Tribhuvan University

BICTE 7th Semester

Network Security

All in One Chapter Note

1. Course Outlines:

Specific Objectives	Contents	Hour
• Explain the concept of Computer Security. • Explain the OSI Security Architecture. • Classify the security attacks. • Explain the different security services. • Explain the mechanism for securing information in network. • Explain the Model of network security. • Discuss classical cryptography approaches.	**1. Introduction** 1.1 Computer Security Concept 1.2 The OSI Security Architecture 1.3 Security Attacks 1.4 Security Services 1.5 Security Mechanism 1.6 A Model for Network Security 1.7 Classical cryptography	10
• Differentiate between Cryptography and Cryptanalysis. • Explain the Feistel Cipher Structure. • Demonstrate the DES, 3DES, and AES algorithms. • Explain the Cipher Block Modes: Electronic Codebook Mode, Cipher Block Chaining Mode, Cipher feedback Mode and Counter Mode	**2. Symmetric Encryption and message Confidentiality** 2.1 Symmetric Encryption Principles 2.2 Symmetric Block Encryption Algorithms 2.3 Cipher Block Modes of Operations	12
• Explain the hash function requirements. • Demonstrate the SHA Secure hash function. • Demonstrate the working principle of MD. • Explain the Public-Key Encryption Structure. • Explain the Applications for Public-Key Cryptosystem. • Explain the requirements for Public-Key Cryptography. • Explain the RSA Public-Key Encryption Algorithm. • Explain the Diffie-Hellman Key Exchange algorithm. • Explain Digital signature algorithm with example.	**3. Public-Key Cryptography and Message Digest** 3.1 Secure Hash functions 3.2 Message Digest(MD) 3.3 Public-Key Cryptography Principles 3.4 Public-Key Cryptography Algorithms 3.5 Digital Signatures	16
• Explain the Public-Key Infrastructure Functions and Protocols. • Explain the different types of transport layer security. • Explain the different protocols of wireless security. • Explain the protocols used in email security. • Explain the mechanism of IP Security.	**4. Network Security Applications** 4.1 Public-Key Infrastructure 4.2 Transport Layer Security: SSL, HTTPs, Secure Shell(SSH) 4.3 Wireless Security: WEP, WAP, WPA2 4.4 E-Mail Security: PGP, S/MIME 4.5 IP Security	18
• Explain the different methods of intrusion detection. • Explain the different types of Malicious Software. • Explain the Characteristics and types of firewalls. • Implement the basic features of firewall.	**5. System Security** 5.1 Intruders 5.2 Malicious Software 5.3 Firewall	14
• Explain the basic concept of SNMP. • Explain the features of SNMPv1. • Explain the features of SNMPv3.	**6. Network management Security** 6.1 Basic Concept of SNMP 6.2 SNMPv1 6.3 SNMPv3	10

UNIT 1
Network Security
Introduction Network

Network is collection of communicating devices which are connected to each other in a certain pattern and communicate with each other using some protocol
Protocol are set of rules that defines how communication is going to happen

Computer Security Concept

- Computer security is refers to techniques for ensuring that data stored in a computer cannot be read or compromised by any individuals without authorization.
- Most computer security measures involve data encryption and passwords.
- The purpose of computer security is to device ways to prevent the weaknesses from being exploited

These are the three goals in computing Security.
1.Confidentiality
2.Integrity
3.Availability
Confidentiality: ensures that computer-related assets are accessed only by authorized parties. Confidentiality is sometimes called secrecy or privacy.
Integrity: it means that assets can be modified only by authorized parties or only in authorized ways.
Availability: it means that assets are accessible

VULNERABILITY

Vulnerability is a weakness in the security system.
Weaknesses can appear in any element of a computer, both in the hardware, operating system, and the software.
The types of vulnerabilities we might find as they apply to the assets of hardware, software, and data.

HARDWARE VULNERABILITY

Hardware is more visible than software, largely because it is composed of physical objects.
It is rather simple to attack by adding devices, changing them, removing them, intercepting the traffic to them, or flooding them with traffic until they can no longer function.other ways that computer hardware can be attacked physically.
Computers have been drenched with water, burned, frozen, gassed, and electrocuted with power surges.

SOFTWARE VULNERABILITIES

Software can be replaced, changed, or destroyed maliciously, or it can be modified, deleted, or misplaced accidentally.
Whether intentional or not, these attacks exploit the
software's vulnerabilities.
Sometimes, the attacks are obvious, as when the software no longer runs.
More subtle are attacks in which the software has been altered but seems to run normally.

DATA VULNERABILITY

A data attack is a more widespread and serious problem than either a hardware or software attack. data items have greater public value than hardware and software because more people know how to use or interpret data.

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 forms are
- Security attacks, security Mechanism , security services

- Security attack - Any action that compromises the security of information owned by an organization
  Security mechanism - A process that is designed to detect, prevent or recover from a security attack
- Security Services - A processing or communication services that enhances the security of data processing system and information transfers of an organization.

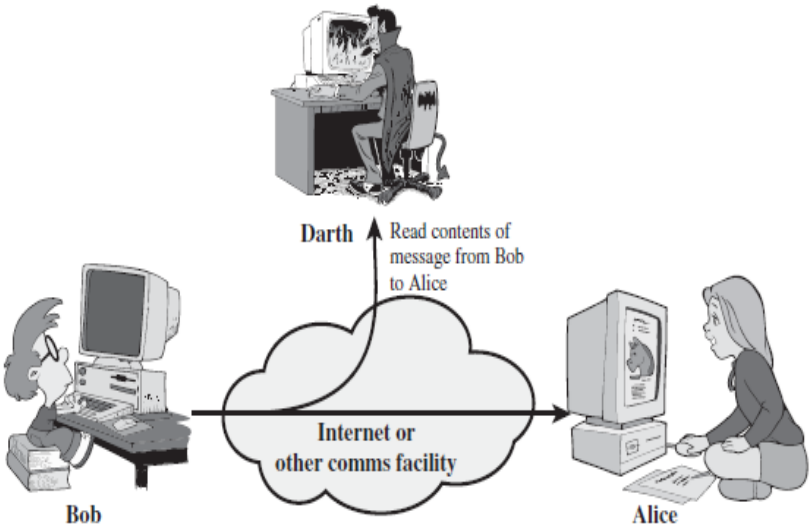
The services are intended to counter security attacks and they make use of one or more security mechanism to provide the service
Threat - A potential for violation of security which exists when there is a circumstance capability action event that could breath security and causes harm
Threat is a possible danger that might exploit vulnerability
Attack- an assault on system security that derives from an intelligent threat.

