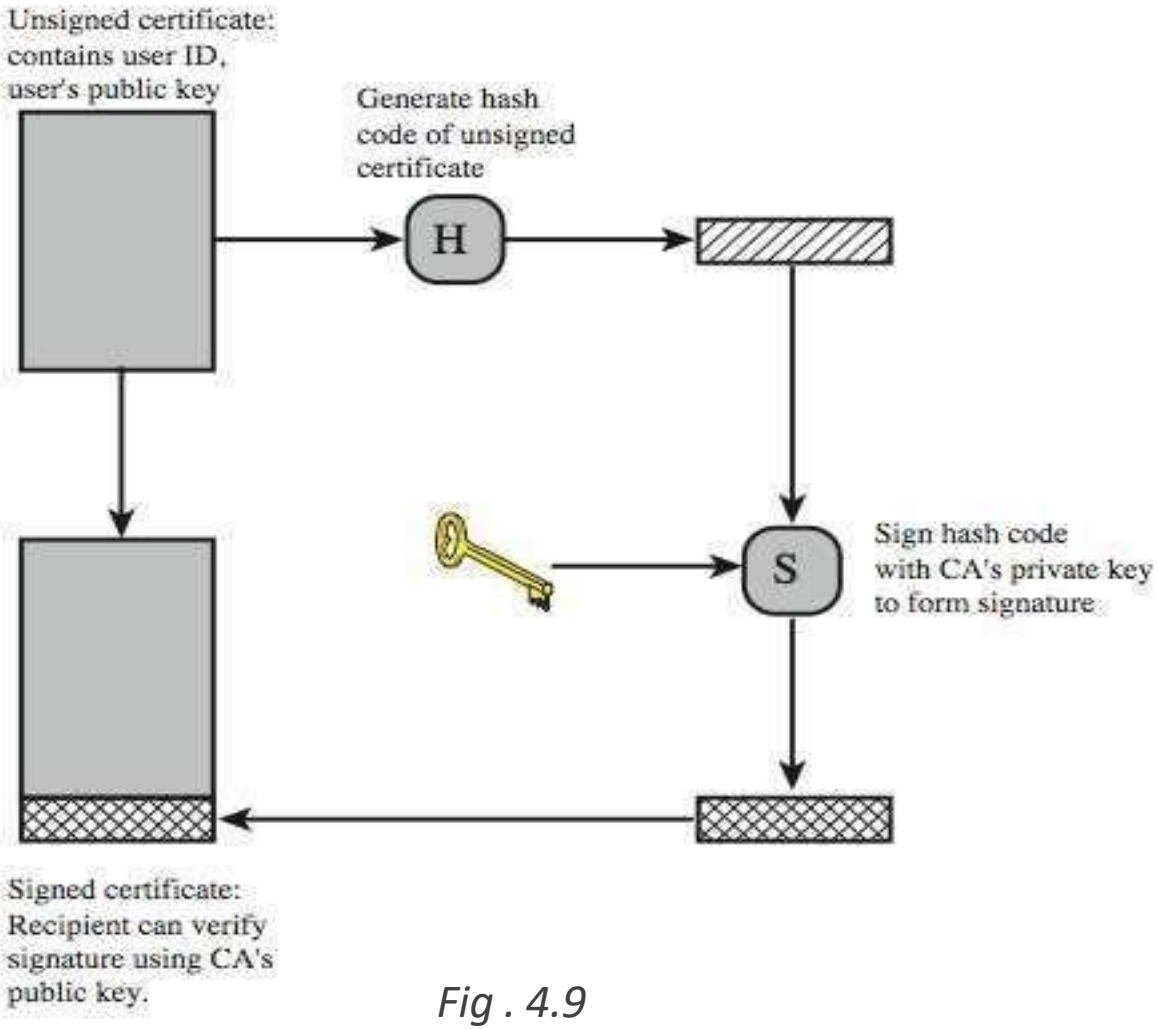


Fig . 4.9

CERTIFICATES

Issued by a certification authority (CA), containing:

- Version V (1, 2, or 3)
- Serial number SN (unique within CA) identifying certificate
- Signature algorithm identifier AI
- Issuer X.500 name CA)
- Period of validity TA (from - to dates)
- Subject X.500 name A (name of owner)
- Subject public-key info ap (algorithm, parameters, key)
- Issuer unique identifier (v2+)
- Subject unique identifier (v2+)
- Extension fields (v3)
- Signature (of hash of all fields in certificate)

Notation CA<<A>> denotes certificate for A signed by CA

CERTIFICATES

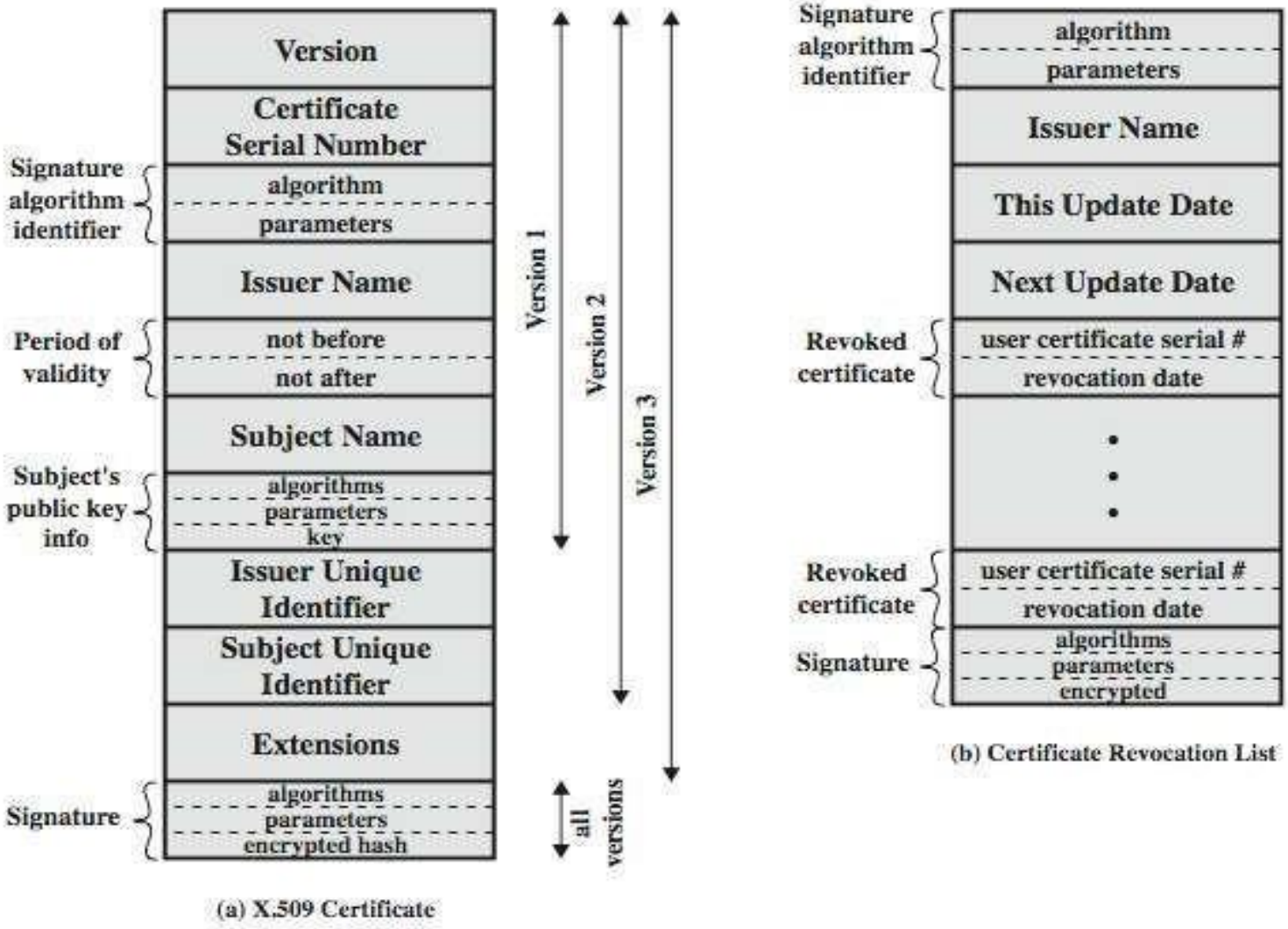


Fig . 4.10

Obtaining a Certificate

- Any user with access to CA can get any certificate from it
- Only the CA can modify a certificate
- Because cannot be forged, certificates can be placed in a public directory

HIERARCHY

- If both users share a common CA then they are assumed to know its public key
- Otherwise ca's must form a hierarchy
- Use certificates linking members of hierarchy to validate other ca's
  - Each CA has certificates for clients (forward) and parent (backward)
- Each client trusts parents certificates
- Enable verification of any certificate from one CA by users of all other cas in hierarchy

HIERARC
HY

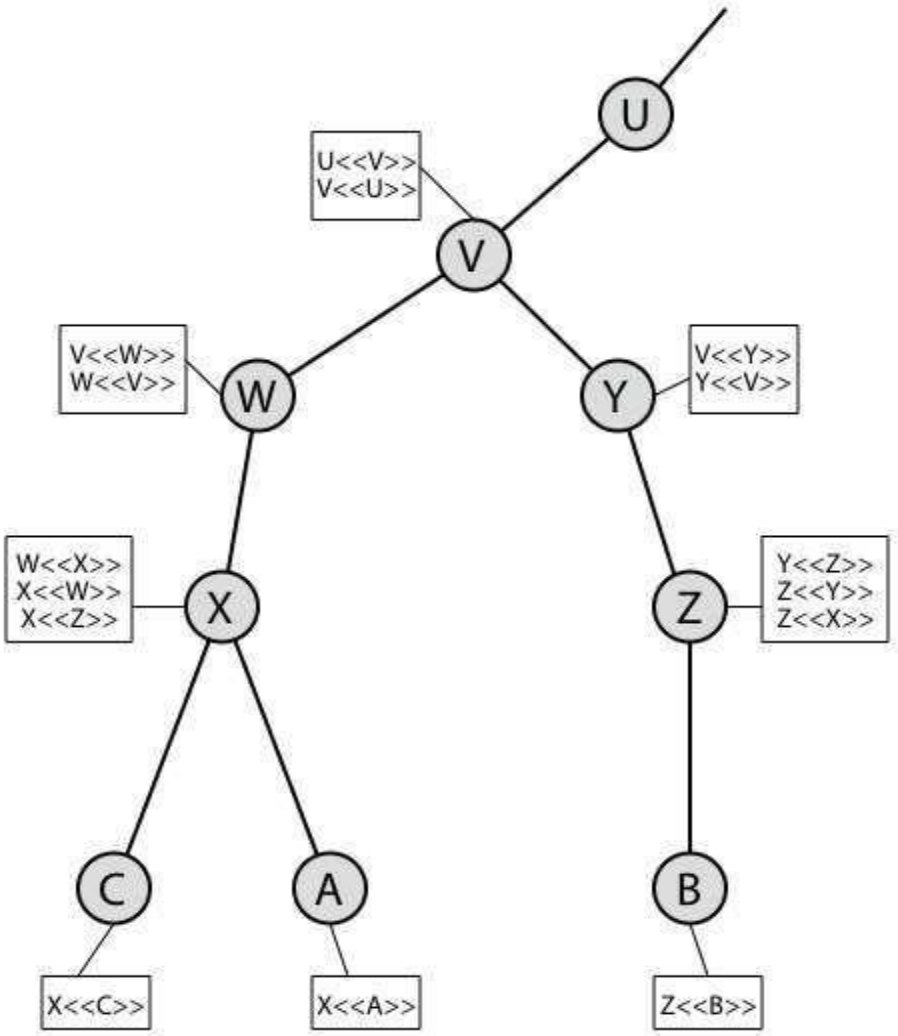


Fig . 4.11

CERTIFICATE REVOCATION

Certificates have a period of validity

May need to revoke before expiry, eg:

- User's private key is compromised
- User is no longer certified by this CA
- Ca's certificate is compromised

Ca’s maintain list of revoked certificates

The certificate revocation list (CRL)

Users should check certificates with ca’s CRL

VERSION 3

- Has been recognised that additional information is needed in a certificate

  Email/URL, policy details, usage constraints

- Rather than explicitly naming new fields defined a general extension method

- Extensions consist of:

  - Extension identifier
  - Criticality indicator
  - Extension value

CERTIFICATE EXTENSIONS

- Key and policy information
  - Convey info about subject & issuer keys, plus indicators of certificate policy
- Certificate subject and issuer attributes
  - Support alternative names, in alternative formats for certificate subject and/or issuer
- Certificate path constraints
  - Allow constraints on use of certificates by other ca's

INFRASTRUCTURE

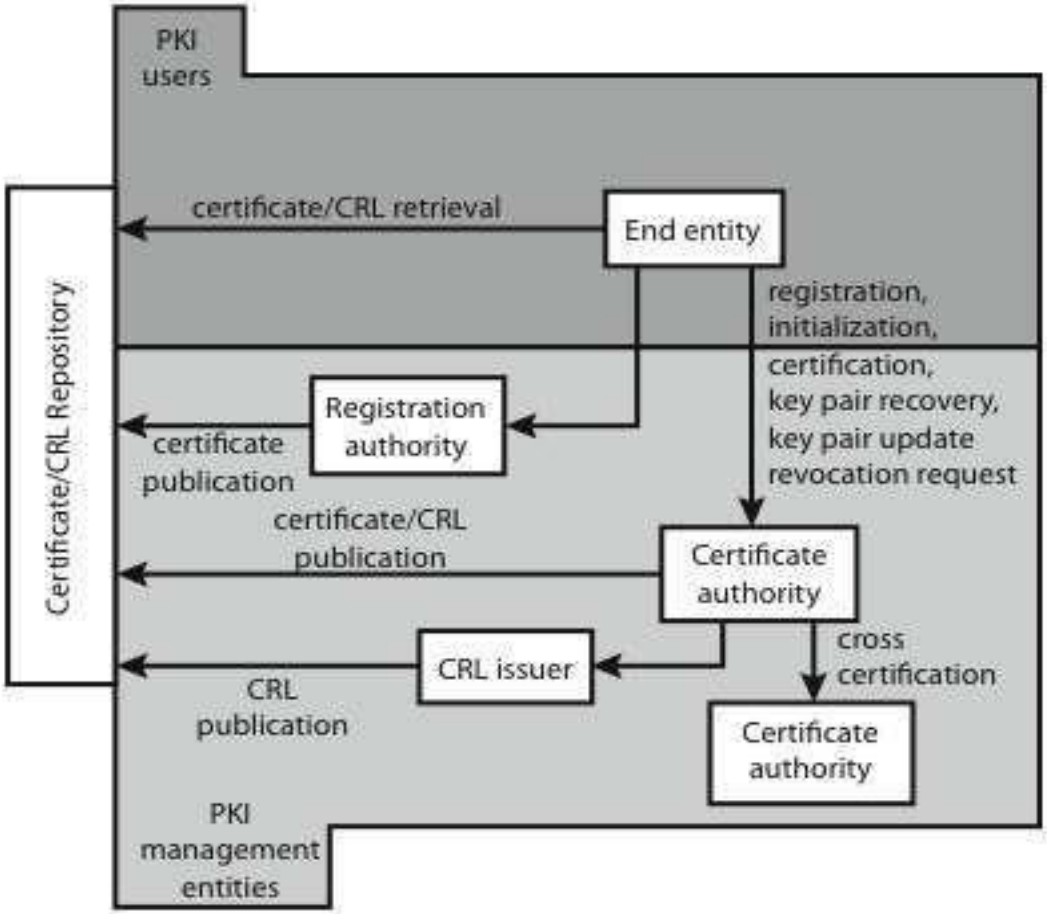


Fig . 4.12

MANAGEMENT

- Functions:
  - Registration
  - Initialization
  - Certification
  - Key pair recovery
  - Key pair update
  - Revocation request
  - Cross certification
- Protocols: CMP, CMC

Information security

UNIT-IV

CHAPTER-II

TENTS

- **User Authentication**
- **Remote user Authentication Principles**
- **Remote user Authentication using Symmetric Encryption**
- **Kerberos**
- **Remote user Authentication using Asymmetric Encryption**
- **Federated Identity Management**
- **Electronic mail security**
- **Pretty Good Privacy (PGP)**
- **S/MIME**

SCHROEDER PROTOCOL

Original third-party key distribution protocol

For session between A B mediated by KDC

Protocol overview is:

- A->kdc: id_a || id_b || n_1
- KDC -> A: e(k_a,[k_s||id_b||n_1|| e(k_b,[k_s||id_a])])
- A -> B: e(k_b, [k_s||id_a])
- B -> A: e(k_s, [N_2])
- A -> B: e(k_s, [f(n_2)])

SCHROEDER PROTOCOL

- Used to securely distribute a new session key for communications between a & b
- But is vulnerable to a replay attack if an old session key has been compromised
  - Then message 3 can be resent convincing b that is communicating with a
- Modifications to address this require:
  - Timestamps in steps 2 & 3 (denning 81)
  - Using an extra nonce (neuman 93)

AUTHENTICATION

- Use refinement of KDC to secure email
  - Since B no online, drop steps 4 & 5
- Protocol becomes:
  - A->kdc: id_a || id_b || n_1
  - KDC -> A: e(k_a, [k_s||id_b||n_1|| e(k_b,[k_s||id_a])])
  - A -> B: e(k_b, [k_s||id_a]) || e(k_s, M)
- Provides encryption & some authentication
- Does not protect from replay attack

KERB
EROS

- Trusted key server system from MIT
- Provides centralised private-key third-party authentication in a distributed network
  - Allows users access to services distributed through network
  - Without needing to trust all workstations
  - Rather all trust a central authentication server
- Two versions in use: 4 & 5

REQUIREMENTS

- Its first report identified requirements as:
  - Secure
  - Reliable
  - Transparent
  - Scalable
- Implemented using an authentication protocol based on needham-schroeder

V4 OVERVIEW

- A basic third-party authentication scheme
- Have an authentication server (AS)
  - Users initially negotiate with AS to identify self
  - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- Have a ticket granting server (TGS)
  - Users subsequently request access to other services from TGS on basis of users TGT
- Using a complex protocol using DES

V4

DIALOGUE

(1) C → AS $ID_c \| ID_{tgs} \| TS_1$

(2) AS → C $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) C → TGS $ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) TGS → C $E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$

$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) C → V $Ticket_v \| Authenticator_c$