Security Attacks

- Passive attack
- Active attack

Passive attacks

Passive attacks are accomplished "...by monitoring a system performing its tasks and collecting information".
Once this information is monitored and collected about a particular system, it may be used later to attack the same system or one related to it.
A passive attack monitors unencrypted traffic and looks for clear-text passwords and sensitive information that can be used in other types of attacks.
Passive attacks include traffic analysis, monitoring of unprotected communications, decrypting weakly encrypted traffic, and capturing authentication information such as passwords.
Two Types :
1. Release of message content
2. Traffic Analysis



Release of message content : 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.

Traffic Analysis

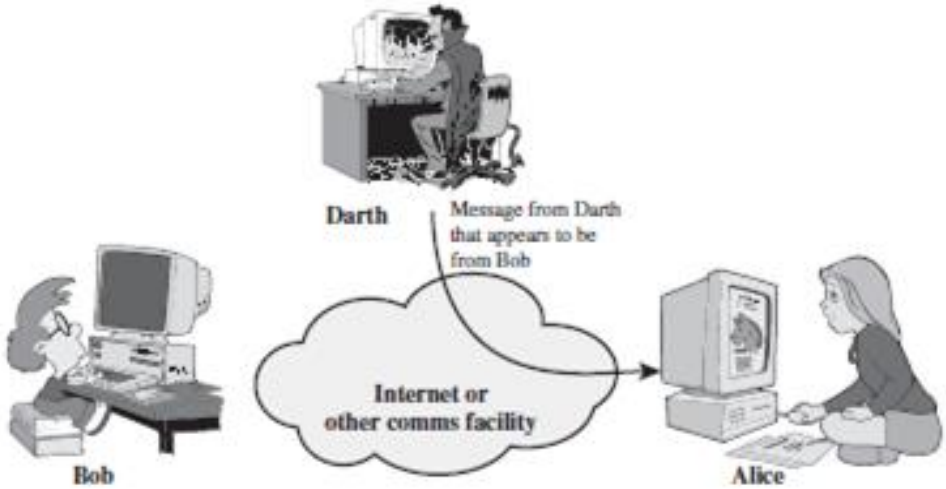
traffic analysis, is subtler
Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption.
If we had encryption protection in place, an opponent still might be able to observe the pattern of these messages.
The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged.
This information might be useful in guessing the nature of the communication that was taking place.

Sniffer

"A sniffer is an application or device that can read, monitor, and capture network data exchanges and read network packets".

Some consider sniffing as the most common type of passive attack on networks.
A sniffer can provide a full view of data inside a packet if the packets are not encrypted.
Sniffing programs come in the form of either commercial packet sniffers used to help maintain networks or underground packet sniffers used to break into computers.
An attacker using a sniffer can read a network's communications and analyze it to gain information to eventually cause the network to crash or even become corrupted.
Vulnerable protocols that are often sniffed, especially for passwords, include telnet, ftp, rlogin, IMAP, and POP

Active Attract
It involves some modification of data stream or the creation of false streamivided in four types

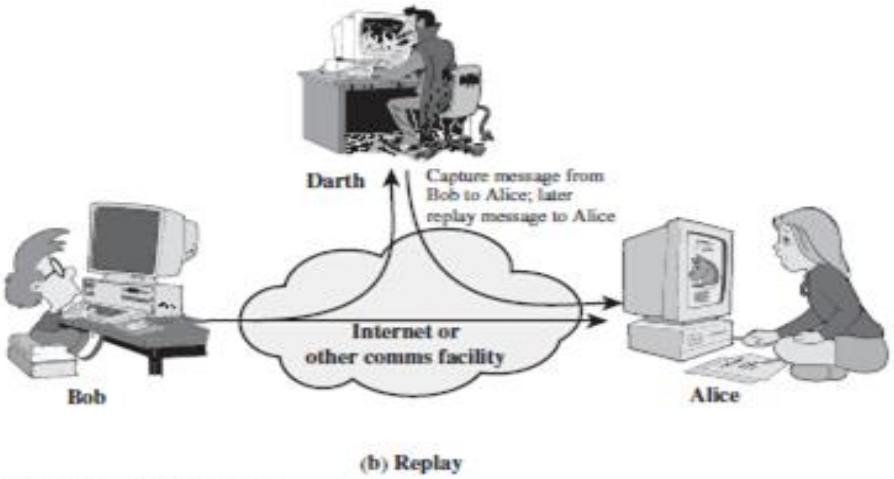


(a) Masquerade

1. Masquerade
2. Replay
3. Modification of message
4. Denial of services

Masquerade
A masquerade takes place when one entity pretends to be a different entity (Figure a) . A masquerade attack usually includes one of the other forms of active attack.
For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.
Replay
Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (Figure 1.3b).
Modification of message
Modification of messages simply means that some por tion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (Figure 1.3c).
For example, a message meaning "Allow John Smith to read confidential file accounts" is modified to mean "Allow Fred Brown to read confidential file accounts."
The denial of service prevents or inhibits the normal use or management of communications facilities (Figure 1.3d).
This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service).

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.



(b) Replay

Figure 1.3 Active Attacks

SECURITY SERVICES

X.800 defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers.

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

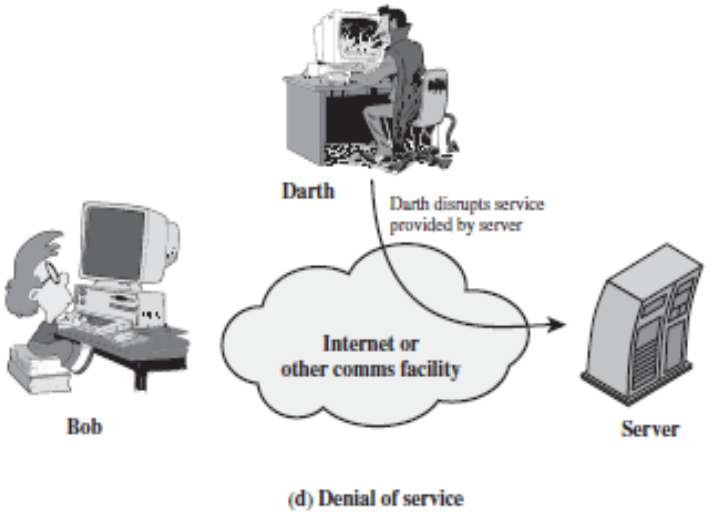
SECURITY SERVICES

Authentication ○ Peer Entity Authentication ○ Data-Origin Authentication ● Access Control ● Data Confidentiality Connection Confidentiality Connectionless Confidentiality Selective-Field **Confidentiality** Traffic-Flow Confidentiality	**Data Integrity** • Connection Integrity with Recovery • Connection Integrity without Recovery • Selective-Field Connection **Integrity** • Connectionless Integrity • Selective-Field Connectionless Integrity **Non Repudation** ○ Non repudiation, Origin ○ Non repudiation, Destination

Authentication

The authentication service is concerned with assuring that a communication is authentic.

- In the case of a single/ incoming message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from.
- In the case of an ongoing interaction, such as the connection of a terminal to a host, two aspects are involved.
- First, at the time of connection initiation, the service assures that the two entities are authentic (that is, that each is the entity that it claims to be).



(d) Denial of service

Figure 1.3 Active Attacks (*Continued*)

- Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unauthorized transmission or reception.

● Two specific authentication services are defined in X.800:

○ Peer entity authentication

○ Data origin authentication

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 the same protocol in different systems (e.g., two TCP modules in two communicating systems).

● Peer entity authentication is provided for use at the establishment of or during the data transfer phase of a connection.

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.

This type of service supports applications like electronic mail, where there are no prior interactions between the communicating entities.

Access Control

In the context of network security, access control 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.

Ex-:active directory

Data Confidentiality

Confidentiality is the protection of transmitted data from passive attacks

Protection of data from unauthorized disclosure

- Connection Confidentiality
- Connectionless Confidentiality
- Selective-Field Confidentiality
- Traffic-Flow Confidentiality

Connection Confidentiality – Protection of all user data on a connection

Connectionless Confidentiality – Protects all user data in a single data block

Selective-Field Confidentiality – The confidentiality of selected fields within the single data block

Traffic flow Confidentiality – Protection of the
information that might be desired from traffic flows

Data Integrity

- As with confidentiality, integrity can apply to a stream of messages, a single message, or selected fields within a message.
- A connection-oriented integrity service deals with a stream of messages and assures that messages are received as sent with no duplication, insertion, modification, reordering, or replays.
- On the other hand, a connectionless integrity service deals with individual messages without regard to any larger context and generally provides protection against message modification only.

- Because the integrity service relates to active attacks, we are concerned with detection rather than prevention.
- If a violation of integrity is detected, then the service may simply report this violation, and some other portion of software or human intervention is required to recover from the violation.
- Alternatively, there are mechanisms available to recover from the loss of integrity of data, as we will review subsequently.

Nonrepudiation

- Provides protection against denial by on of the entities involved in communication
- Nonrepudiation, Origin – proof that the message was sent by specified party
- Nonrepudiation, Destination – proof that the message was received by specified party

Security Mechanism

The 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.

Specific Security Mechanism

May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

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.

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).

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

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

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

Microsoft exchange server.

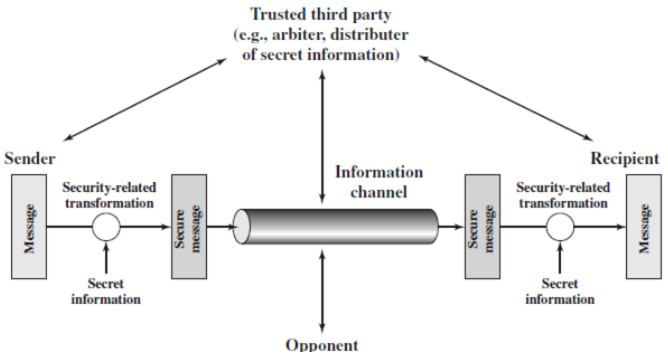
Traffic Padding : The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.

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

Define static path in router.

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

VeriSign authentication give the security again hackers



Pervasive Security Mechanism

Mechanisms that are not specific to any particular OSI security service or protocol layer.

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

Domain to domain

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

Event Detection: Detection of security-relevant events.
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.
Security Recovery: Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.

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 for the exchange to take place. A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.
Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on.
All of 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.

A trusted third party may be needed to achieve secure transmission.
For example, a third party may be responsible for distributing the secret information to the two principals while keeping it from any opponent.

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.

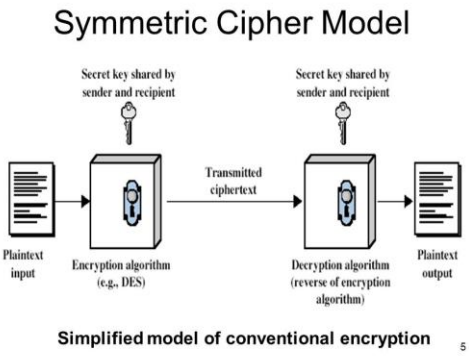
A general model of these other situations is illustrated by Figure below, which reflects a concern for protecting an information system from unwanted access.
Most readers are familiar with the concerns caused by the existence of hackers who attempt to penetrate systems that can be accessed over a network.
The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system.
The intruder can be a disgruntled employee who wishes to do damage or a criminal who seeks to exploit computer assets for financial gain (e.g., obtaining credit card numbers or performing illegal money transfers).

Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect applica tion programs as well as utility programs, such as editors and compilers.
Programs can present two kinds of threats:
1. Information access threats: Intercept or modify data on behalf of users who should not have access to that data.
Key loger, javascript

2. Service threats: Exploit service flaws in computers to inhibit use by legitimate users.
Ex-:svchost services

Viruses and worms are two examples of software attacks.

Such attacks can be introduced into a system by means of a disk that contains the unwanted logic concealed in otherwise useful software.
They also can be inserted into a system across a network; this latter mechanism is of more concern in network security.



Classical cryptography
Some Basic Terminology
Plaintext – original message
Ciphertext – coded message
Cipher – algorithm for transforming plaintext to ciphertext
Key – info used in cipher known only to sender/reciver
Encipher (encrypt) – converting plaintext to ciphertext
Decipher (decrypt)- recovering ciphertext from plaintext

Cryptography – study of encryption principles/methods
Cryptanalysis (codebreaking) – study of principles/ methods of deciphering ciphertext without knowing key
Cryptology – field of both cryptography and cryptanalysis
There are two types for encryption
Symmetric key encryption and
Public key encryption / Asymmetric Key encryption

Symmetric Key Encryption
symmetric key Encryption uses same key to encrypt data and decrypt data

- It is easiest to understand
- Faster compared to public key encryption
- Problem
- Key needs to be stored securely
- Secured channel is required to transfer the key

Public Key Encryption
Public key encryption uses two key Private and Public key
Slower as compare to symmetric key

Two requirements for secure use of symmetric encryption

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

Cryptography
Characterize cryptographic system by
Type of encryption operations used
○ Substitution / transposition / product
● Number of keys used
○ Single-key or private / two- key or public
● Way in which plaintext is processed
○ Block / stream

Cryptanalysis
● Objective to recover key not just message

- General approaches
- Cryptanalytic attack
- Brute-force attack

Cryptanalytic attacks

- Ciphertext only
- Only know algorithm and ciphertext, is statistical, known or can identify plaintext
- Known plaintext
- Know/suspect pain text and ciphertext
- Chosen plaintext
- Select plaintext and obtain ciphertext

Brute force search

- Always possible to simply try every key
- Most basic attack, proportional to key size
- Assume either known / recognize plaintext

Classical substitution ciphers

- The letters of plaintext are replaced by other letters or by numbers or symbols
- Plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit

$$c_i = E(p_i) = p_i + 3$$

A full translation chart of the Caesar cipher is shown here.