(6) V → C $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$

$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$

(c) Client/Server Authentication Exchange to obtain service

Fig . 4.13

KERBEROS 4 OVERVIEW

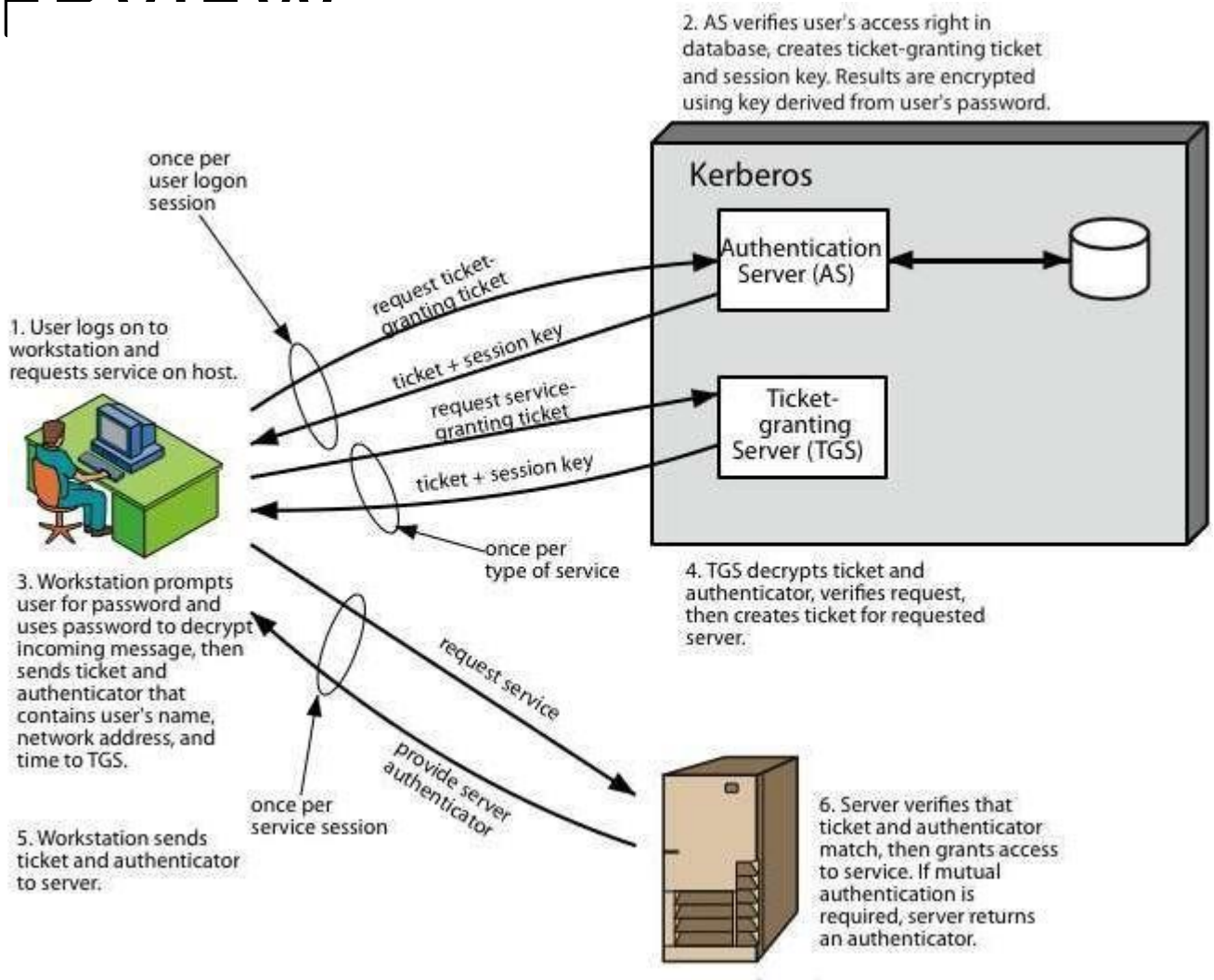


Fig . 4.14

KERBEROS REALMS

- A kerberos environment consists of:
  - A kerberos server
  - A number of clients, all registered with server
  - Application servers, sharing keys with server
- This is termed a realm

  Typically a single administrative domain
- If have multiple realms, their kerberos servers must share keys and trust

KERBEROS REALMS

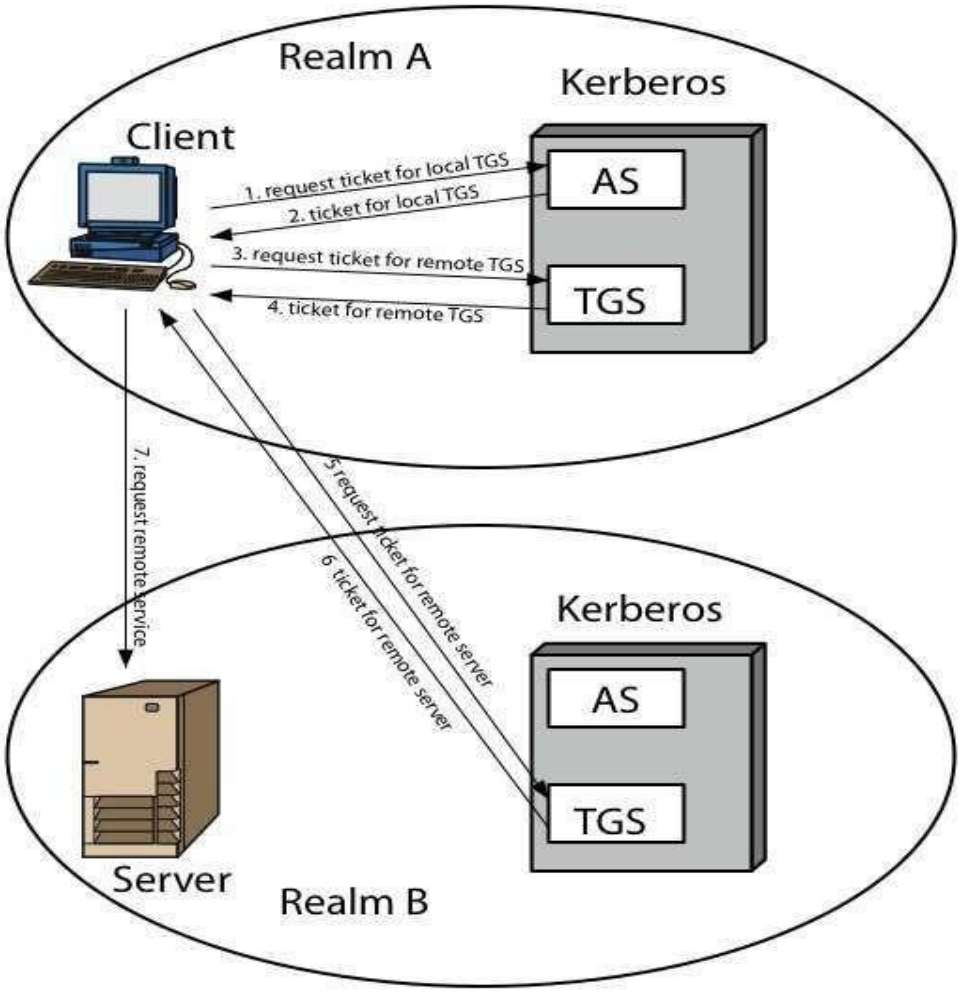


Fig . 4.15

KERBEROS VERSION 5

- Developed in mid 1990's
- Specified as internet standard RFC 1510
- Provides improvements over v4
  - Addresses environmental shortcomings

    Encryption alg, network protocol, byte order, ticket lifetime, authentication forwarding, interrealm auth
  - And technical deficiencies

    Double encryption, non-std mode of use, session keys, password attacks

V5

D'ALOGUE

(1) C → AS Options || ID_c || $Realm_c$ || ID_{tgs} || $Times$ || $Nonce_1$

(2) AS → C $Realm_c$ || ID_C || $Ticket_{tgs}$ || $E(K_c, [K_{c,tgs} || Times || Nonce_1 || Realm_{tgs} || ID_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [Flags || K_{c,tgs} || Realm_c || ID_C || AD_C || Times])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) C → TGS Options || ID_v || $Times$ || || $Nonce_2$ || $Ticket_{tgs}$ || $Authenticator_c$

(4) TGS → C $Realm_c$ || ID_C || $Ticket_v$ || $E(K_{c,tgs}, [K_{c,v} || Times || Nonce_2 || Realm_v || ID_v])$

$Ticket_{tgs} = E(K_{tgs}, [Flags || K_{c,tgs} || Realm_c || ID_C || AD_C || Times])$

$Ticket_v = E(K_v, [Flags || K_{c,v} || Realm_c || ID_C || AD_C || Times])$

$Authenticator_c = E(K_{c,tgs}, [ID_C || Realm_c || TS_1])$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) C → V Options || $Ticket_v$ || $Authenticator_c$

(6) V → C $E_{K_{C,V}}$ [TS_2 || Subkey || Seq#]

$Ticket_v = E(K_v, [Flags || K_{c,v} || Realm_c || ID_C || AD_C || Times])$

$Authenticator_c = E(K_{c,v}, [ID_C || Realm_c || TS_2 || Subkey || Seq\#])$

(c) Client/Server Authentication Exchange to obtain service

Fig . 4.16

AUTHENTICATION

- In ch 14 saw use of public-key encryption for session key distribution
  - Assumes both parties have other's public keys
  - May not be practical
- Have denning protocol using timestamps
  - Uses central authentication server (AS) to provide public-key certificates
  - Requires synchronized clocks
- Have woo and lam protocol using nonces
- Care needed to ensure no protocol flaws

AUTHENTICATION

- Have public-key approaches for email
  - Encryption of message for confidentiality, authentication, or both
  - Must now public keys
  - Using costly public-key alg on long message
- For confidentiality encrypt message with one-time secret key, public-key encrypted
- For authentication use a digital signature
  - May need to protect by encrypting signature
- Use digital certificate to supply public key

IDENTITY MANAGEMENT

- Use of common identity management scheme
  - Across multiple enterprises & numerous applications
  - Supporting many thousands, even millions of users
- Principal elements are:
  - Authentication, authorization, accounting, provisioning, workflow automation, delegated administration, password synchronization, self-service password reset, federation
- Kerberos contains many of these elements

MANAGEMENT

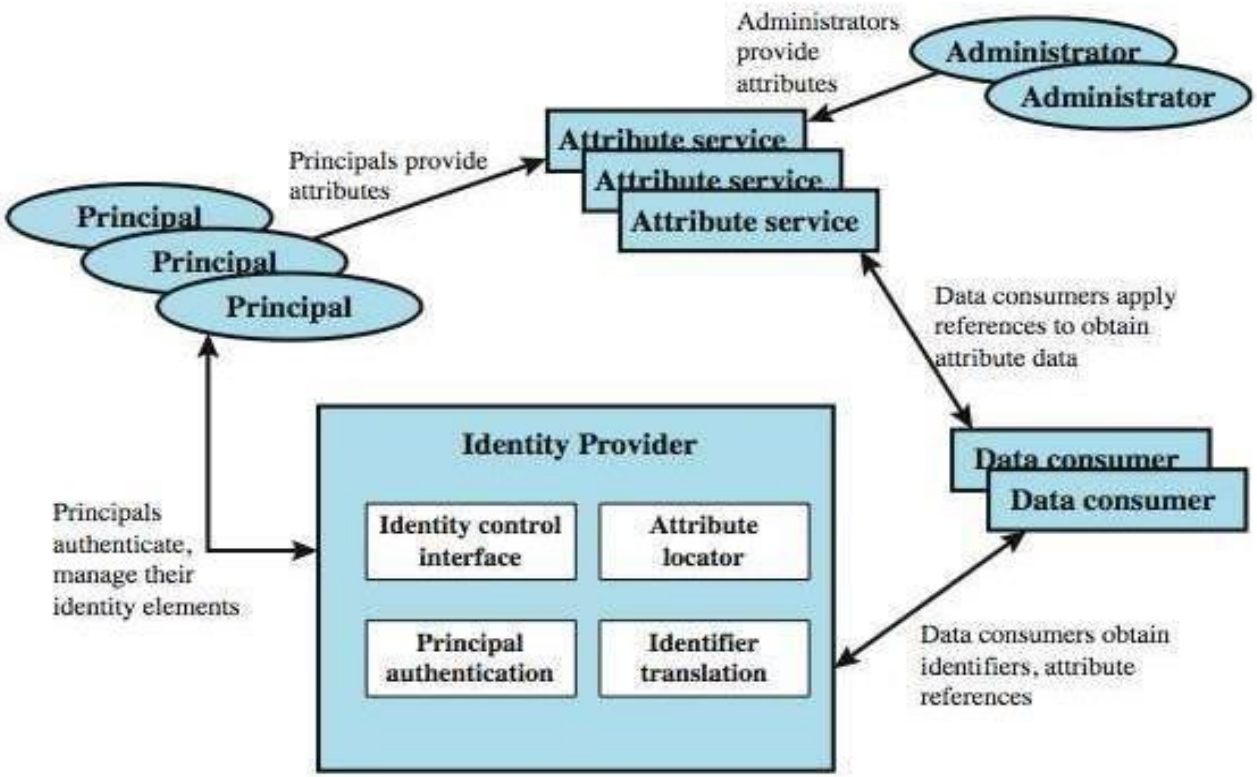
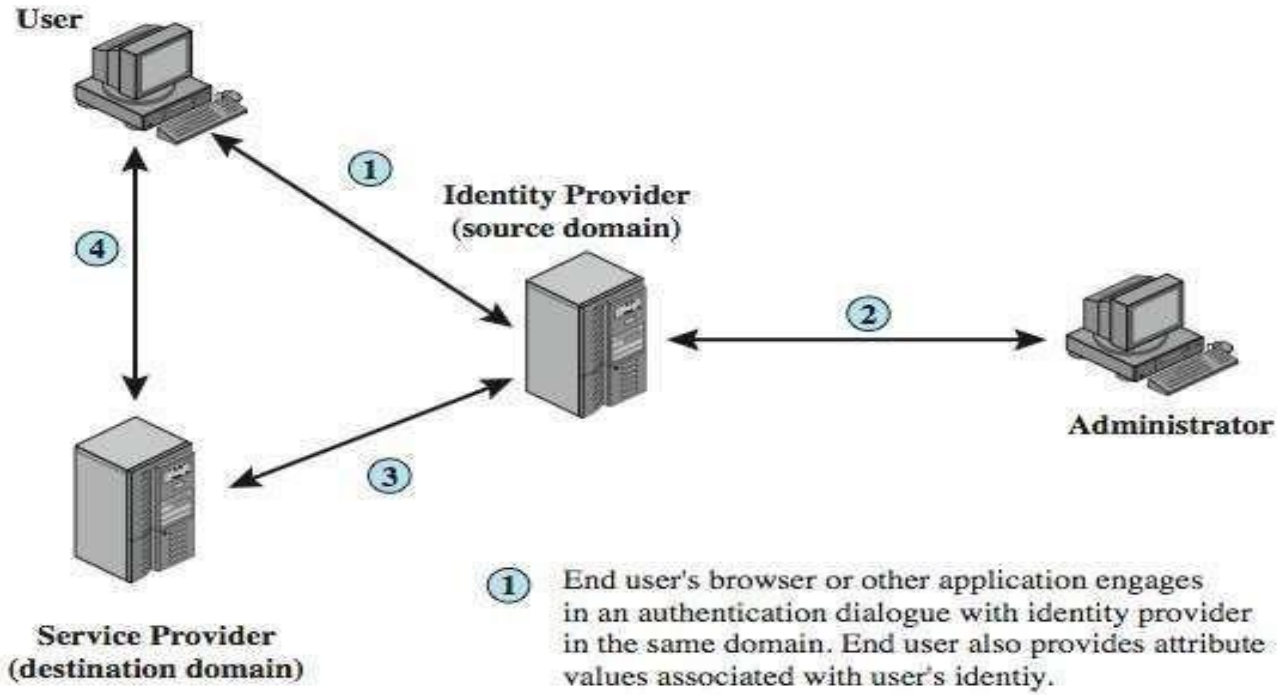


Fig . 4.17

FEDERATIO N



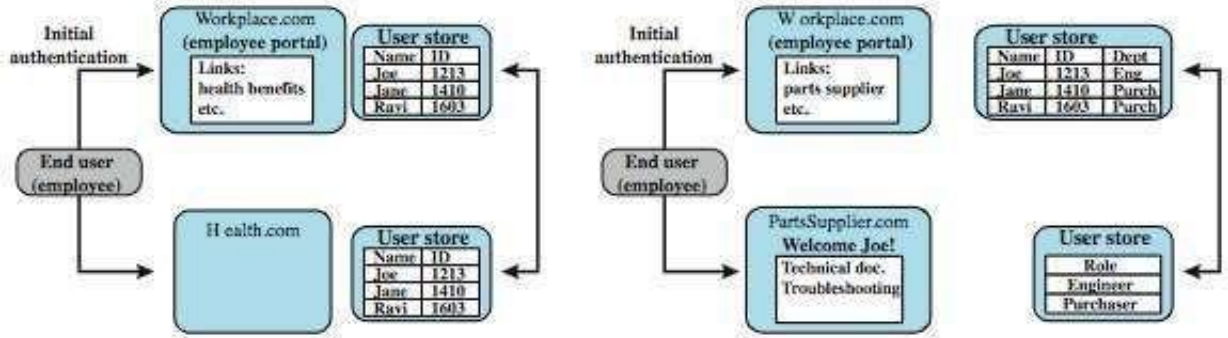
1. End user's browser or other application engages in an authentication dialogue with identity provider in the same domain. End user also provides attribute values associated with user's identiy.
2. Some attributes associated with an identity, such as allowable roles, may be provided by an administrator in the same domain.
3. A service provider in a remote domain, which the user wishes to access, obtains identity information, authentication information, and associated attributes from the identity provider in the source domain.
4. Service provider opens session with remote user and enforces access control restrictions based on user's identity and attributes.

Fig . 4.18

STANDARDS USED

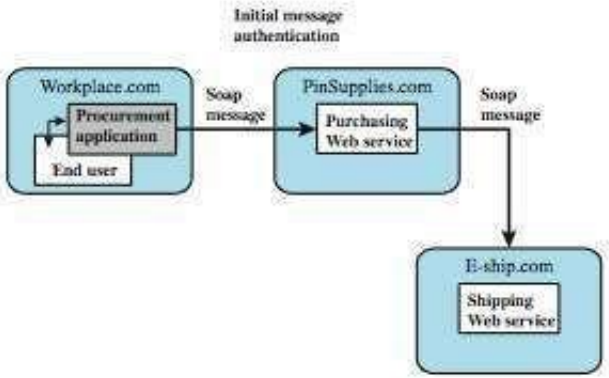
- Security assertion markup language (SAML)
  - Xml-based language for exchange of security information between online business partners
- Part of OASIS (organization for the advancement of structured information standards) standards for federated identity management
  - E.G. Ws-federation for browser-based federation
- Need a few mature industry standards

IDENTITY EXAMPLES



(a) Federation based on account linking

(b) Federation based on roles



(b) Chained Web Services

Fig . 4.19

EMAIL SECURITY ENHANCEMENTS

- Confidentiality

  Protection from disclosure

- Authentication

  Of sender of message

- Message integrity

  Protection from modification

- Non-repudiation of origin

  Protection from denial by sender

PRETTY GOOD PRIVACY (PGP)

- Widely used de facto secure email
- Developed by phil zimmermann
- Selected best available crypto algs to use
- Integrated into a single program
- Available on unix, PC, macintosh and amiga systems
- Originally free, now have commercial versions available also

PGP OPERATION – AUTHENTICATION

- Sender creates a message
- SHA-1 used to generate 160-bit hash code of message
- Hash code is encrypted with RSA using the sender's private key, and result is attached to message
- Receiver uses RSA with sender's public key to decrypt and recover hash code
- Receiver generates new hash code for message and compares with decrypted hash code, if match, message is accepted as authentic

PGP OPERATION – CONFIDENTIALITY

- Sender generates message and random 128-bit number to be used as session key for this message only
- Message is encrypted, using CAST-128 / IDEA/3DES with session key
- Session key is encrypted using RSA with recipient's public key, then attached to message
- Receiver uses RSA with its private key to decrypt and recover session key
- Session key is used to decrypt message

CONFIDENTIALITY & AUTHENTICATION

- Uses both services on same message
  - Create signature & attach to message
  - Encrypt both message & signature
  - Attach RSA encrypted session key

PGP OPERATION – COMPRESSION

By default PGP compresses message after signing but before encrypting

Uses ZIP compression algorithm

SESSION KEYS

- Need a session key for each message

  Of varying sizes: 56-bit DES, 128-bit CAST or IDEA, 168-bit triple-des

- Uses random inputs taken from previous uses and from keystroke timing of user

KEY RINGS

- Each PGP user has a pair of keyrings:
  - Public-key ring contains all the public-keys of other PGP users known to this user, indexed by key ID
  - Private-key ring contains the public/private key pair(s) for this user, indexed by key ID & encrypted keyed from a hashed passphrase

MANAGEM ENT

- Rather than relying on certificate authorities
- In PGP every user is own CA

  Can sign keys for users they know directly
- Forms a “web of trust”

(SECURE/MULTIPURPOSE INTERNET MAIL EXTENSIONS)

- Security enhancement to MIME email
  - Original internet RFC822 email was text only
  - MIME provided support for varying content types and multi-part messages

    Image, video, audio, PS, octet-stream
  - S/MIME added security enhancements
- Have S/MIME support in various modern mail agents: MS outlook, netscape etc

FUNCTIONS

- Enveloped data

  Encrypted content and associated keys

- Signed data

  Encoded message + signed digest

- Clear-signed data

  Cleartext message + encoded signed digest

- Signed & enveloped data

  Nesting of signed & encrypted entities

CRYPTOGRAPHIC ALGORITHMS

- Hash functions: SHA-1 & MD5
- Digital signatures: DSS & RSA
- Session key encryption: D-H & RSA
- Message encryption: triple-des, RC2/40 and others
- Have a procedure to decide which algorithms to use
  - According to the capability of the receiving agent