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
Ciphertext	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

Using this encryption, the message

TREATY IMPOSSIBLE

would be encoded as

T	R	E	A	T	Y		I	M	P	O	S	S	I	B	L	E
w	u	h	d	w	b		l	p	s	r	v	v	l	e	o	h

patterns with ciphertext bit patterns

Caesar cipher

- Earliest known substitution cipher
- By julius caesar (50 – 60 BC)
- First attested use in military affairs
- Used until 16th century
- Replaces each letter by 3rd letter

Monoalphabetic cipher

- Rather than just shifting the alphabet
- Could shuffle (jumble) the letters arbitrarily
- Each plaintext letter maps to a different random ciphertext letter
- Hence key is 26 letters long

Monoalphabetic Substitution Cipher

Because additive, multiplicative, and affine ciphers have small key domains, they are very vulnerable to brute-force attack.

A better solution is to create a mapping between each plaintext character and the corresponding ciphertext character. Alice and Bob can agree on a table showing the mapping for each character.

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

We can use the key in Figure to encrypt the message

this message is easy to encrypt but hard to find the key

The ciphertext is

ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

UNIT 2

Symmetric Encryption and message Confidentiality

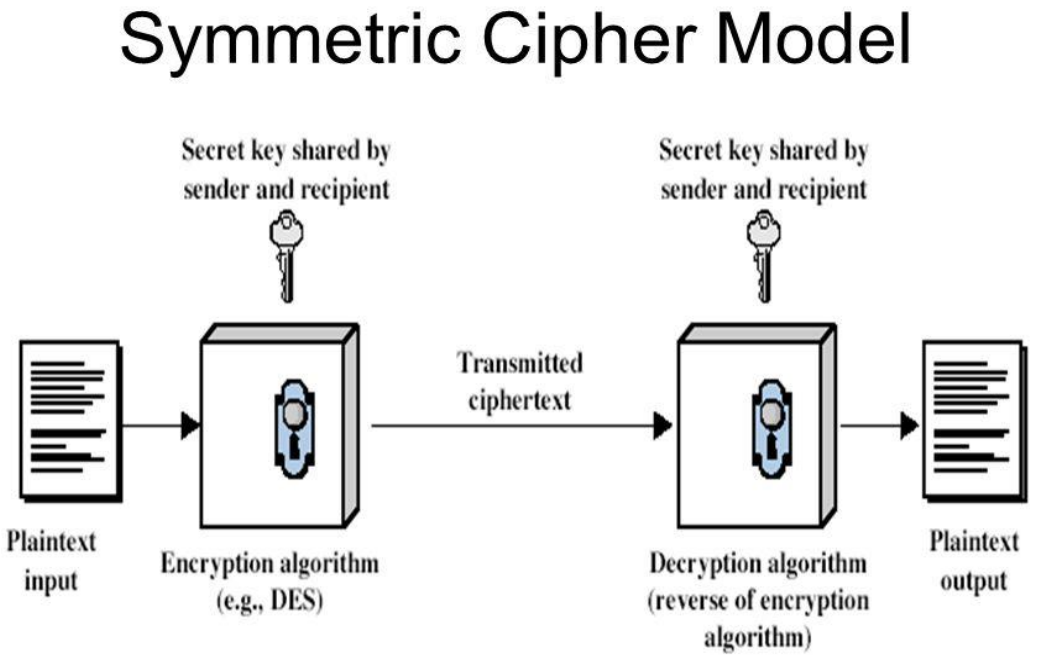
Symmetric Encryption and message Confidentiality

- Symmetric Encryption Principles
- Symmetric Block Encryption Algorithms
- Cipher Block Modes of Operations

- Encryption : A main information of message is been converted into a cipher text or unreadable form, so that form is called as an Encryption
- Symmetric means with a help of single key a message is converted to unreadable form

Symmetric Encryption Principles

- A Symmetric Encryption Principles has five components
- Plain Text - This is the original message / data that is fed into the algorithms as an input
- Encryption Algorithm - It performs various substitution and transformation on the plain text
- Secret key - It is given to algorithms. The exact substitution and transformation performed by the algorithm depends on key
- Cipher text - This is scramble message produce as output
- Decryption Algorithm - This is essentially a reverse of the encryption algorithm. It takes cipher text and same key and produce the plain text



Simplified model of conventional encryption 5

Requirements

- Two requirements for secure use of symmetric encryption:

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

- Mathematically have :
  - Y=E(K,X)
  - X=D(K,Y)
- Assume encryption algorithm is known, but we need to keep only the key secret
- Implies a secure channel to distribute key

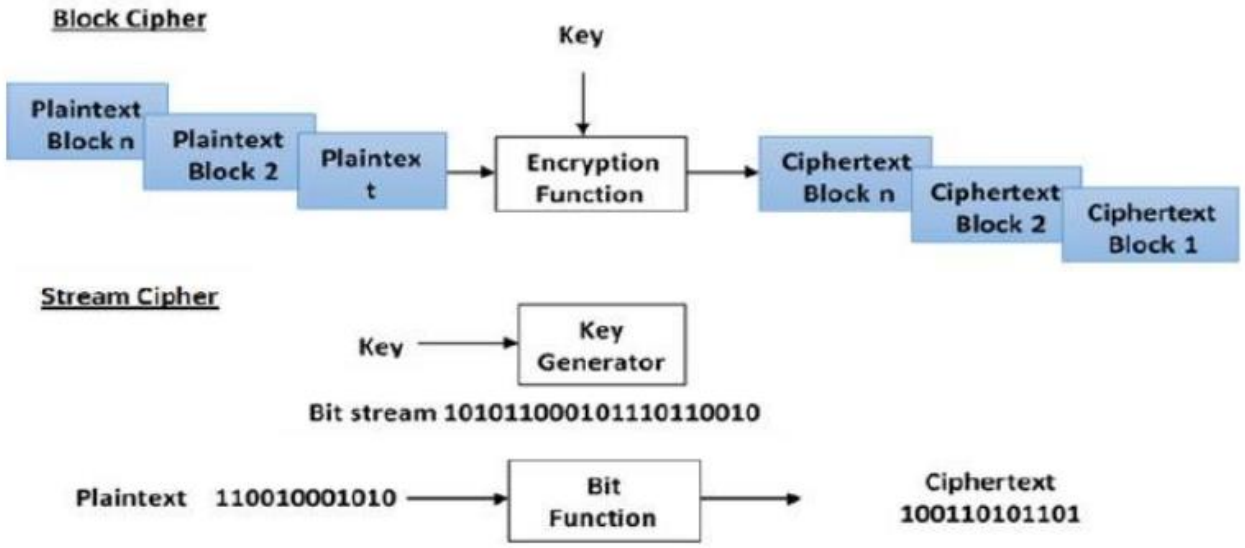
Cryptography

It is nothing but the study of encryption principles/methods, it can characterize by

Types of encryption operations used

- Substitution (plain text = DTE cipher text = AXM)
- Transposition (Plain text = DTE cipher text =TED)
- product

- Number of keys used
  - single-key or private
  - Two-key or public
- Way in which plaintext is processed
  - Block
  - Stream



Cryptanalysis

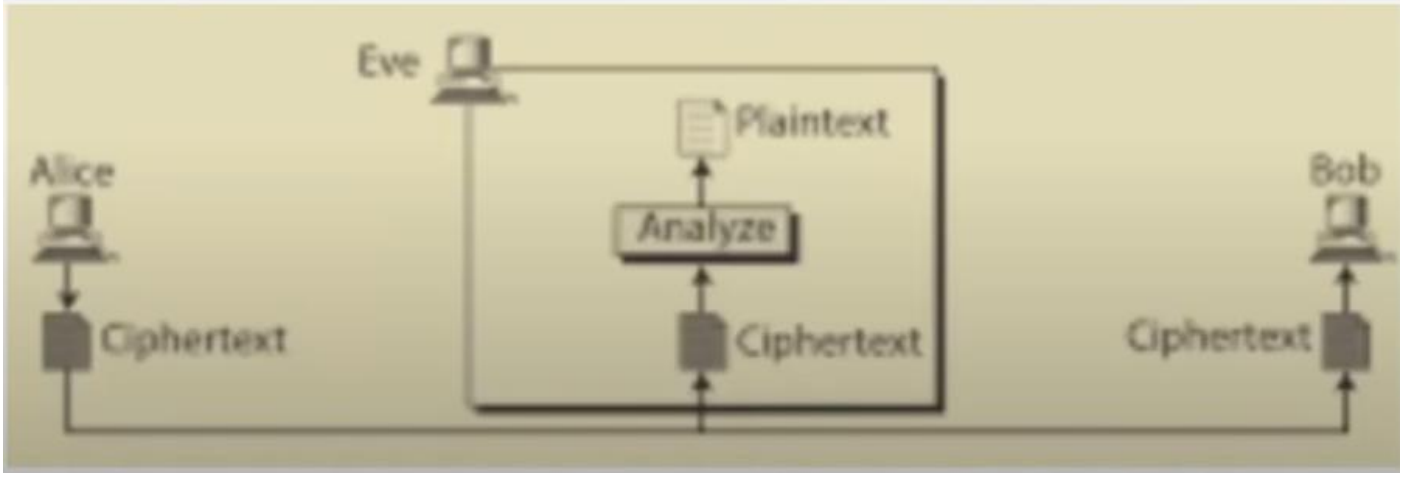
- It is a science of recovering plaintext of the message without having access to key
- Objective to recover key not just message
- General approaches to recover the plaintext
  - cryptanalytic attack: it rely on the nature of the algorithm, and some knowledge of the general characteristics of the pain text or even some sample plaintext-cipher text pairs
  - brute-force attack : it is the one that doesn't use any intelligence and enumerate all possibilities
- if either succeed all key use compromised

Types of Cryptanalytic Attacks

1) Chipher text only

It is a case in which only the encrypted message is available for attack.C1=EK(P1), C2=Ek(P2)

Task : To Fine the Plain Text and Key



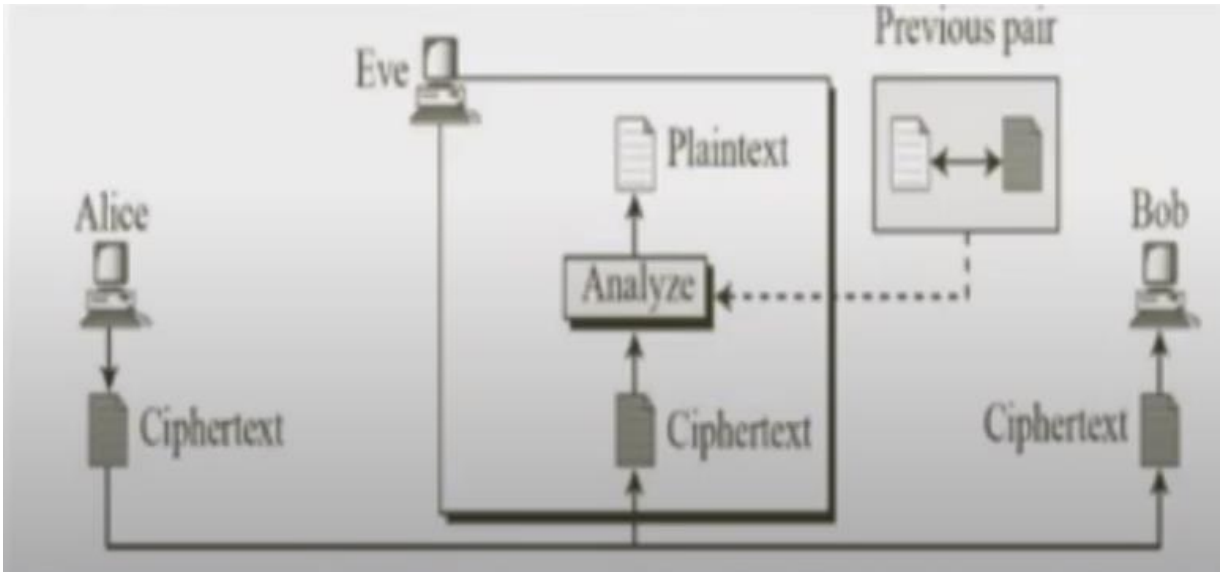
Types of Cryptanalytic Attacks

2) Known Plaintext

The attacker knows or can guess the plaintext for some part of the ciphertext

P1C1=Ek(P1), P2C2=EK(P2)

Task: To find the Key



Types of Cryptanalytic Attacks

3) Chosen Plaintext

Select plaintext and obtain ciphertext this attack occur when attacker gain access to the target encryption device.