CONTENT BEYOND SYLLABUS

Secure Electronic Transactions (SET)

- Protocol- to protect Internet credit card transactions
- developed in 1996 by Mastercard, Visa etc
- not a payment system
- rather a set of security protocols & formats
  - secure communications amongst parties
  - trust from use of X.509v3 certificates
  - privacy by restricted info to those who need it

COMPONENTS

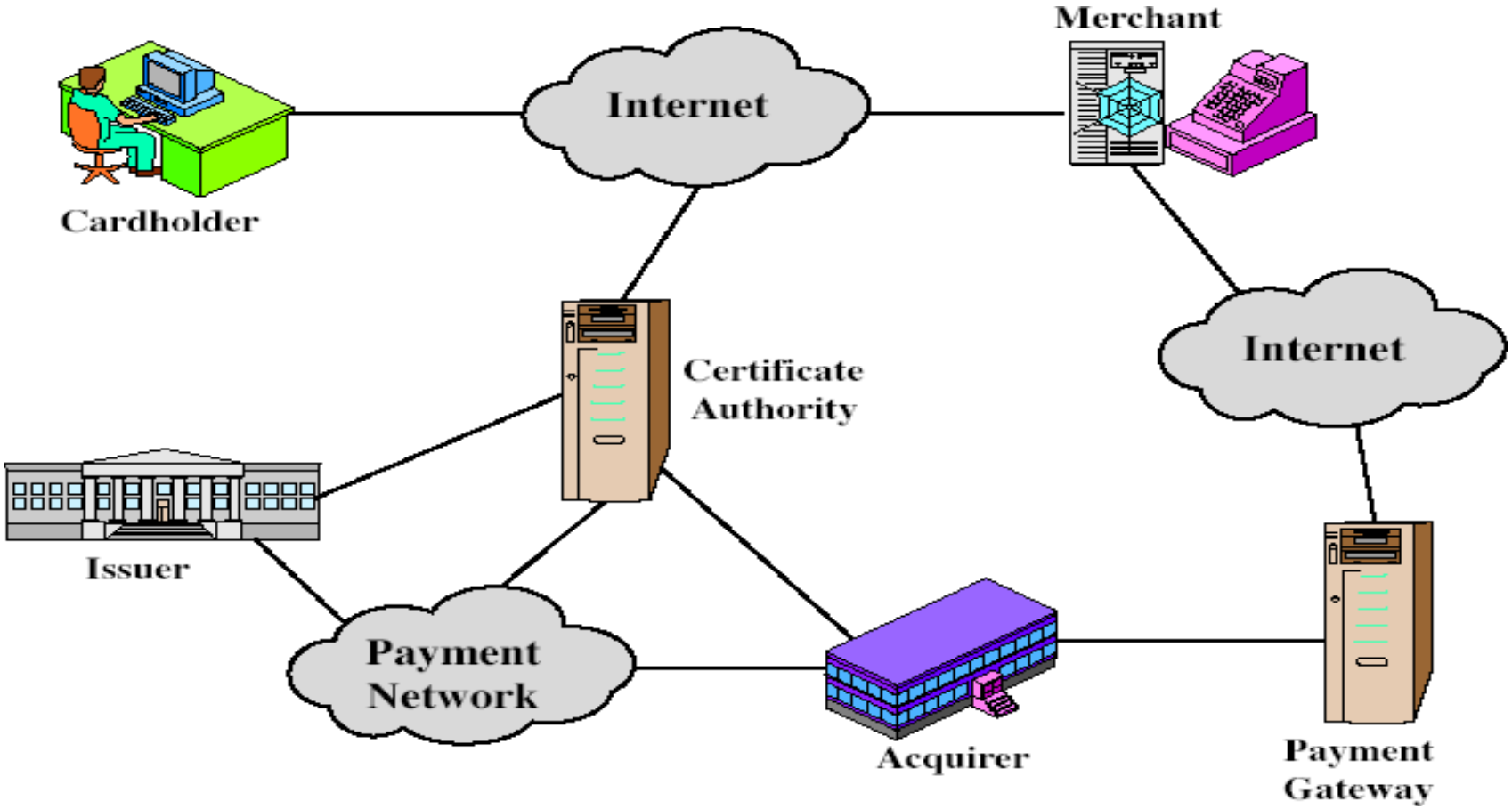


Fig . 4.20

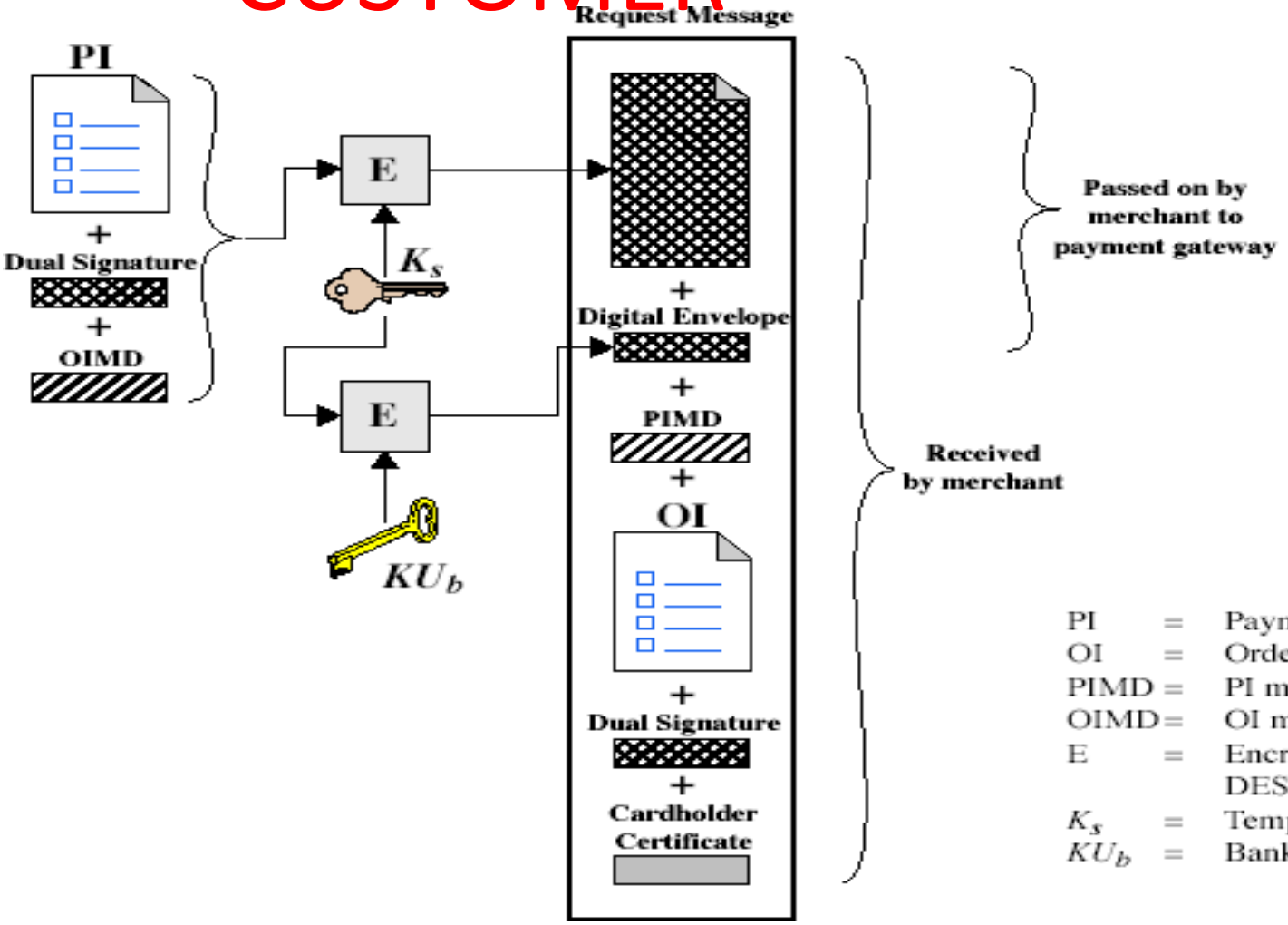
TRANSACTION

1. customer opens account
2. customer receives a certificate
3. merchants have their own certificates
4. customer places an order
5. merchant is verified
6. order and payment are sent
7. merchant requests payment authorization
8. merchant confirms order
9. merchant provides goods or service
10. merchant requests payment

SIGNATURE

- customer creates dual messages
  - order information (OI) for merchant
  - payment information (PI) for bank
- neither party needs details of other
- but **must** know they are linked
- use a dual signature for this
  - Signed(by encryption) and concatenated hashes of OI & PI

REQUEST – CUSTOMER



PI = Payment Information
OI = Order Information
PIMD = PI message digest
OIMD = OI message digest
E = Encryption (RSA for asymmetric; DES for symmetric)
K_s = Temporary symmetric key
KU_b = Bank's public key-exchange key

Fig . 4.21

REQUEST – MERCHANT

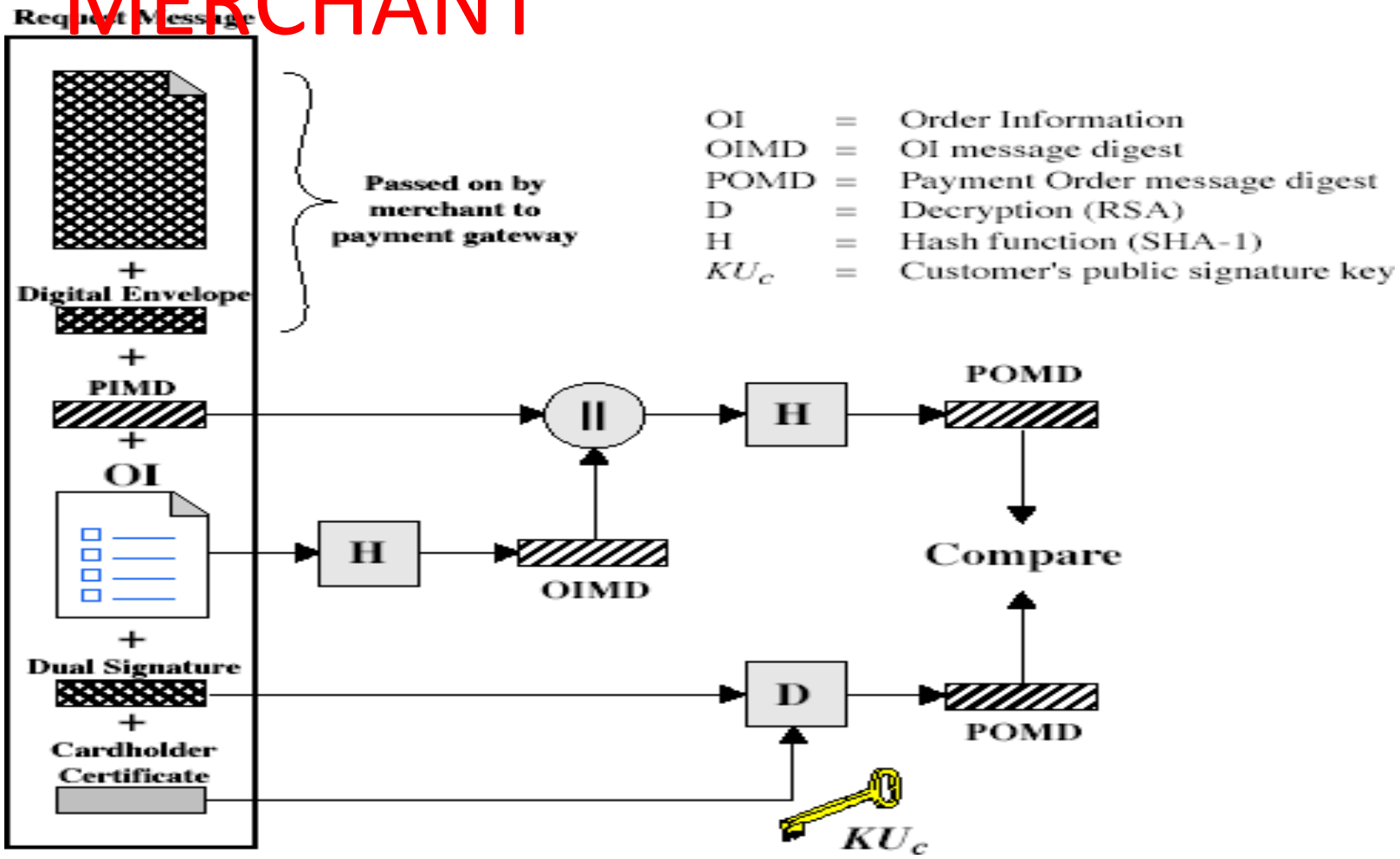


Fig . 4.22

REQUEST – MERCHANT

1. verifies cardholder certificates using CA signs
2. verifies dual signature using customer's public signature key to ensure order has not been tampered with in transit & that it was signed using cardholder's private signature key
3. andforwards the payment information to the payment gateway for authorization (described later)
4. sends a purchase response to cardholder

GATEWAY AUTHORIZATION

1. verifies all certificates
2. decrypts digital envelope of authorization block to obtain symmetric key & then decrypts authorization block
3. verifies merchant's signature on authorization block
4. decrypts digital envelope of payment block to obtain symmetric key & then decrypts payment block
5. verifies dual signature on payment block
6. verifies that transaction ID received from merchant matches that in PI received (indirectly) from customer
7. requests & receives an authorization from issuer
8. sends authorization response back to merchant

PAYMENT CAPTURE

- Merchant sends payment gateway a payment capture request
- gateway checks request
- Then causes funds to be transferred to merchants account
- Notifies merchant using capture response

RESOURCES

- **Lecture Notes - Lecture Notes**
- **Video Lectures - Video Lecture**
- **E-Book - Information Security Concepts**
- **Model Papers - JNTUA Question Papers**

SVEC TIRUPATI

EDUCATION FOR A BETTER SOCIETY

DEPT & SEM : CSE-CS & I SEM

SUBJECTNAME : **INFORMATION SECURITY**

COURSE CODE : **IS**

UNIT : **V**

PREPARED BY : **ANUSHA K**

TLI
NE

- **SECURITY AT THE TRANSPORT LAYER(SSL AND TLS)**
- **SSL ARCHITECTURE**
- **FOUR PROTOCOLS**
- **SSL MESSAGE FORMATS**
- **TRANSPORT LAYER SECURITY**
- **HTTPS**
- **SSH**

SSL (SECURE SOCKET LAYER)

- Transport layer security service
- Originally developed by Netscape
- Version 3 designed with public input
- Subsequently became internet standard known as TLS (transport layer security)
- Uses TCP to provide a reliable end-to-end service
- SSL has two layers of protocols

SSL ARCHITECTURE

- **SSL CONNECTION**
  - A transient, peer-to-peer, communications li n k
  - Associated with 1 SSL session
- **SSL SESSION**
  - An association between client & server
  - Created by the handshake protocol
  - Define a set of cryptographic parameters
  - May be shared by multiple SSL connections

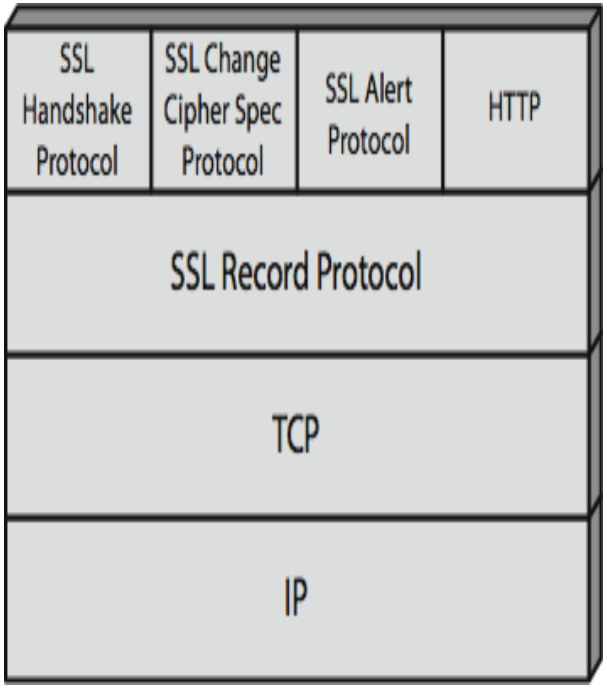


Fig . 5.1

SSL RECORD PROTOCOL SERVICES

- **CONFIDENTIALITY**

  - Using symmetric encryption with a shared secret key defined by handshake protocol

  - AES, IDEA, RC2-40, DES-40, DES, 3DES, fortezza, RC4-40, RC4-128

  - Message is compressed before encryption

- **MESSAGE INTEGRITY**

  - Using a MAC with shared secret key

  - Similar to HMAC but with different padding

SSL RECORD PROTOCOL OPERATION

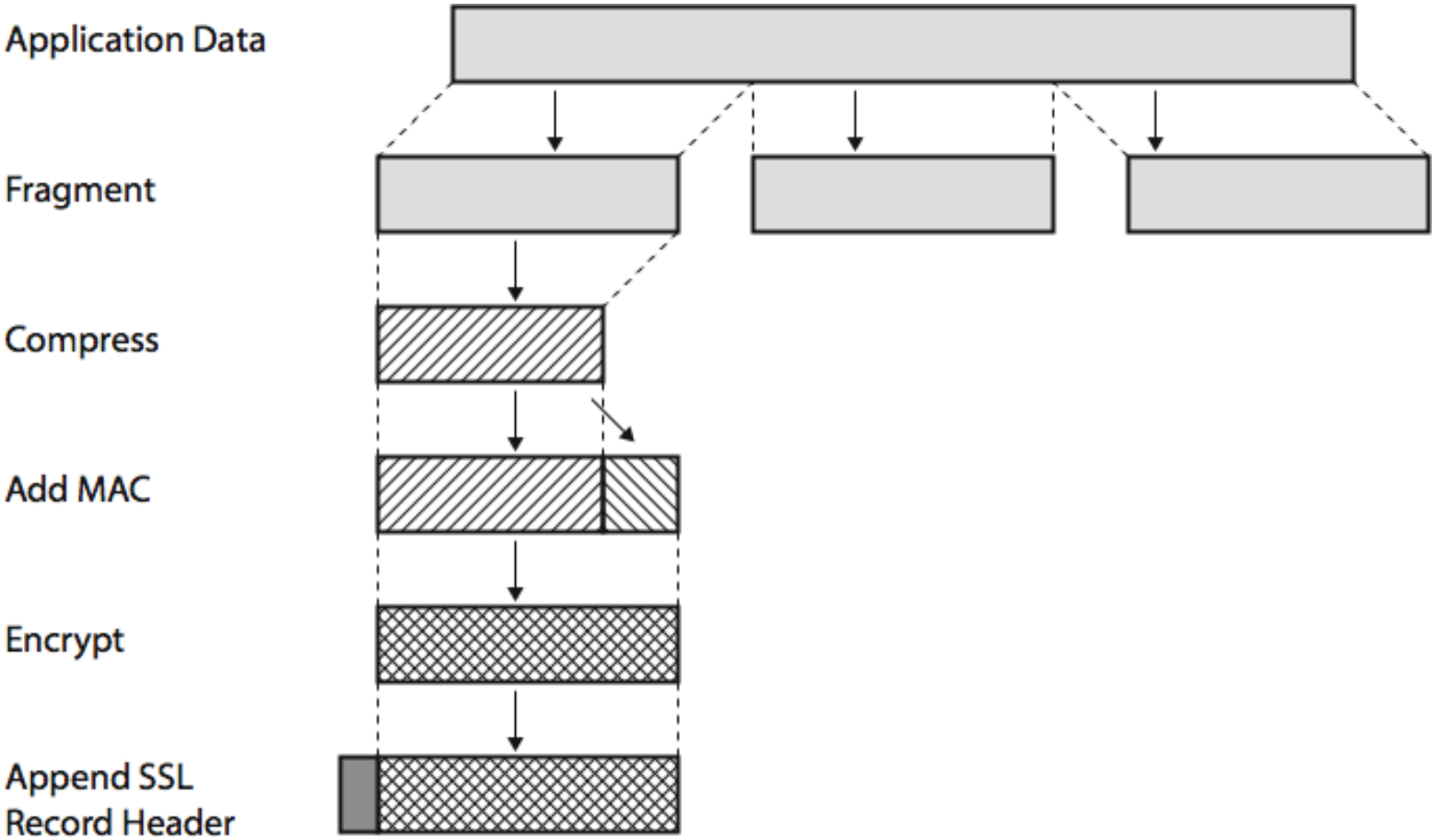
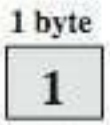