P1C1=EK(P1) P2C2=EK(P2) where p1 or p2 can be chosen

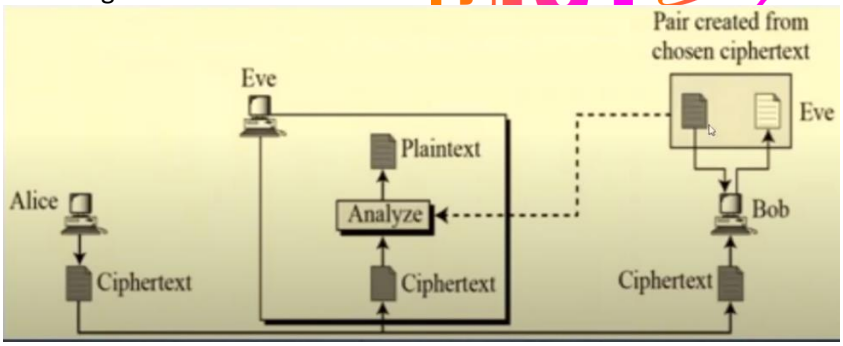
Task: To find the key

4) Chosen Ciphertext

Select ciphertext anb obtain plaintext

C1P1=DK(C1) C2P2=DK(C2)

Task: To find the key



Types of Cryptanalytic Attacks

5) Chosen Text
Select plaintext or ciphertext to en/decrypt
P1C1=EK(P1) P2C2=EK(P2)
C1P1=DK(C1) C2P2=DK(C2)
An encryption scheme: Computationally secure if
The cost of breaking the cipher exceeds the value of information
The time required to break the cipher exceeds the life time of information.

Brute Force Search

- Always possible to simply try every key
- Most basic attack, proportional to key size
- Assume either know / recognize plaintext

Feistel Cipher Structure

- Feistel Structure was described by Horst Feistel of IBM in 1973
- based on concept of invertible product cipher
- it is an example of the general structure used by all symmetric block cipher
- Partitions input block into two halves
- process through multiple rounds which
- perform a substitution on left data half
- based on round function of right half and subkey
- then have permutation swapping halves

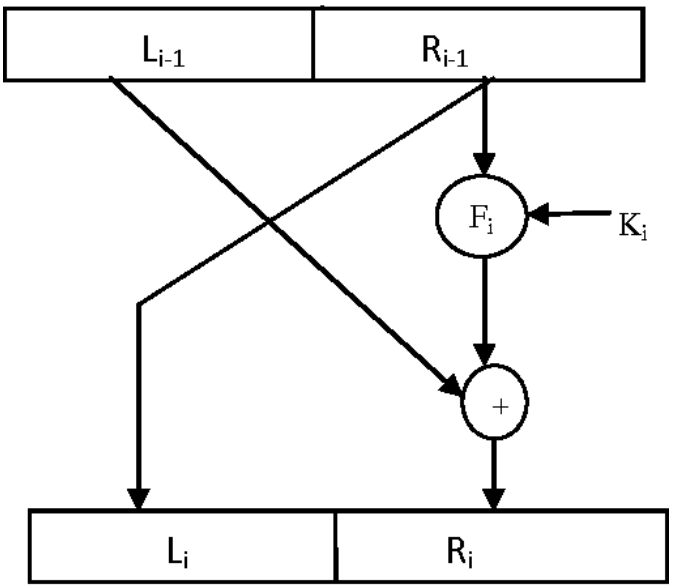
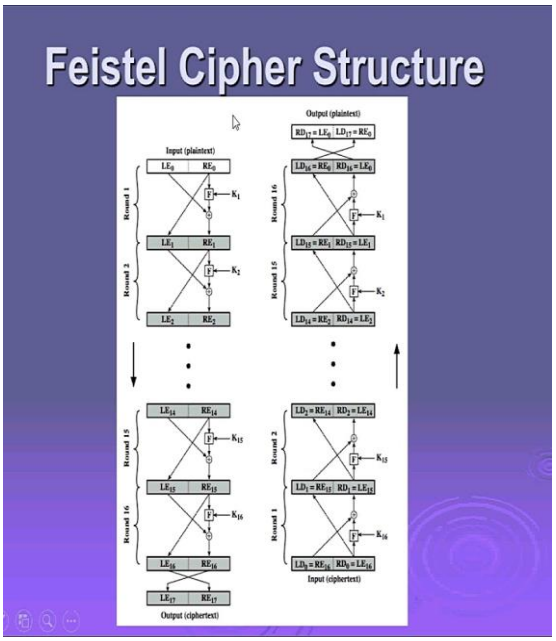


Fig 2: Single Round in Feistel cipher Structure



Feistel cipher Design Elements

- Block size : 128 bits
- key size : 128 bits
- Number of rounds : 16
- Sub key generation algorithm

Round function

The addition two consideration are

- Fast software en/decryption
- Ease of analysis

Symmetric Block Cipher Algorithms

A block cipher processes the plaintext into fixed sized blocks and produce block of ciphertext of equal size for each plaintext block.

The three most important symmetric block cipher are

- DES (Data Encryption Standard)
  3DES (Triple DES)
- AES (Advanced Encryption Standard)

Strength of DES

- The strength of DES fall into two categories
  Concerns about the algorithms
- Concerns about the use of a 56 bit key

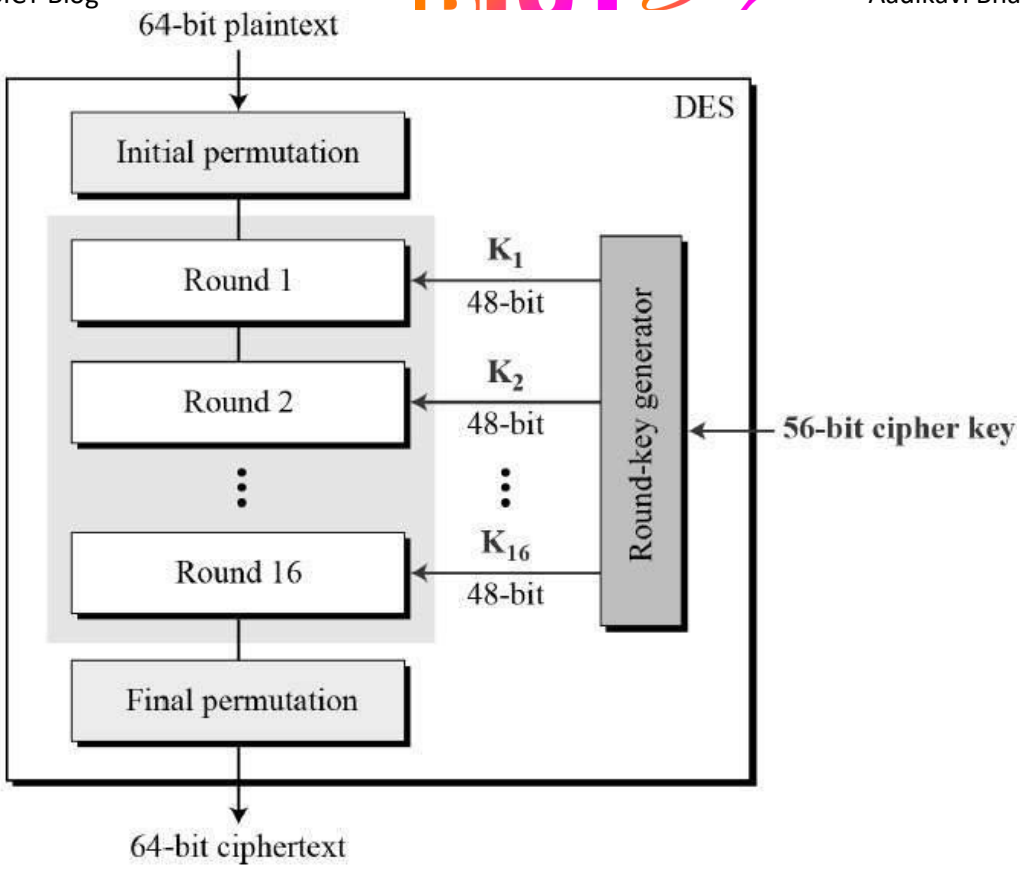
In the first concern the cryptanalysis tries numerous attempts to find and exploit the weakness in the algorithm, but couldn't succeed.