Fig . 5.2

SSL CHANGE CIPHER SPEC PROTOCOL

- One of 3 SSL specific protocols which use the SSL record protocol
- A single message
- Causes pending state to become current
- Hence updating the cipher suite in use

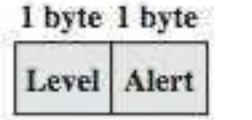


(a) Change Cipher Spec Protocol

Fig . 5.3

SSL ALERT PROTOCOL

- **Conveys SSL-related alerts to peer entity**
- **SEVERITY**
  - Warning or fatal
- **SPECIFIC ALERT**
  - Fatal: unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
  - Warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- **Compressed & encrypted like all SSL data**



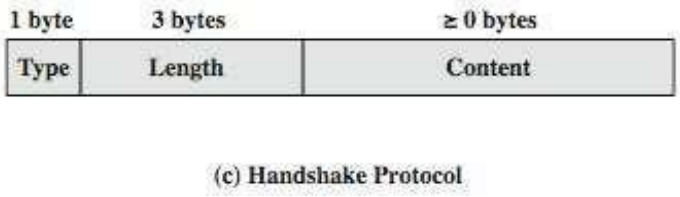
(b) Alert Protocol

Fig . 5.4

SSL HANDSHAKE PROTOCOL

ALLOWS SERVER & CLIENT TO

- Authenticate each other
- To negotiate encryption & MAC algorithms
- To negotiate cryptographic keys to be used



(c) Handshake Protocol

Fig . 5.5

COMPRISES A SERIES OF MESSAGES IN PHASES

- Establish security capabilities
- Server authentication and key exchange
- Client authentication and key exchange
- Finish

SSL HANDSHAKE PROTOCOL

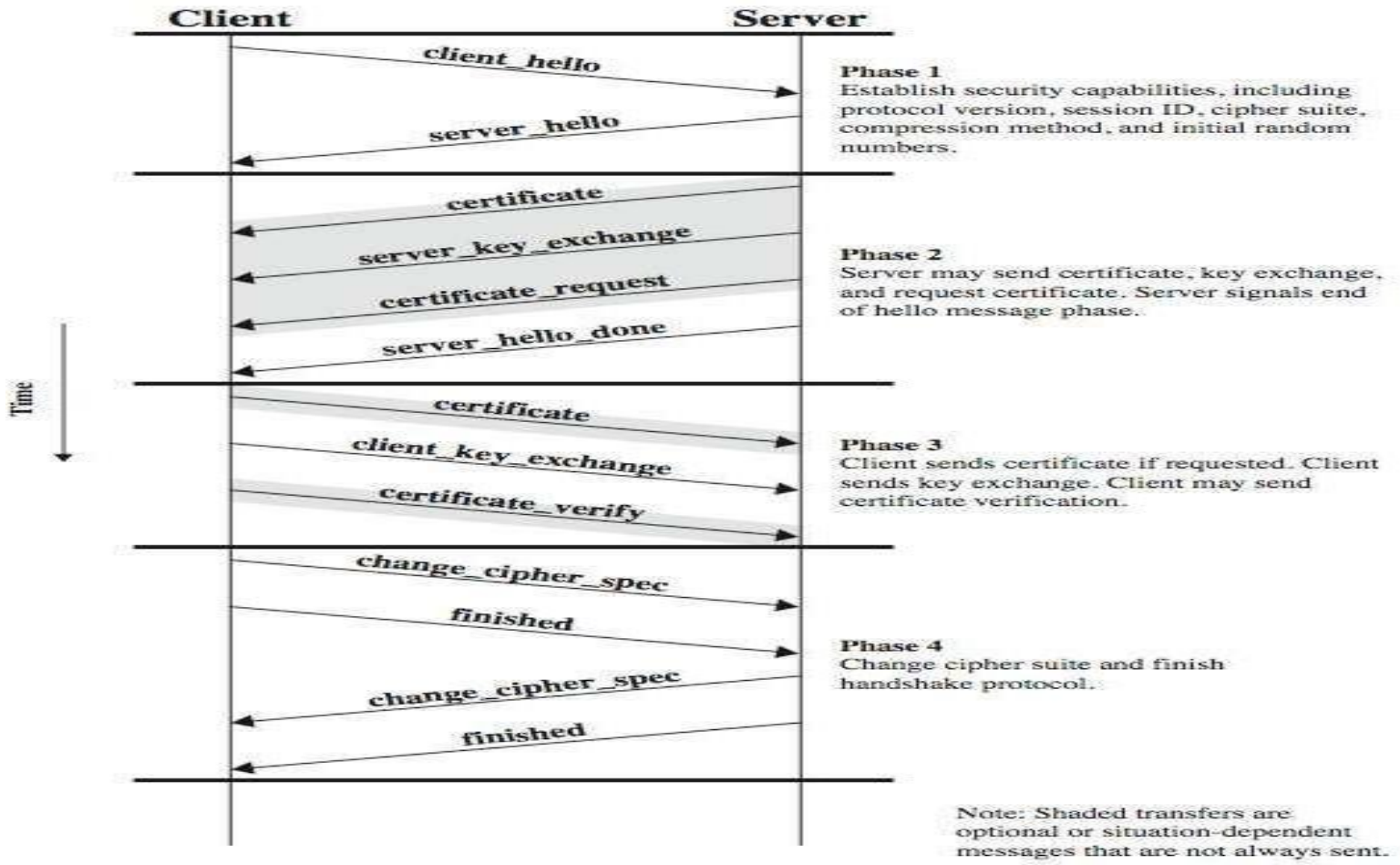


Fig . 5.6

CRYPTOGRAPHIC COMPUTATIONS

- **MASTER SECRET CREATION**
  - A one-time 48-byte value
  - Generated using secure key exchange (RSA / diffie-hellman) and then hashing info
- **GENERATION OF CRYPTOGRAPHIC PARAMETERS**
  - Client write MAC secret, a server write MAC secret, a client write key, a server write key, a client write IV, and a server write IV
  - Generated by hashing master secret

TLS (Transport Layer Security)

- **IETF standard RFC 2246 similar to sslv3**
- **<u>WITH MINOR DIFFERENCES</u>**
  - In record format version number
  - Uses HMAC for MAC
  - A pseudo-random function expands secrets
- **Based on HMAC using SHA-1 or MD5**
  - Has additional alert codes
  - Some changes in supported ciphers
  - Changes in certificate types & negotiations
  - Changes in crypto computations & padding

HTTPS

- **HTTPS (HTTP OVER SSL)**
  - Combination of HTTP & SSL/TLS to secure communications between browser & server
    - Documented in RFC2818
    - No fundamental change using either SSL or TLS
- **USE HTTPS:// URL RATHER THAN HTTP://**
  - And port 443 rather than 80
- **ENCRYPTS**
  - URL, document contents, form data, cookies, HTTP headers

HTTPS USE

- **CONNECTION INITIATION**
  - TLS handshake then HTTP request(s)
- **CONNECTION CLOSURE**
  - Have “connection: close” in HTTP record
  - TLS level exchange close_notify alerts
  - Can then close TCP connection
  - Must handle TCP close before alert exchange sent or completed

SECURE SHELL (SSH)

- **PROTOCOL FOR SECURE NETWORK COMMUNICATIONS**
  - Designed to be simple & inexpensive
- **SSH1 PROVIDED SECURE REMOTE LOGON FACILITY**
  - Replace TELNET & other insecure schemes
  - Also has more general client/server capability
- SSH2 fixes a number of security flaws
- Documented in rfcs 4250 through 4254
- SSH clients & servers are widely available
- Method of choice for remote login/ X tunnels

SSH PROTOCOL STACK

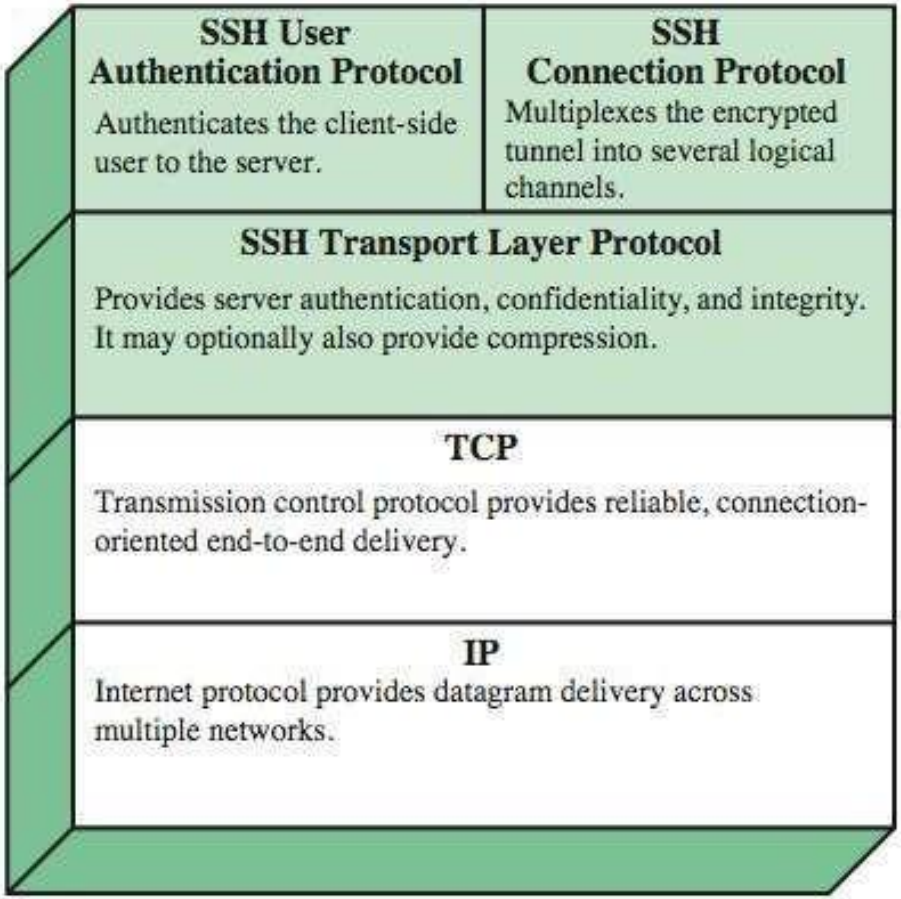


Fig . 5.7

SSH TRANSPORT LAYER PROTOCOL

- **SERVER AUTHENTICATION OCCURS AT TRANSPORT LAYER, BASED ON SERVER/HOST KEY PAIR(S)**
  - Server authentication requires clients to know host keys in advance
- **Packet exchange**
  - Establish TCP connection
  - Can then exchange data
  - Identification string exchange, algorithm negotiation, key exchange, end of key exchange, service request
- Using specified packet format

SSH USER AUTHENTICATION PROTOCOL

- Authenticates client to server
- **THREE MESSAGE TYPES**
  - Ssh_msg_userauth_request
  - Ssh_msg_userauth_failure
  - Ssh_msg_userauth_success
- Authentication methods used
  - Public-key, password, host-based

SSH CONNECTION PROTOCOL

- Runs on SSH transport layer protocol
- Assumes secure authentication connection
- Used for multiple logical channels
  - SSH communications use separate channels
  - Either side can open with unique id number
  - Flow controlled
  - Have three stages:
    - Opening a channel, data transfer, closing a channel
  - Four types:
    - Session, x11, forwarded-tcpip, direct-tcpip.

SSH Connection Protocol Exchange

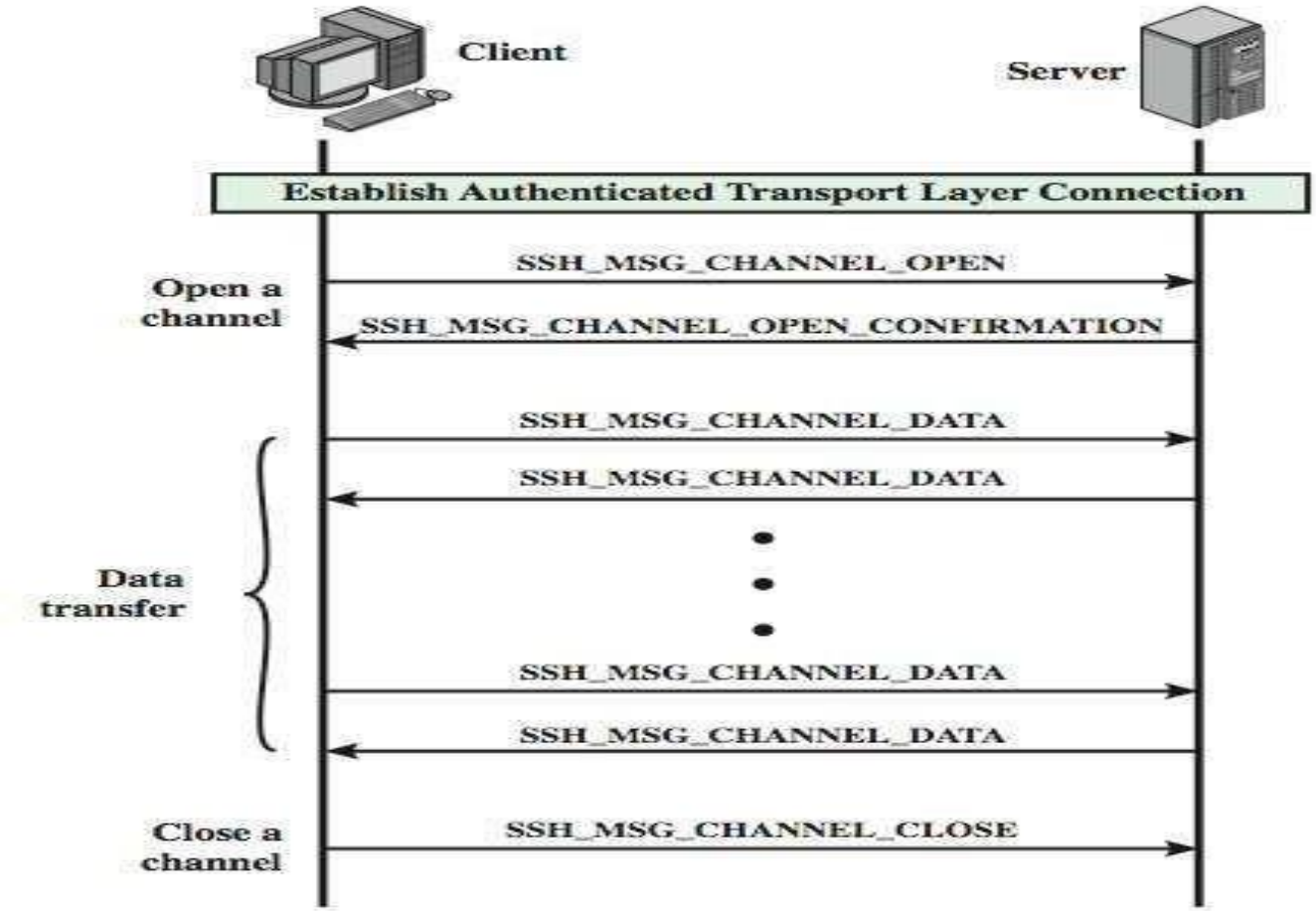


Fig . 5.8

PORT FORWARDING

- **CONVERT INSECURE TCP CONNECTION INTO A SECURE SSH CONNECTION**
  - SSH transport layer protocol establishes a TCP connection between SSH client & server
  - Client traffic redirected to local SSH, travels via tunnel, then remote SSH delivers to server
- **SUPPORTS TWO TYPES OF PORT FORWARDING**
  - Local forwarding – hijacks selected traffic
  - Remote forwarding – client acts for server

CHAPTER-2

SECURITY AT THE NETWORK LAYER (IPSEC)

TLI
NE

- **TWO MODES**
- **TWO SECURITY PROTOCOLS**
- **SECURITY ASSOCIATION**
- **SECURITY POLICY**
- **INTERNET KEY EXCHANGE**

IPSECURITY (IPSEC)

IP Security (IP Sec) is a collection of protocols designed by the Internet Engineering Task Force (IETF) to provide security for a packet at the network level.

Topics discussed in this section:

- Two Modes
- Two Security Protocols
- Security Association
- Internet Key Exchange (IKE)
- Virtual Private Network

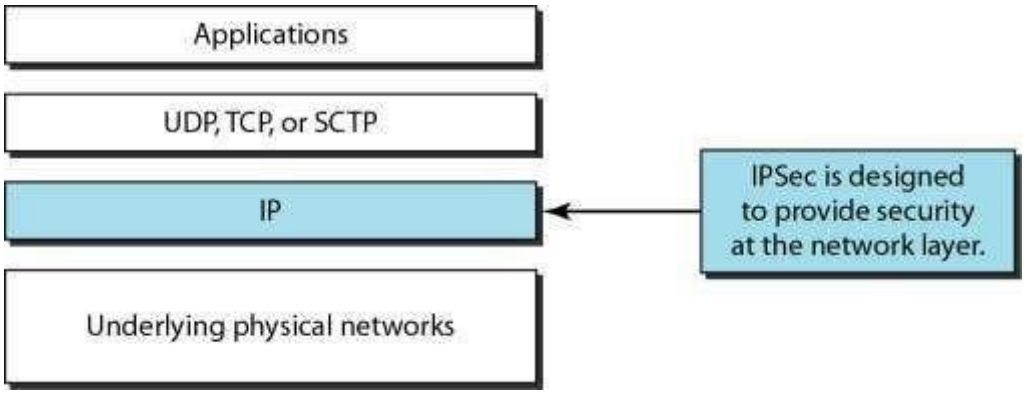


Fig . 5.9

TWO MODES

1. Transport Mode

2. Tunnel Mode

IP Sec in the transport mode does not protect the IP header; it only protects the information coming from the transport layer.

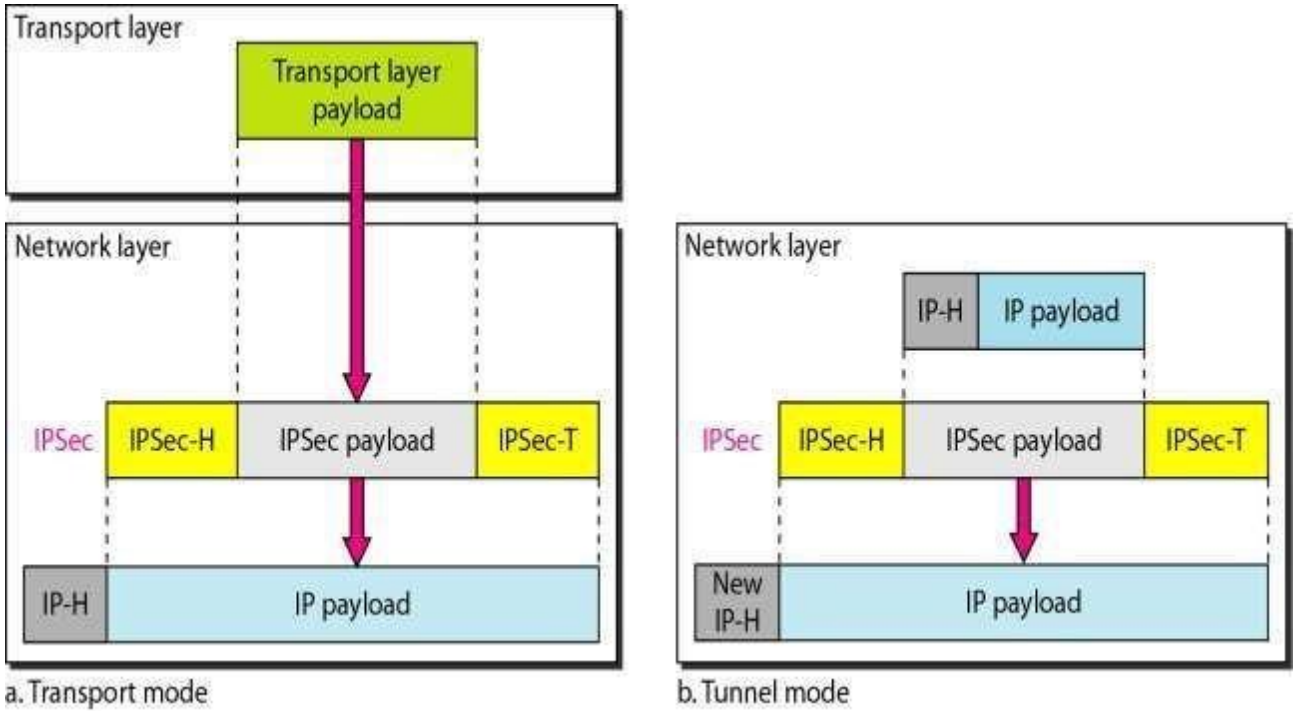


Fig . 5.10 Transport mode and tunnel modes of IPSec protocol

TWO MODES

- IPSec in tunnel mode protects the original IP header.

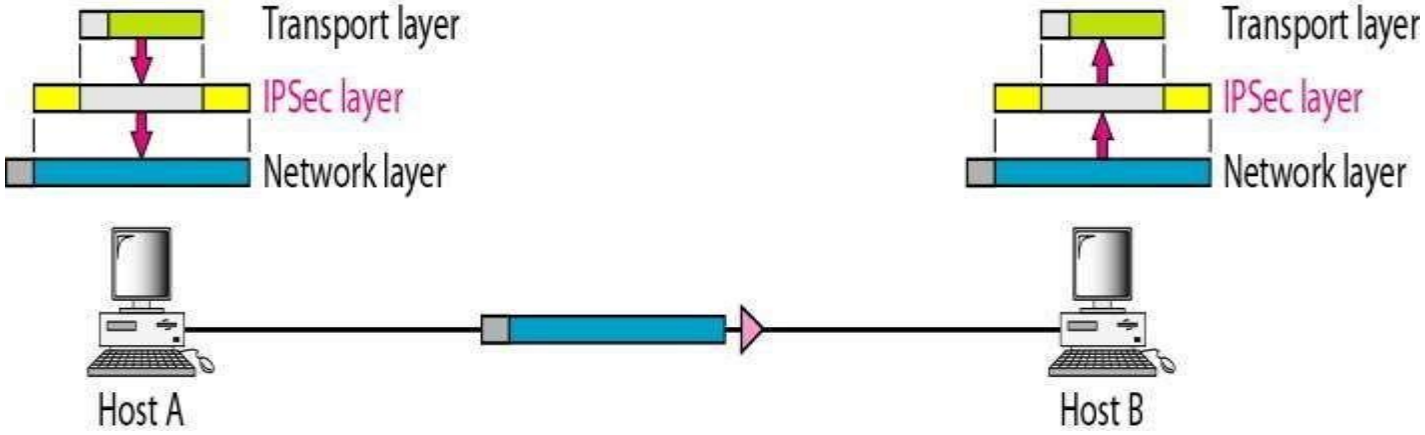


Fig . 5.11 Transport mode in action

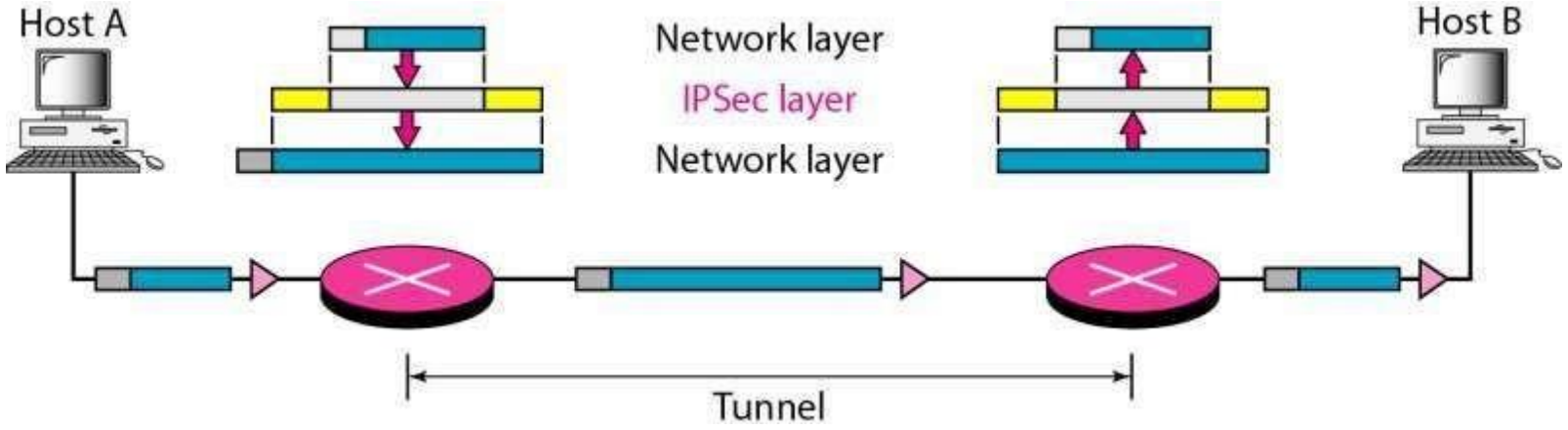


Fig . 5.12 Tunnel mode in action

TWO SECURITY PROTOCOLS

- The AH Protocol provides source authentication and data integrity, but not privacy.

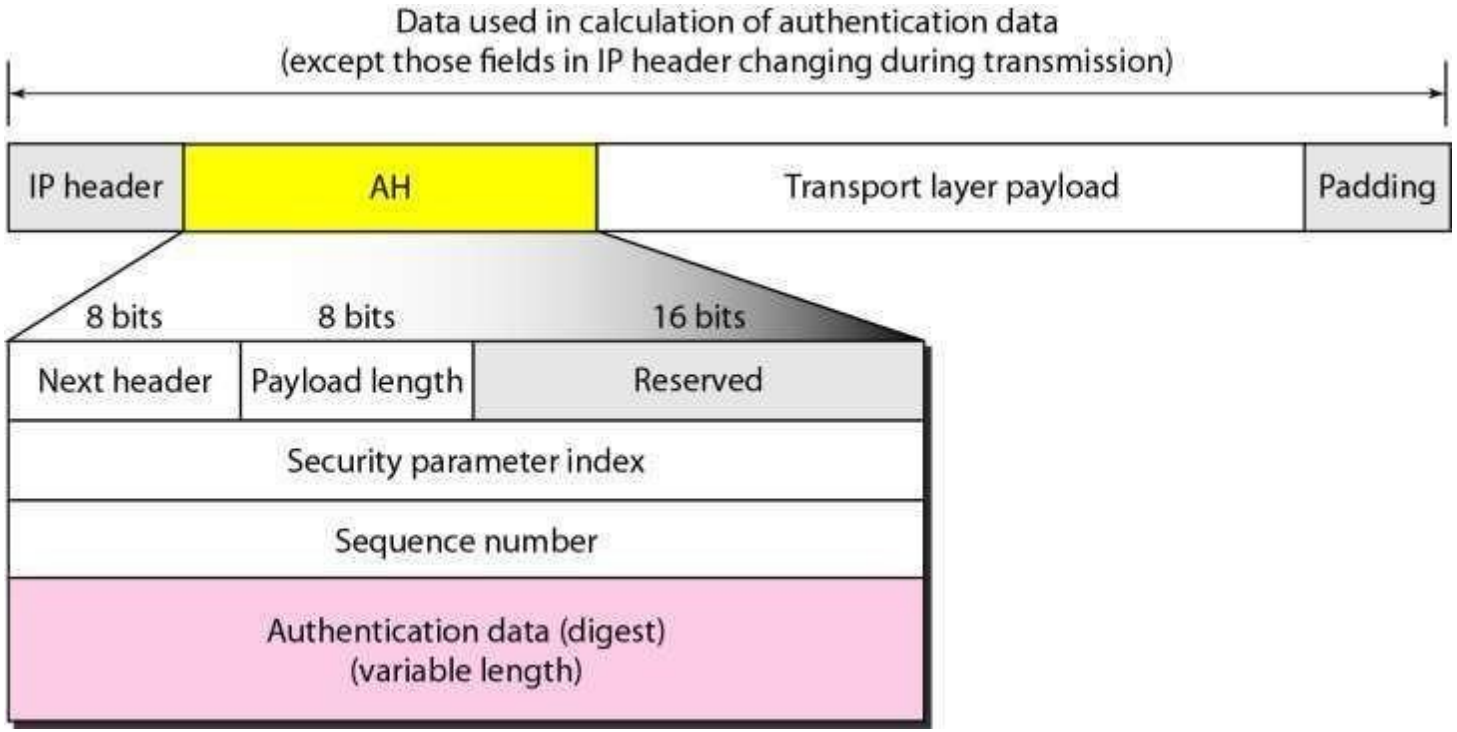


Fig . 5.13 Authentication Header (AH) Protocol in transport mode

TWO SECURITY PROTOCOLS

- ESP provides source authentication, data integrity, and privacy.

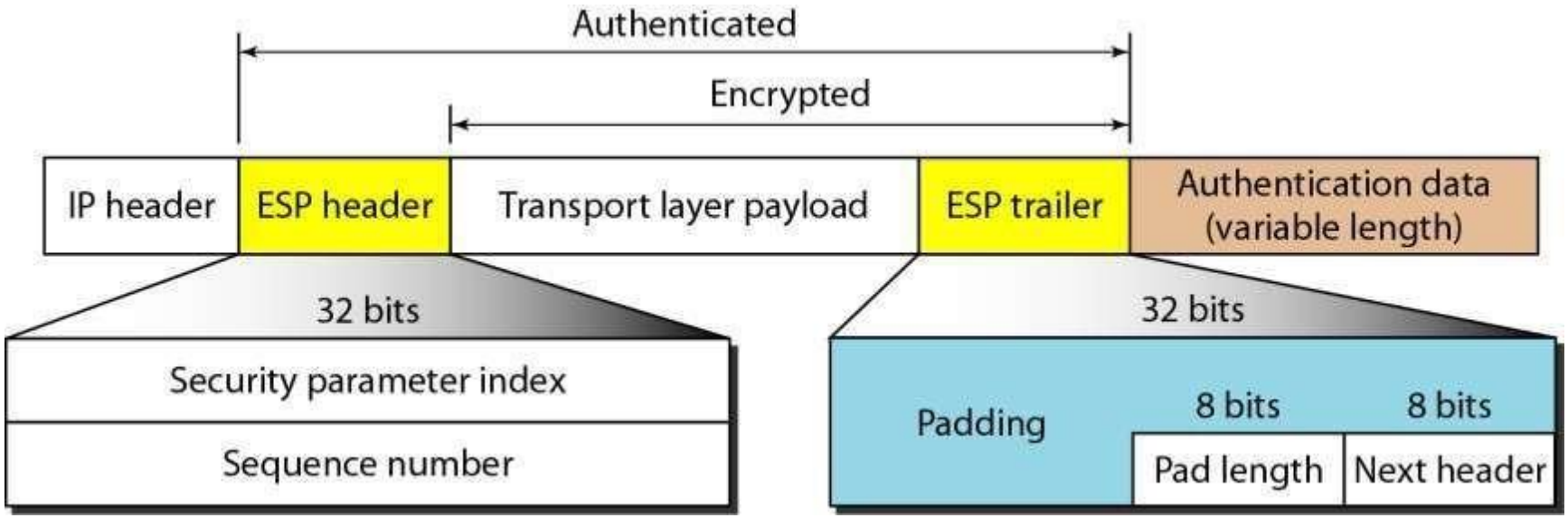


Fig . 5.14 *Encapsulating Security Payload (ESP) Protocol in transport mode*

TWO SECURITY PROTOCOLS

Table 5.1: IPSec services

Services	*AH*	*ESP*
Access control	Yes	Yes
Message authentication (message integrity)	Yes	Yes
Entity authentication (data source authentication)	Yes	Yes
Confidentiality	No	Yes
Replay attack protection	Yes	Yes

SECURITY ASSOCIATION

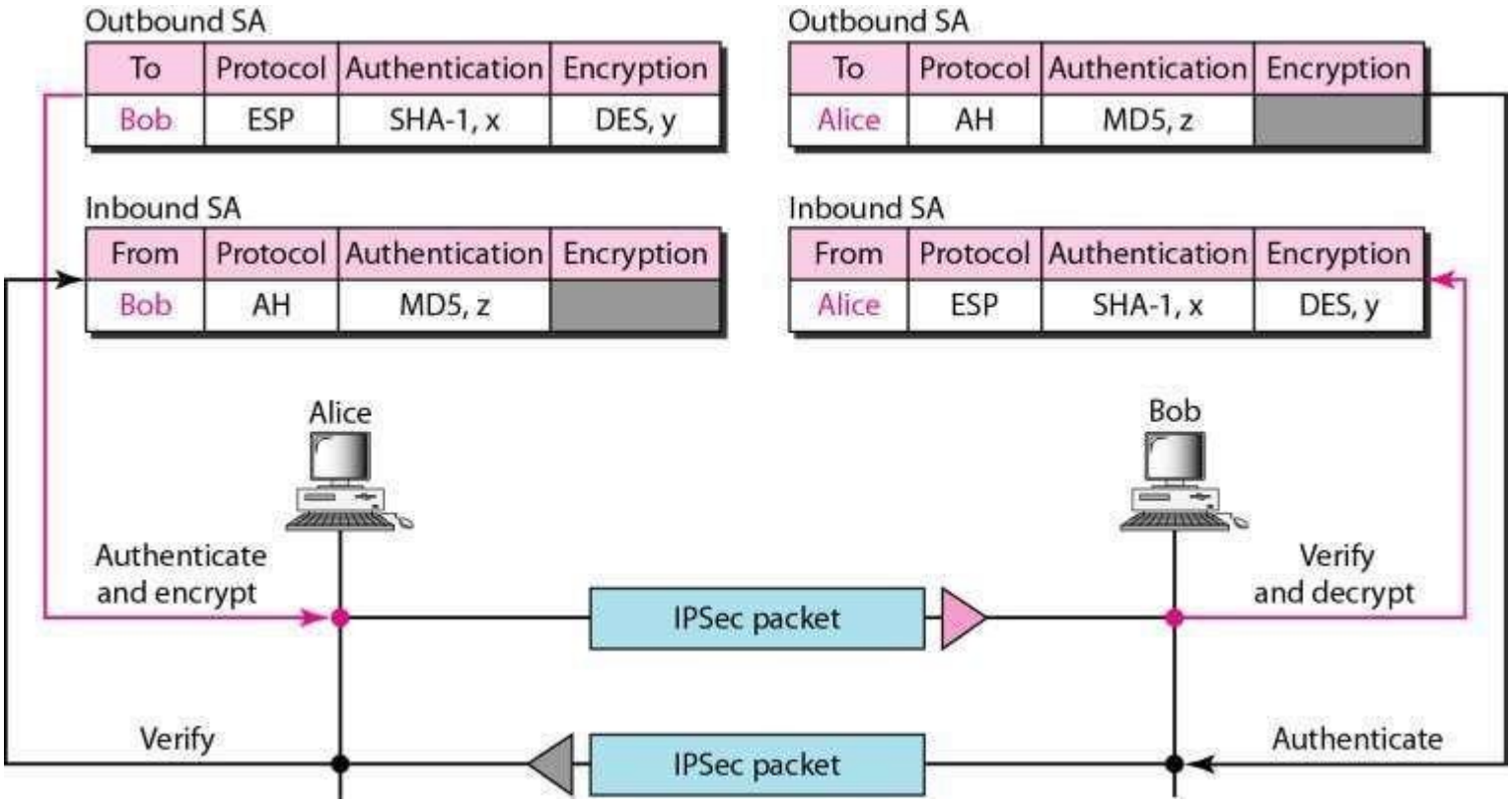


Fig . 5.15 Simple inbound and outbound security associations

INTERNET KEY EXCHANGE(IKE)

- IKE creates SAs for IP Sec.

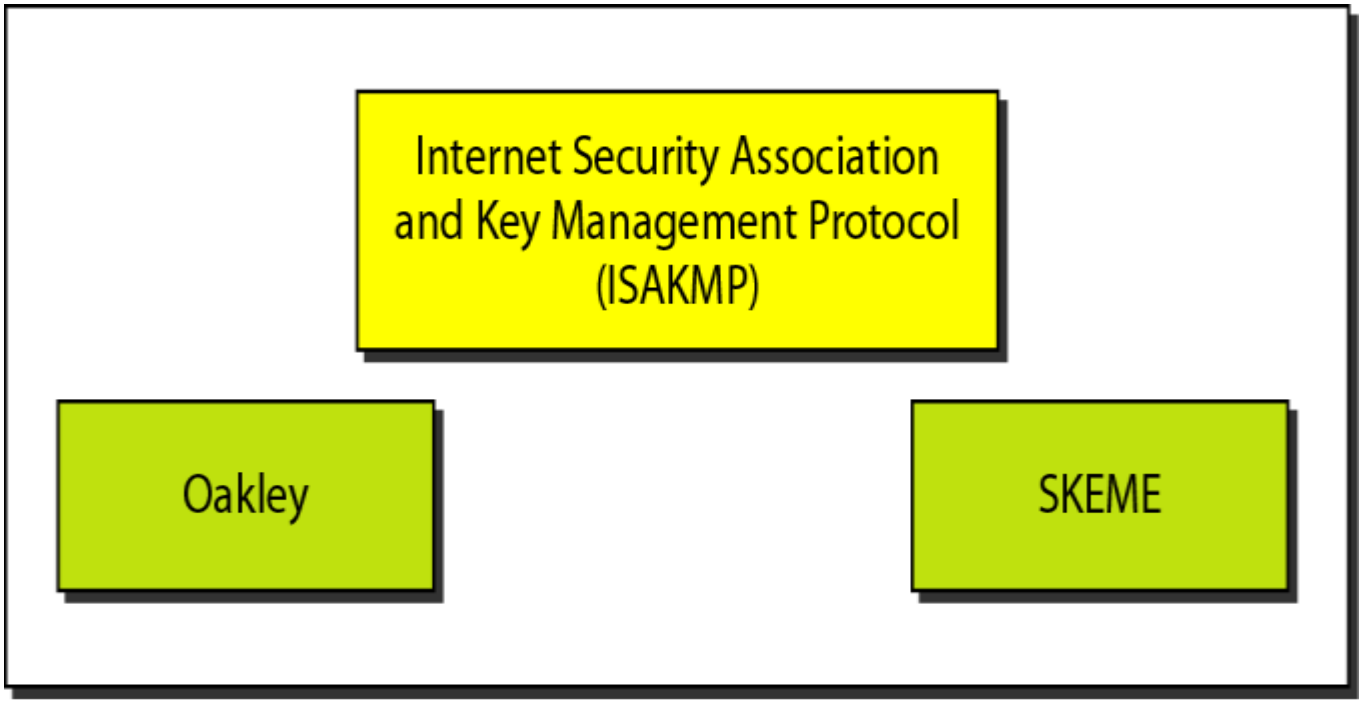


Fig . 5.16

INTERNET KEY EXCHANGE(IKE)