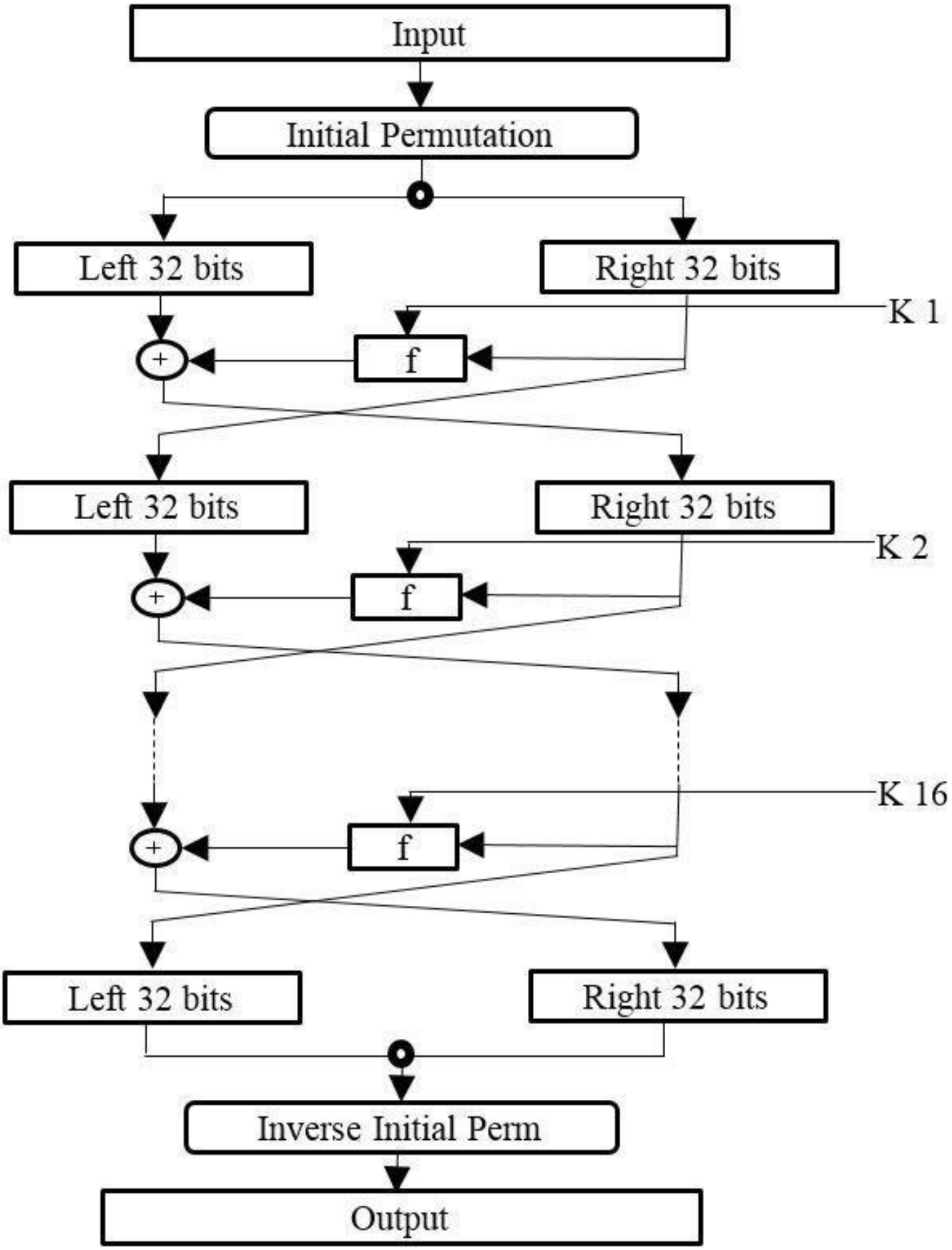
The second concern is key length, with the 56 bit key length there are 256 possible keys, which is approximately 7.2*1016 keys.

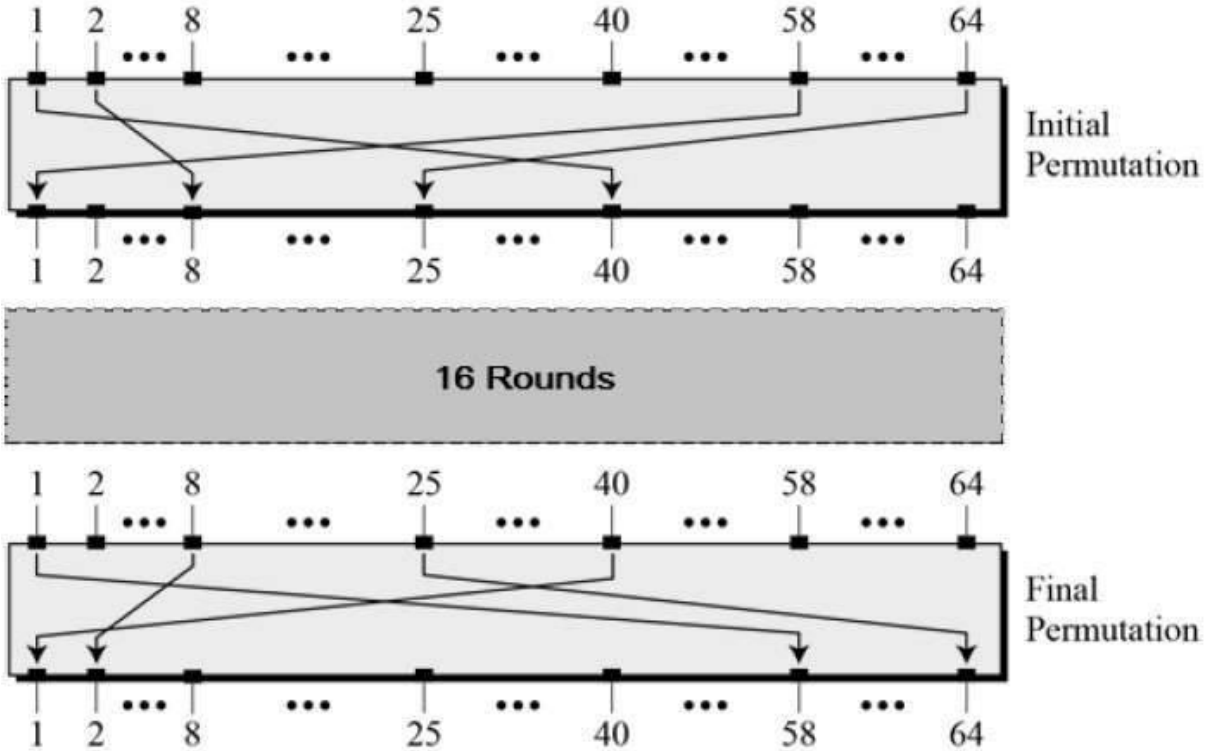
DES finally and definitively proved insecure in July 1998 when EFF announce that is has broken DES encryption using a special purpose "DES Cracker" machine