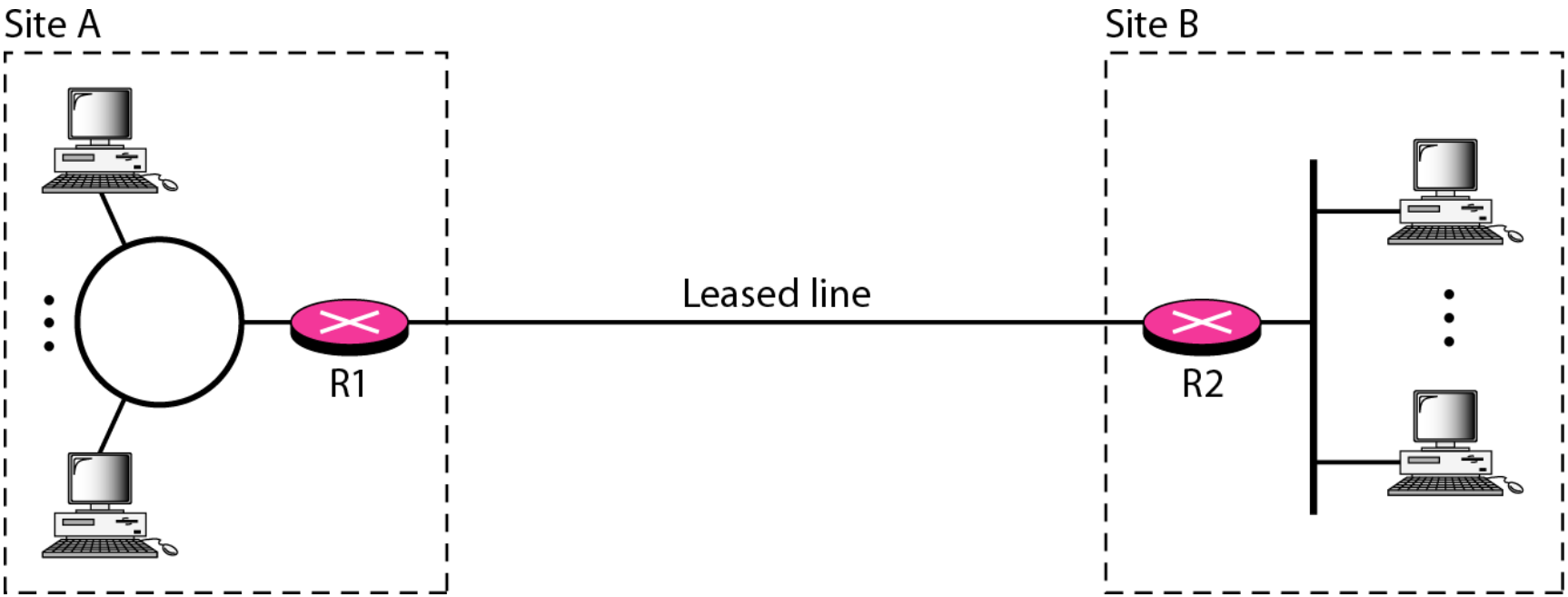


Fig . 5.17 Private network

INTERNET KEY EXCHANGE(IKE)

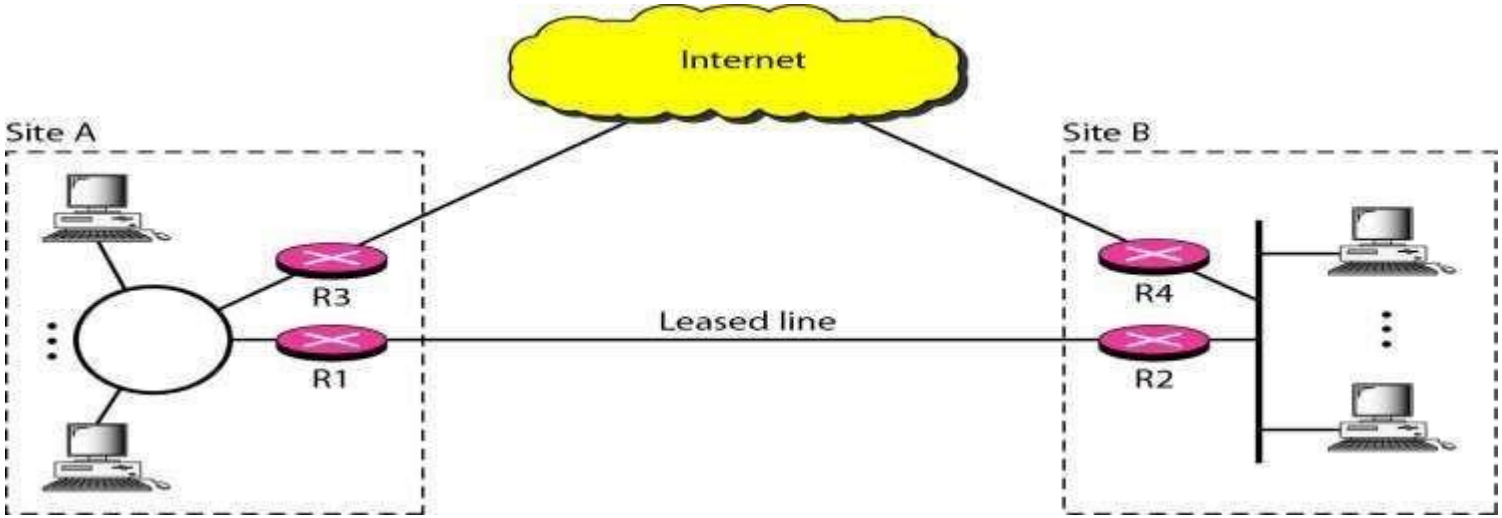


Fig . 5.18 Virtual private network

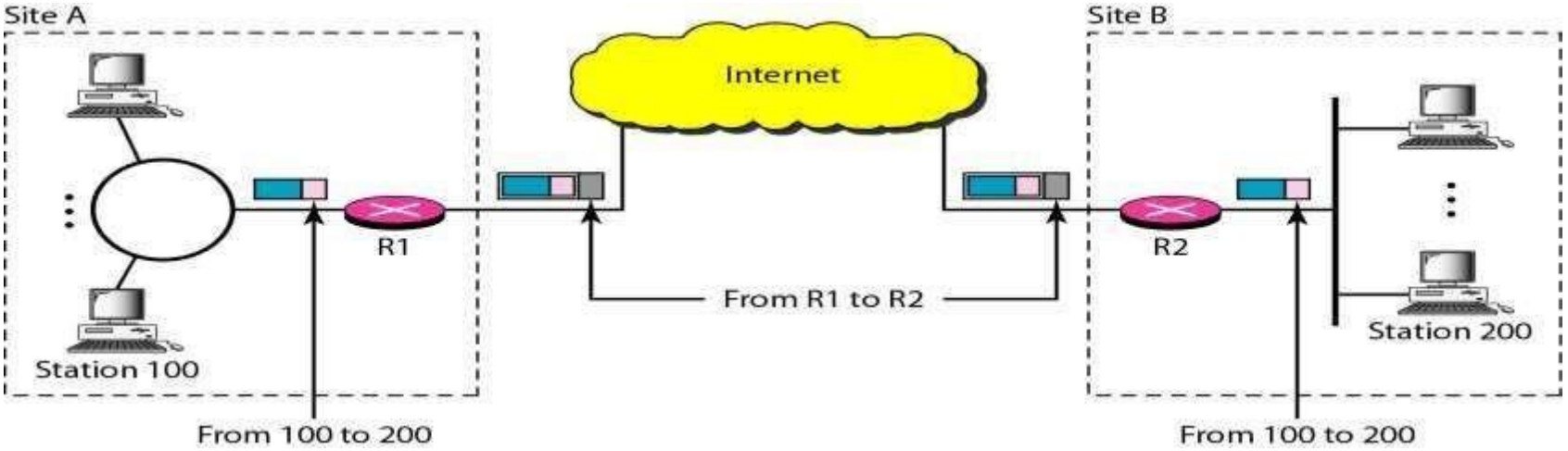


Fig . 5.19 Addressing in a VPN

chapter-3

SYSTEM SECURITY

TLI
NE

DESCRIPTION OF THE SYSTEM

- **USERS**
- **TRUST AND TRUSTED SYSTEMS**
- **BUFFER OVERFLOW AND MALICIOUS SOFTWARE**
- **MALICIOUS PROGRAMS**
- **WORMS**
- **VIRUSES**
- **INTRUSION DETECTION SYSTEM(IDS)**
- **FIREWALLS**

SYSTEM SECURITY

- There are security classifications or security levels
- Subjects have security clearances
- Objects have security classifications
- Example of security levels
- Top Secret
- Secret
- Confidential
- Unclassified
- In this case Top Secret > Secret > Confidential > Unclassified 5 5 Data

DESCRIPTION OF THE SYSTEM

- A **system** is a group of interacting or interrelated entities that form a unified whole.
- A **system** is described by its spatial and temporal boundaries, surrounded and influenced by its environment, described by its structure and purpose and expressed in its functioning.
- **Systems** are the subjects of study of **systems** theory.
- Security can be compromised via any of the breaches mentioned:
- **Breach of confidentiality:** This type of violation involves the unauthorized reading of data.
- **Breach of integrity:** This violation involves unauthorized modification of data.
- **Breach of availability:** It involves an unauthorized destruction of data.
- **Theft of service:** It involves an unauthorized use of resources.
- **Denial of service:** It involves preventing legitimate use of the system. As mentioned before, such attacks can be accidental in nature.

USERS

- A user or human visible level and a machine level.
- The human-level authentication is a simple login where you provide a net ID and a password to gain access.
- Machine level authentication is however more complex and involves a predetermined ID and password that only a machine authorized to access the network can know.
- **End-user education** addresses the most unpredictable cyber-security factor: people.
- Anyone can accidentally introduce a virus to an otherwise secure system by failing to follow good security practices.
- Teaching users to delete suspicious email attachments, not plug in unidentified USB drives, and various other important lessons is vital for the security of any organization.

TRUST AND TRUSTED SYSTEMS

TRUSTED SYSTEMS

- Protection of data and resources on the basis of levels of security (e.g. military)
- In military, information is categorize as unclassified , confidential , secret , top secret .
- Users can be granted clearances to access certain categories of data.

MULTILEVEL SECURITY

- In which a subject at high level may not convey information to a subject at low level
- A multilevel secure system must enforce:
- No read up: A subject can only read an object of less or equal security level (Simple Security Property)
- No write down: A subject can only write into an object of greater or equal security level (*-Property)
- Reference Monitor Concept: Multilevel security for a data processing system

BUFFER OVERFLOW

- A very common attack mechanism
  - From 1988 morris worm to code red, slammer, sasser and many others
- Prevention techniques known
- Still of major concern due to
  - Legacy of widely deployed buggy
  - Continued careless programming techniques

BUFFER OVERFLOW BASICS

- Caused by programming error
- Allows more data to be stored than capacity available in a fixed sized buffer
  - Buffer can be on stack, heap, global data

- **OVERWRITING ADJACENT MEMORY LOCATIONS**
  - Corruption of program data
  - Unexpected transfer of control
  - Memory access violation
  - Execution of code chosen by attacker

BUFFER OVERFLOW EXAMPLE

```
int main(int argc, char *argv[]) {
    int valid = FALSE;
    char  str1[8];
    char str2[8];

    next_tag(str1);
    gets(str2);
    if (strncmp(str1, str2, 8) == 0)
        valid = TRUE;
    printf("buffer1: str1(%s), str2(%s),
        valid(%d)\n", str1, str2, valid);
}
```

```
$ cc -g -o buffer1 buffer1.c
$ ./buffer1
START
buffer1: str1(START), str2(START), valid(1)
$ ./buffer1
EVILINPUTVALUE
buffer1: str1(TVALUE),
str2(EVILINPUTVALUE), valid(0)
$ ./buffer1
BADINPUTBADINPUT
buffer1: str1(BADINPUT),
str2(BADINPUTBADINPUT), valid(1)
```

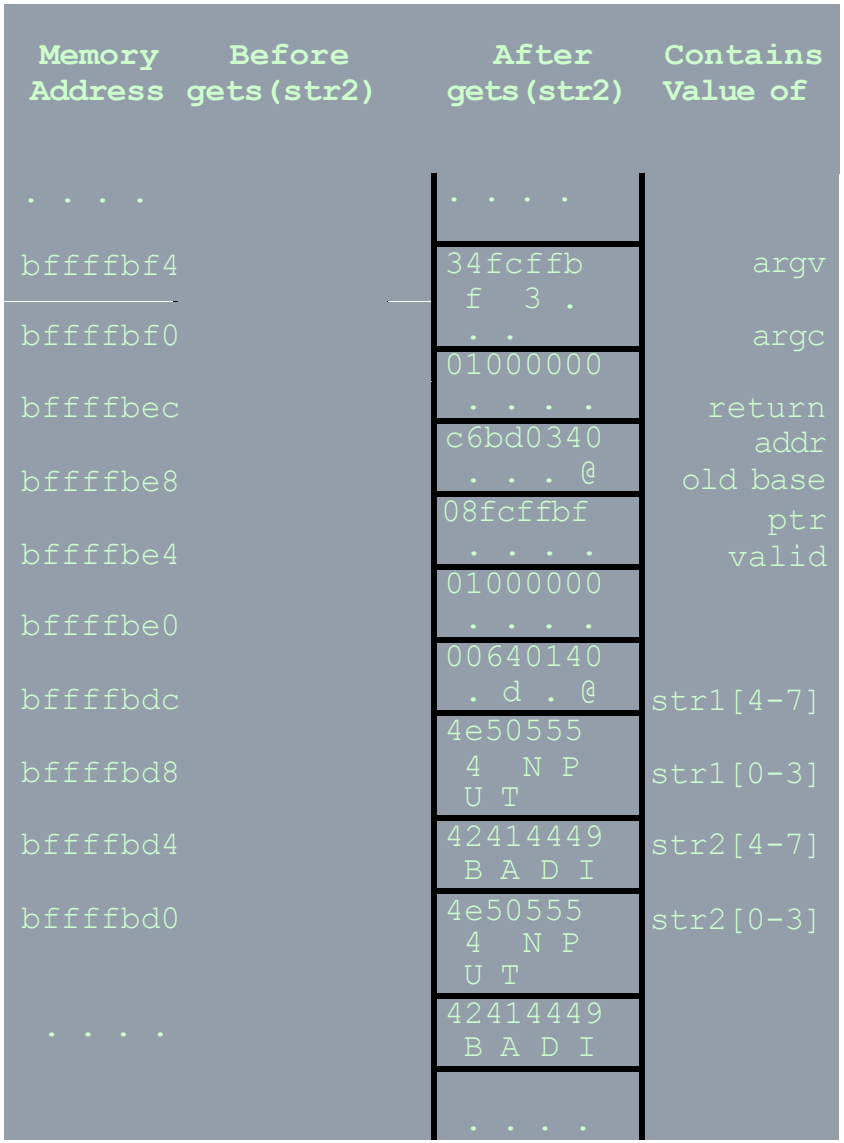
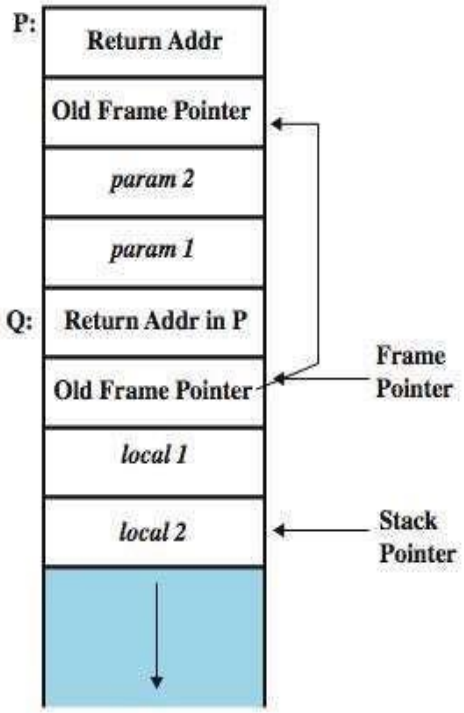


Fig . 5.20

BUFFER OVERFLOW ATTACKS

- To exploit a buffer overflow an attacker
  - Must identify a buffer overflow vulnerability in som
- Inspection, tracing execution, fuzzing tools
  - Understand how buffer is stored in memory d det
    an for corruption



Function Calls and Stack Frames

Fig . 5.21

BUFFER OVERFLOW DEFENSES

- Buffer overflows are widely exploited
- Large amount of vulnerable code in use
  - Despite cause and countermeasures known

- **Two broad defense approaches**
  - Compile-time - harden new programs
  - Run-time - handle attacks on existing programs

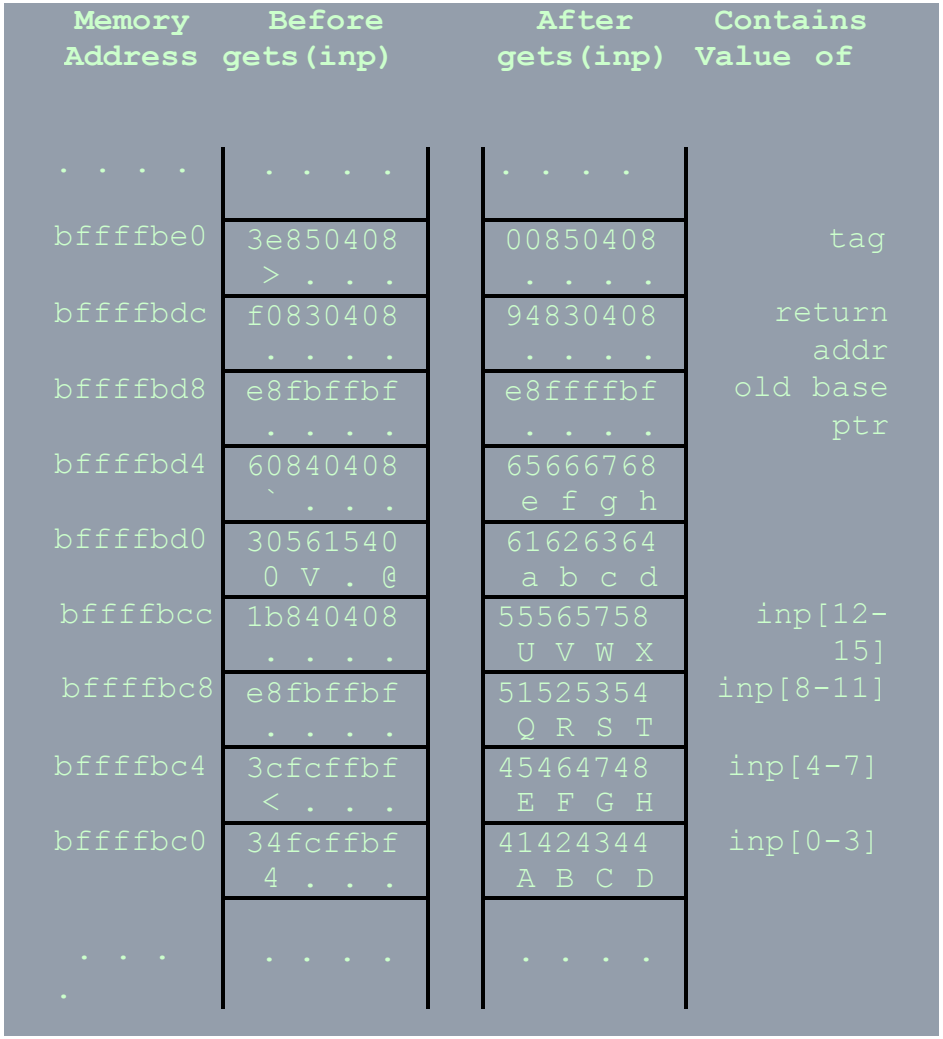


Fig . 5.22

HEAP OVERFLOW

- **ALSO ATTACK BUFFER LOCATED IN HEAP**
  - Typically located above program code
  - Memory requested by programs to use in dynamic data structures, e.G. Linked lists
- **NO RETURN ADDRESS**
  - Hence no easy transfer of control
  - May have function pointers can exploit
  - Manipulate management data structures
- Defenses: non executable or random heap

MALWARE

- General misconception among people
- Malware = “malicious software”
- Malware is any kind of unwanted software that is installed without your consent on your computer.
- Viruses, worms, Trojan horses, bombs, spyware, adware, Ransomware are subgroups of malware.

VIRUSES

- A virus tries to infect a carrier, which in turn relies on the carrier to spread the virus around.
- A computer virus is a program that can replicate itself and spread from one computer to another.
- **Direct infection**: Virus can infect files every time a user opens that specific infected program, document or file.
- **Fast infection**: is when a virus infects any file that is accessed by the program that is infected.
- **Slow infection**: is when the virus infects any new or modified program, file or document.
  - Great way to trick a antivirus program!
- **Sparse Infection**: is the process of randomly infecting files, etc. on the computer.
- **RAM-resident infection**: is when the infection buries itself in your computer's random access memory.

TROJANS

- Trojan horse: is a program or software designed to look like a useful or legitimate file.
- Once the program is installed and opened it steals information or deletes data.
- Trojan horses compared to other types of malware is that it usually runs only once and then is done functioning.
- Some create back-door effects
- Another distribution of Trojans is by infecting a server that hosts websites.
- Downfall of Trojans: very reliant on the user.

WORMS

- Worms and viruses get interchanged commonly in the media.
- In reality a worm is more dangerous than a virus.
- User Propagation vs. Self Propagation
- Worm is designed to replicate itself and disperse throughout the user's network.
- Email Worms and Internet Worms are the two most common worm.
- **EMAIL WORM**
- Email worm goes into a user's contact/address book and chooses every user in that contact list.
- It then copies itself and puts itself into an attachment; then the user will open the attachment and the process will start over again!
- **INTERNET WORMS**
- A internet worm is designed to be conspicuous to the user.
- The worms scans the computer for open internet ports that the worm can download itself into the computer.
- Once inside the computer the worms scans the internet to infect more computers.

ADWARE AND SPYWARE

- Adware is a type of malware designed to display advertisements in the user's software.
- They can be designed to be harmless or harmful; the adware gathers information on what the user searches the World Wide Web for.
- With this gathered information it displays ads corresponding to information collected.
- Spyware is like adware it spies on the user to see what information it can collect off the user's computer to display pop ads on the user's computer.
- Spyware unlike adware likes to use memory from programs running in the background of the computer to keep close watch on the user.
- This most often clogs up the computer causing the program or computer to slow down and become un-functional.

ANTIVIRUS PROGRAMS

- Antivirus programs are designed to detect malware trying to enter the user's system.
- There are several ways a antivirus program can track malware entering the computer.
- **Software can use**
  - Signature based detection
  - Heuristics
  - Cloud Antivirus
  - Network Firewall

NETWORK FIREWALL

- Operating systems way of protecting the user from unknown programs.
- Not technically a antivirus program
- Monitors the TCP/IP ports programs tries to access.

How can we protect ourselves

- Use an antivirus program and keep it up to date!
  - Yes they only protect from know malicious code out there, but it's still something!

Operating System's Security

- Keep your Operating System up to date!
  - Windows is one of the most hacked OS on the market.
  - The updates are mostly focused on security patches

INTRUDERS

- Significant issue for networked systems is hostile or unwanted access
- Either via network or local
- **CAN IDENTIFY CLASSES OF INTRUDERS**
  - Masquerader
  - Misfeasor
  - Clandestine user
- Varying levels of competence

EXAMPLES OF INTRUSION

- Remote root compromise
- Web server defacement
- Guessing / cracking passwords
- Copying viewing sensitive data / databases
- Running a packet sniffer
- Distributing pirated software
- Using an unsecured modem to access net
- Impersonating a user to reset password
- Using an unattended workstation

HACKERS

- **MOTIVATED BY THRILL OF ACCESS AND STATUS**
  - Hacking community a strong meritocracy
  - Status is determined by level of competence
- **BENIGN INTRUDERS MIGHT BE TOLERABLE**
  - Do consume resources and may slow performance
  - Can’t know in advance whether benign or malign
- IDS / IPS / vpns can help counter
- Awareness led to establishment of certs
  - Collect / disseminate vulnerability info / responses

HACKER BEHAVIOR EXAMPLE

- Select target using IP lookup tools
- Map network for accessible services
- Identify potentially vulnerable services
- Brute force (guess) passwords
- Install remote administration tool
- Wait for admin to log on and capture password
- Use password to access remainder of network

INTRUSION TECHNIQUES

- Aim to gain access and/or increase privileges on a system
- Often use system / software vulnerabilities
- Key goal often is to acquire passwords
  - So then exercise access rights of owner
- **BASIC ATTACK METHODOLOGY**
  - Target acquisition and information gathering
  - Initial access
  - Privilege escalation

INTRUSION TECHNIQUES

INTRUSION DETECTION

- Covering tracks inevitably will have security failures
- So need also to detect intrusions so can
  - Block if detected quickly
  - Act as deterrent
  - Collect info to improve security
- Assume intruder will behave differently to a legitimate user
  - But will have imperfect distinction between

PASSWORD

Password Guessing

- One of the most common attacks
- Attacker knows a login (from email/web page etc)
- Then attempts to guess password for it
  - Defaults, short passwords, common word searches
  - User info (variations on names, birthday, phone, common words/interests)
  - Exhaustively searching all possible passwords
- Check by login or against stolen password file
- Success depends on password chosen by user
- Surveys show many users choose poorly

PASSWORD

Password Capture

- Another attack involves **password capture**
  - Watching over shoulder as password is entered
  - Using a trojan horse program to collect
  - Monitoring an insecure network login
    - Eg. Telnet, FTP, web, email
  - Extracting recorded info after successful login (web history/cache, last number dialed etc)
- Using valid login/password can impersonate user
- Users need to be educated to use suitable precautions/countermeasures

APPROACHES TO INTRUSION DETECTION

- **STATISTICAL ANOMALY DETECTION**
  - Attempts to define normal/expected behavior
  - Threshold
  - Profile based
- **RULE-BASED DETECTION**
  - Attempts to define proper behavior
  - Anomaly
  - Penetration identification

AUDIT RECORDS

- Fundamental tool for intrusion detection
- **NATIVE AUDIT RECORDS**
  - Part of all common multi-user O/S
  - Already present for use
  - May not have info wanted in desired form
- **DETECTION-SPECIFIC AUDIT RECORDS**
  - Created specifically to collect wanted info
  - At cost of additional overhead on system

STATISTICAL ANOMALY DETECTION

- **Threshold detection**
  - Count occurrences of specific event over time
  - If exceed reasonable value assume intrusion
  - Alone is a crude & ineffective detector
- **Profile based**
  - Characterize past behavior of users
  - Detect significant deviations from this
  - Profile usually multi-parameter

AUDIT RECORD ANALYSIS

- Foundation of statistical approaches
- Analyze records to get metrics over time
  - Counter, gauge, interval timer, resource use
- Use various tests on these to determine if current behavior is acceptable
  - Mean & standard deviation, multivariate, markov process, time series, operational
- Key advantage is no prior knowledge used

RULE-BASED INTRUSION DETECTION

- Observe events on system & apply rules to decide if activity is suspicious or not
- **RULE-BASED ANOMALY DETECTION**
  - Analyze historical audit records to identify usage patterns & auto-generate rules for them
  - Then observe current behavior & match against rules to see if conforms
- Like statistical anomaly detection does not require prior knowledge of security flaws
- **RULE-BASED PENETRATION IDENTIFICATION**
  - Uses expert systems technology
  - With rules identifying known penetration, weakness patterns, or suspicious behavior
  - Compare audit records or states against rules
  - Rules usually machine & o/s specific
  - Rules are generated by experts who inter
  - Quality depends on how well this is done

DISTRIBUTED INTRUSION DETECTION

- Traditional focus is on single systems
- But typically have networked systems
- More effective defense has these working together to detect intrusions
- Issues
  - Dealing with varying audit record formats
  - Integrity & confidentiality of networked data
  - Centralized or decentralized architecture

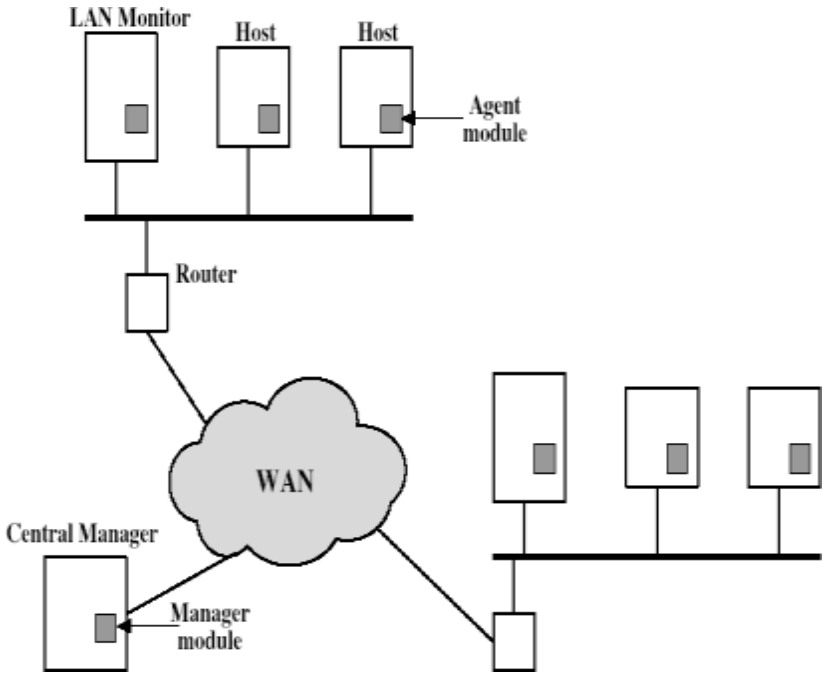


Fig . 5.23 Distributed Intrusion Detection - Architecture

DISTRIBUTED INTRUSION DETECTION – AGENT IMPLEMENTATION

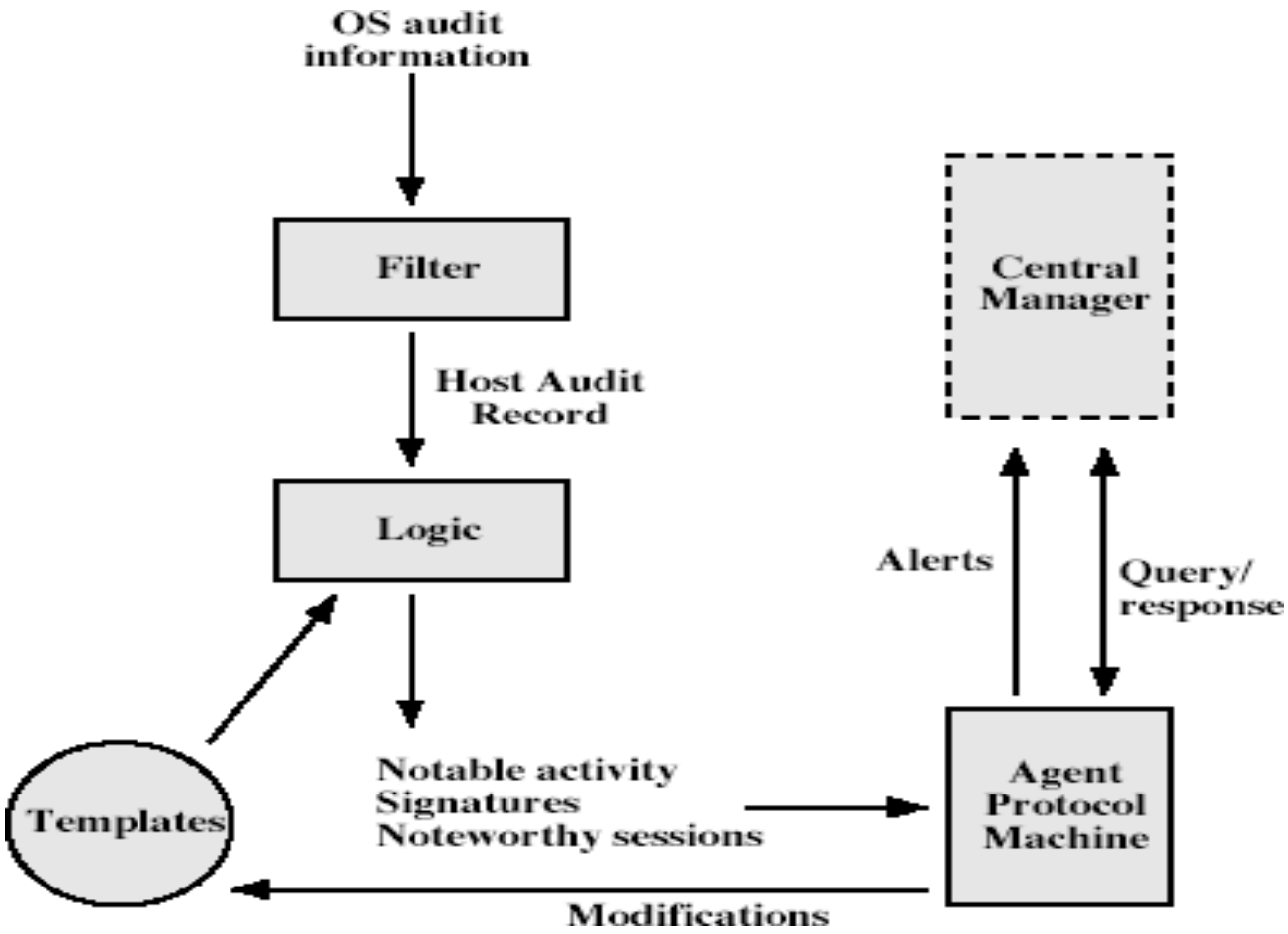


Fig . 5.24

PASSWORD MANAGEMENT

- Front-line defense against intruders
- Users supply both:
  - Login – determines privileges of that user
  - Password – to identify them
- Passwords often stored encrypted
  - Unix uses multiple DES (variant with salt)
  - More recent systems use crypto hash function
- Should protect password file on system

MANAGING PASSWORDS

Managing Passwords - Education

- Can use policies and good user education
- Educate on importance of good passwords
- Give guidelines for good passwords
  - Minimum length (>6)
  - Require a mix of upper & lower case letters, numbers, punctuation
  - Not dictionary words
- But likely to be ignored by many users

MANAGING PASSWORDS

Managing Passwords - Computer Generated

- Let computer create passwords
- If random likely not memorisable, so will be written down (sticky label syndrome)
- Even pronounceable not remembered
- Have history of poor user acceptance
- FIPS PUB 181 one of best generators
  - Has both description & sample code
  - Generates words from concatenating random pronounceable syllables

FIREWALLS-INTRODUCTION

- Seen evolution of information systems
- Now everyone want to be on the internet
- And to interconnect networks
- Has persistent security concerns
  - Can’t easily secure every system in org
- Typically use a firewall
- To provide perimeter defense
- As part of comprehensive security strategy

WHAT IS A FIREWALL?

- A **choke point** of control and monitoring
- Interconnects networks with differing tru st
- Imposes restrictions on network services
  - Only authorized traffic is allowed
- Auditing and controlling access
  - Can implement alarms for abnormal behavior
- Provide NAT & usage monitoring
- Implement vpns using ipsec
- Must be immune to penetration

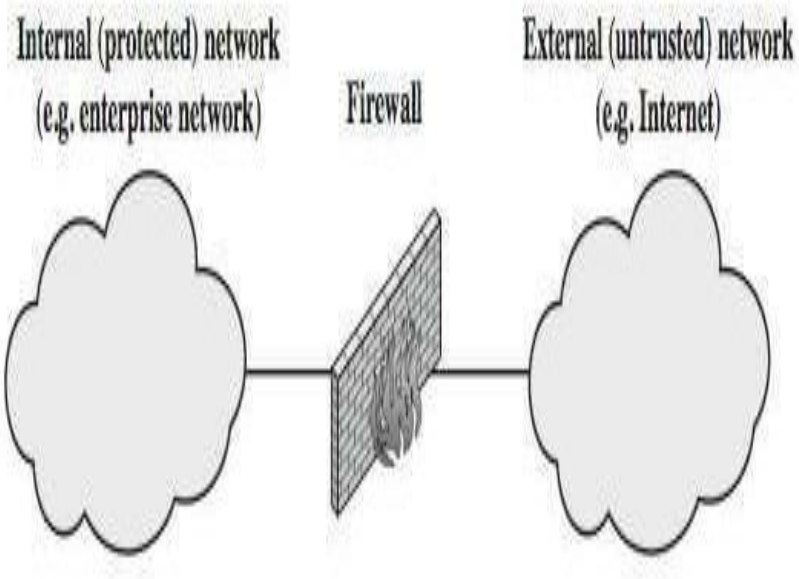


Fig . 5.24

FIREWALL LIMITATIONS

- Eg sneaker net, utility modems, trusted organisations, trusted services (eg ssl/ssh)
- Eg disgruntled or colluding employees

- If improperly secured agaCannot protect from attacks bypassing it
- Cannot protect against internal threats
- Cannot protect against access via wlan
- Cannot protect against malware imported via laptop, pda, storage infected outside
- inst external use

FIREWALLS – PACKET FILTERS

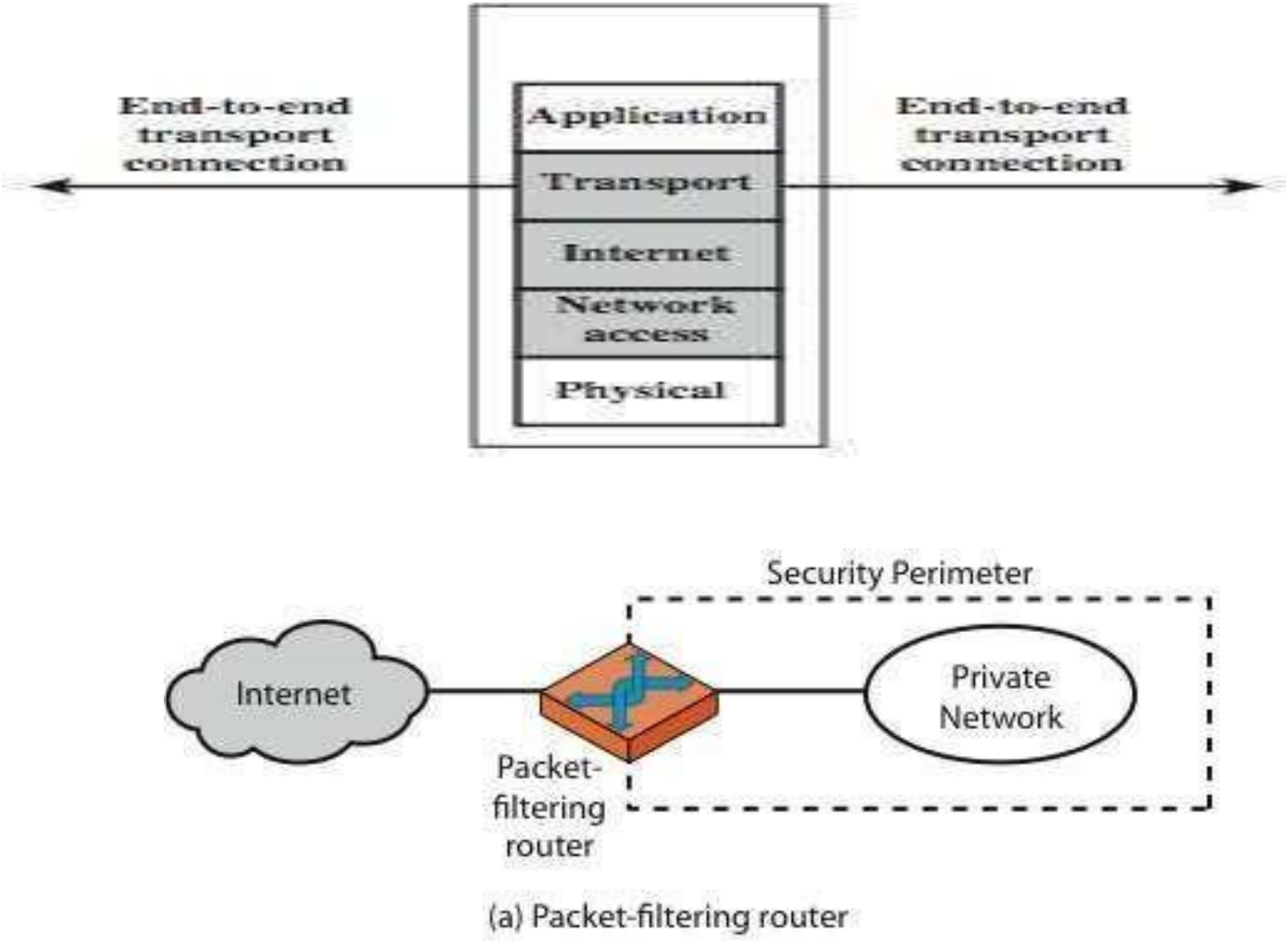


Fig . 5.25

ATTACKS ON PACKET FILTERS

- **IP ADDRESS SPOOFING**
  - Fake source address to be trusted
  - Add filters on router to block
- **SOURCE ROUTING ATTACKS**
  - Attacker sets a route other than default
  - Block source routed packets
- **TINY FRAGMENT ATTACKS**
  - Split header info over several tiny packets
  - Either discard or reassemble before check

FIREWALLS - APPLICATION LEVEL GATEWAY

- Have application specific
- **Has full access to protocc**
  - User requests service from proxy
  - Proxy validates request as legal
  - Then actions request and returns result
  - Can log / audit traffic at application lev
  - Others are more problematic
- **Need separate proxies for each service**

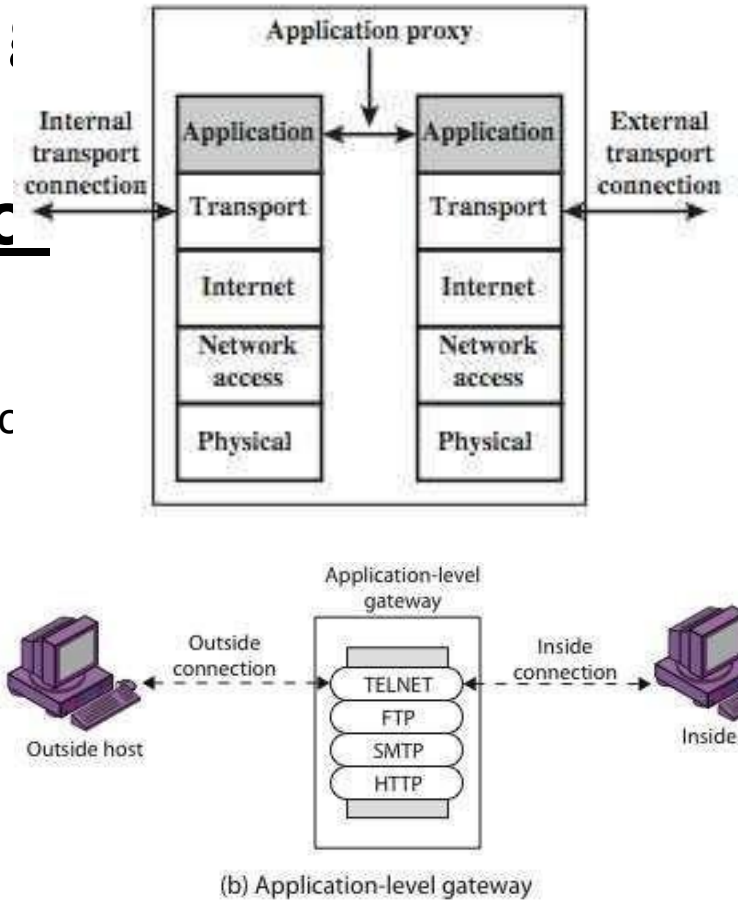
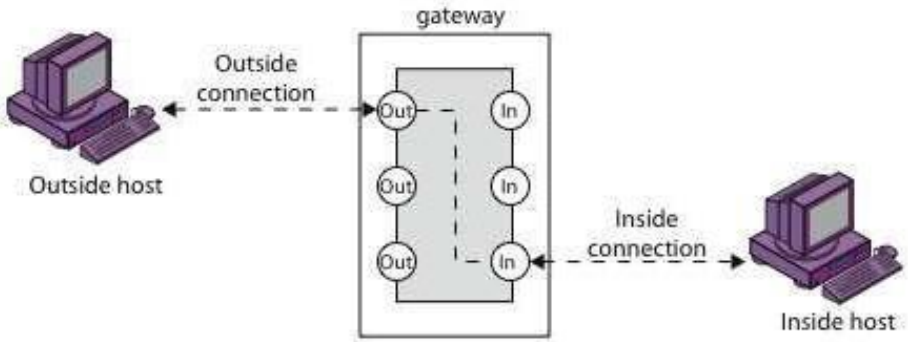


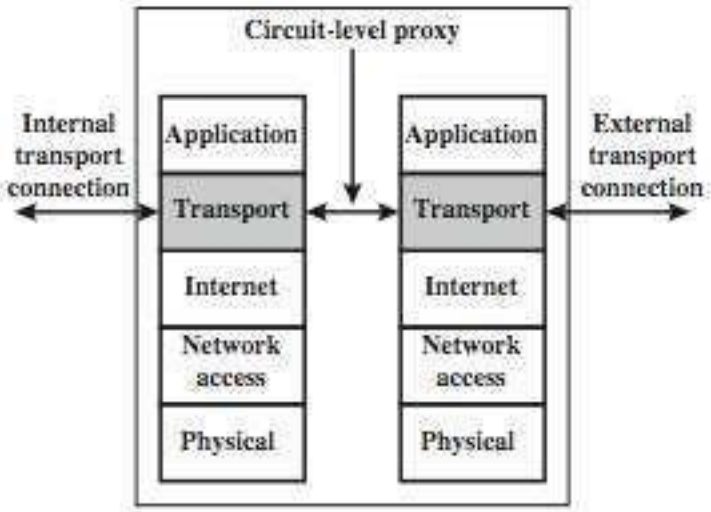
Fig 5.26

FIREWALLS - CIRCUIT LEVEL GATEWAY

- Relays two TCP connections
- Imposes security by limiting which such connections are allowed
- Once created usually relays traf fic without examining contents
- Typically used when trust internal users by allowing general outbound connections
- SOCKS is commonly used



(c) Circuit-level gateway



(e) Circuit-level proxy firewall

Fig . 5.27

PERSONAL FIREWALLS

- Controls traffic between PC/workstation and internet or enterprise network
- A software module on personal computer
- Or in home/office DSL/cable/ISP router
- Typically much less complex than other firewall types
- Primary role to deny unauthorized remote access to the computer
- And monitor outgoing activity for malware

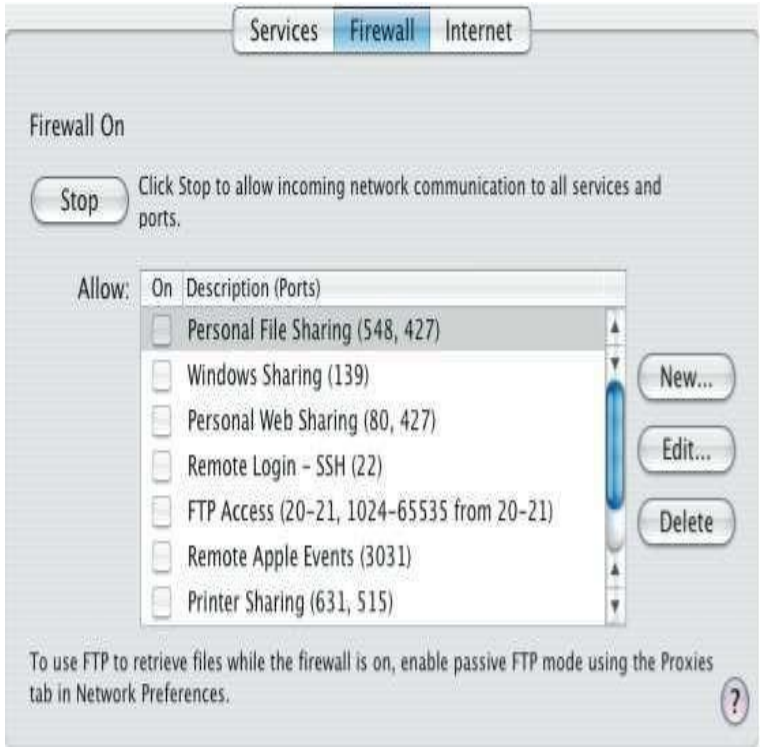
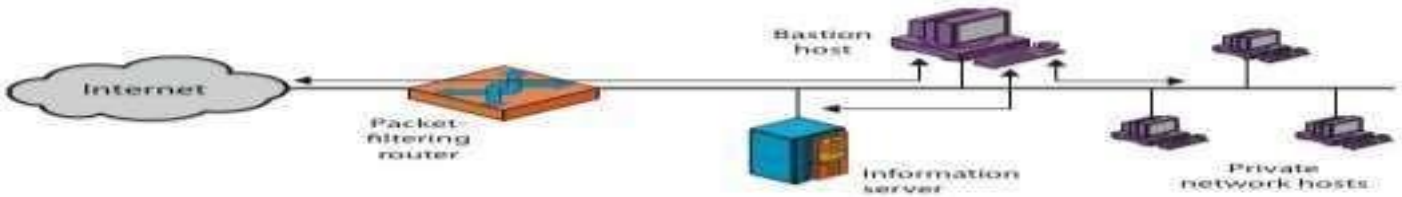
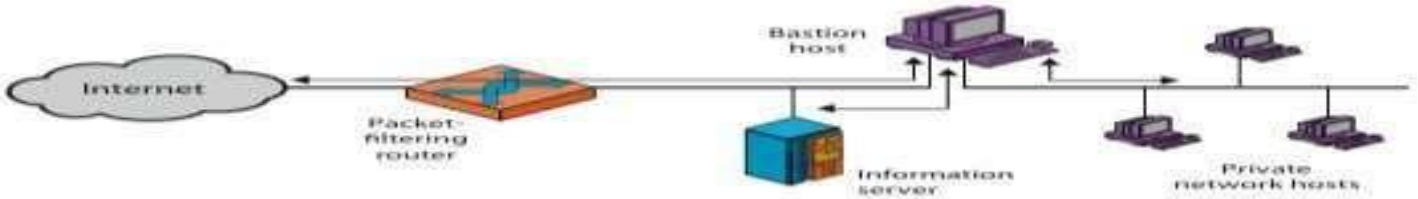


Fig . 5.28

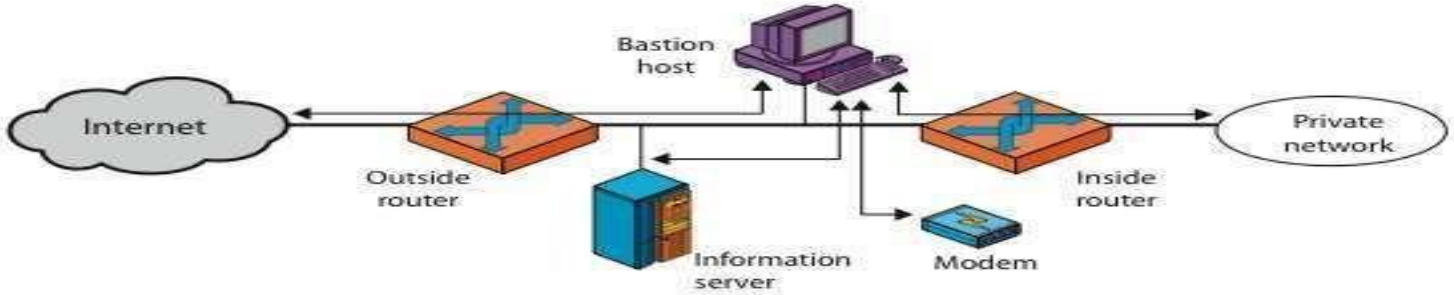
FIREWALL CONFIGURATIONS



(a) Screened host firewall system (single-homed bastion host)



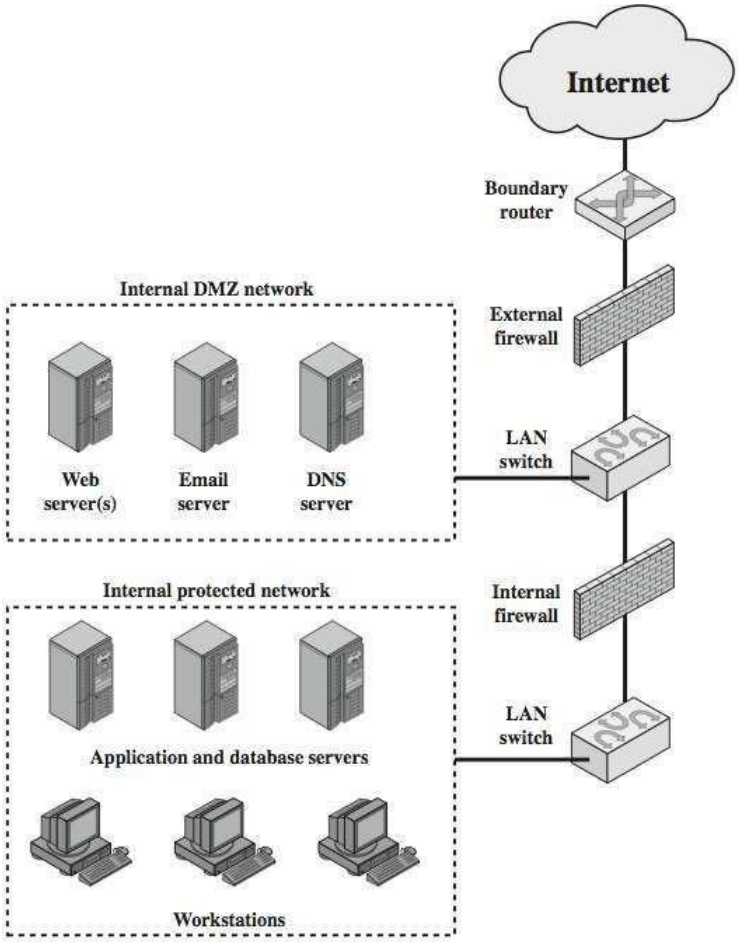
(b) Screened host firewall system (dual-homed bastion host)



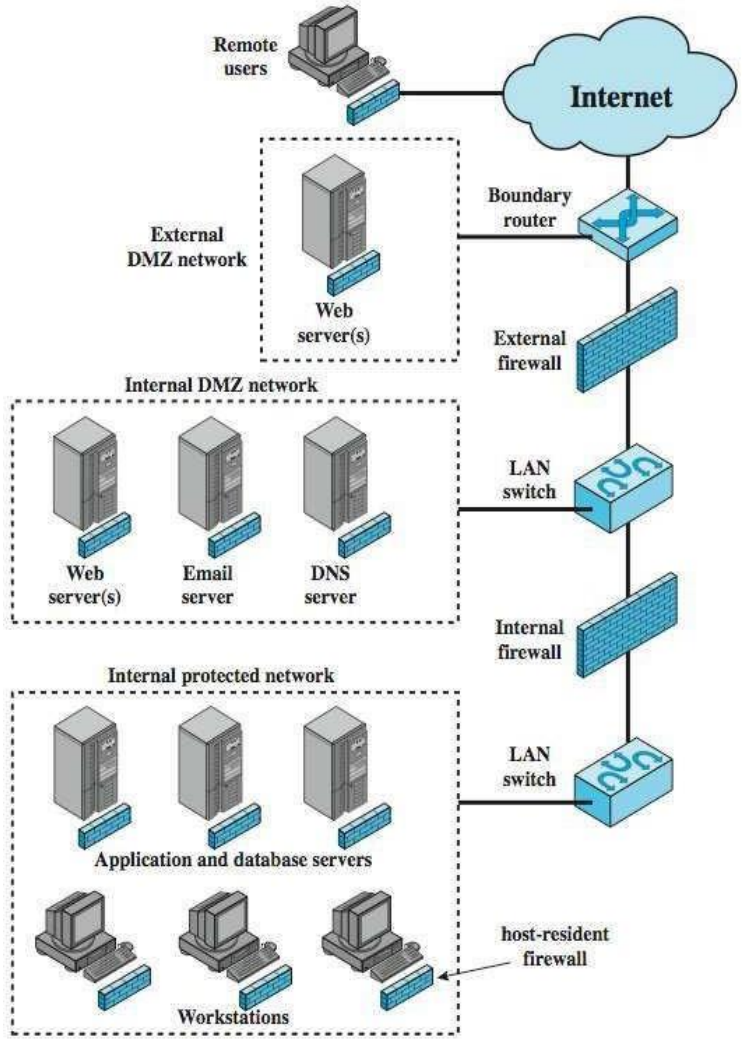
(c) Screened-subnet firewall system

Fig . 5.29

DMZ NETWORKS & DISTRIBUTED FIREWALLS



DMZ Networks



Distributed Firewalls

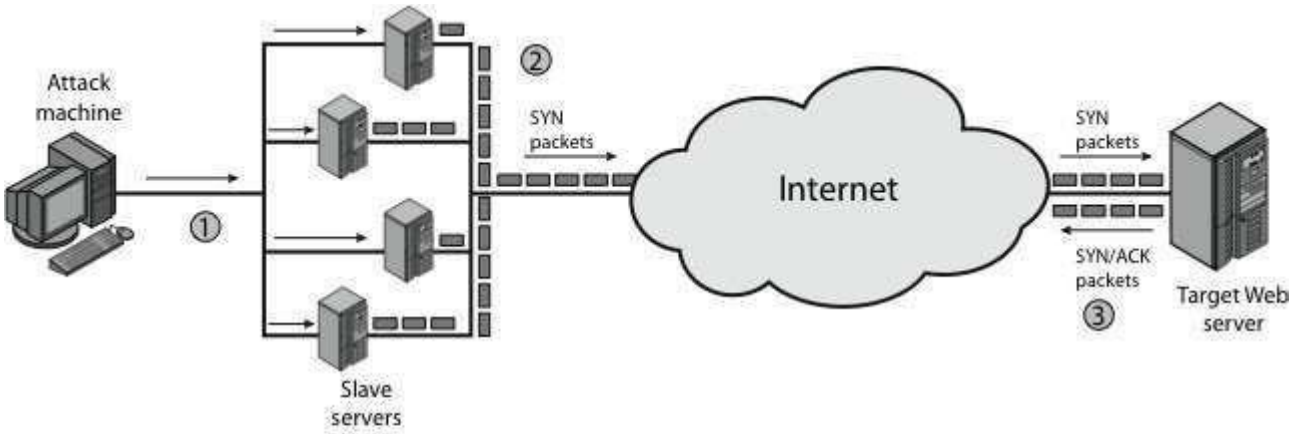
Fig . 5.30

CONTENT BEYOND SYLLABUS

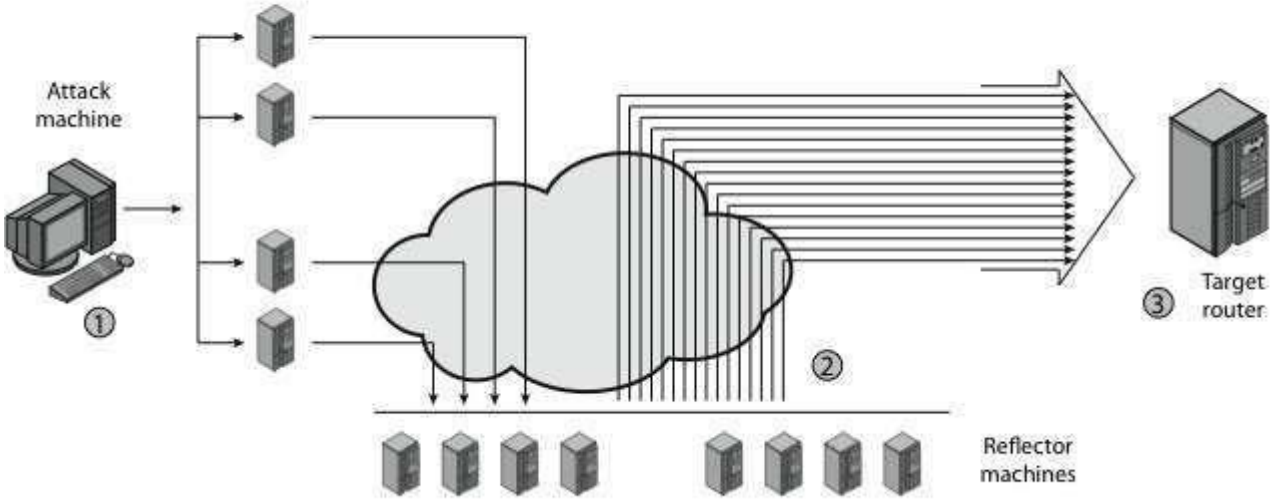
DISTRIBUTED DENIAL OF SERVICE ATTACKS (DDOS)

- Distributed Denial of Service (DDoS) attacks form a significant security threat
- making networked systems unavailable
- by flooding with useless traffic
- using large numbers of “zombies”
- growing sophistication of attacks
- defense technologies struggling to cope

DISTRIBUTED DENIAL OF SERVICE ATTACKS (DDOS)



(a) Distributed SYN flood attack



(a) Distributed ICMP attack

DDOS COUNTERMEASURES

- three broad lines of defense:
  - attack prevention & preemption (before)
  - attack detection & filtering (during)
  - attack source traceback & identification (after)
- huge range of attack possibilities
- hence evolving countermeasures

RESOURCES

- **Lecture Notes - Lecture Notes**
- **Video Lectures - Video Lecture**
- **E-Book - Information Security Concepts**
- **Model Papers - JNTUA Question Papers**