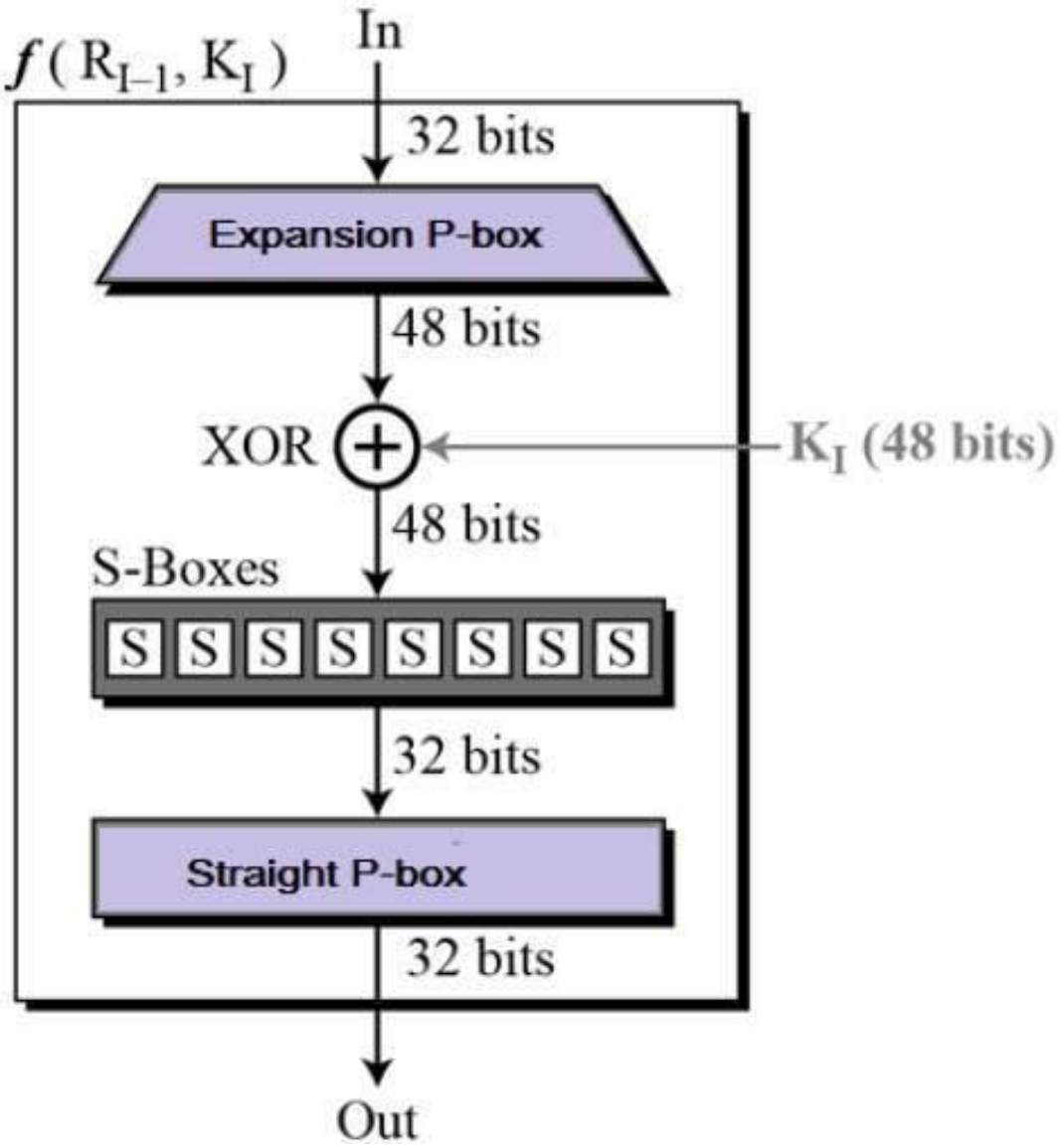
Data Encryption Standard (DES)

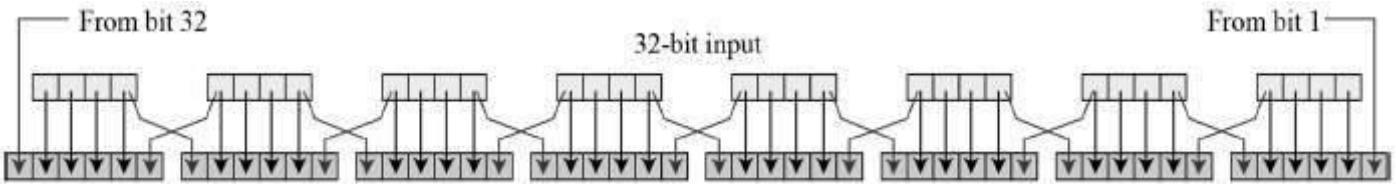
- It follows Feistel Cipher structure
- Block size : 64 bit
- No of Round : 16 round
- Key size : Actually 64 bit it will convert into 56 bit
- No of sub keys 16 Sub Keys
- No of Sub size : 48 bit sub key size
- Cipher text : 64 bit


64-bit plaintext
DES
Initial permutation
Round 1
Round 2
Round 16
Final permutation
K1
48-bit
K2
48-bit
K16
48-bit
Round-key generator
56-bit cipher key
64-bit ciphertext


Input
Initial Permutation
Left 32 bits
Right 32 bits
K 1
f
Left 32 bits
Right 32 bits
K 2
f
K 16
f
Left 32 bits
Right 32 bits
Inverse Initial Perm
Output

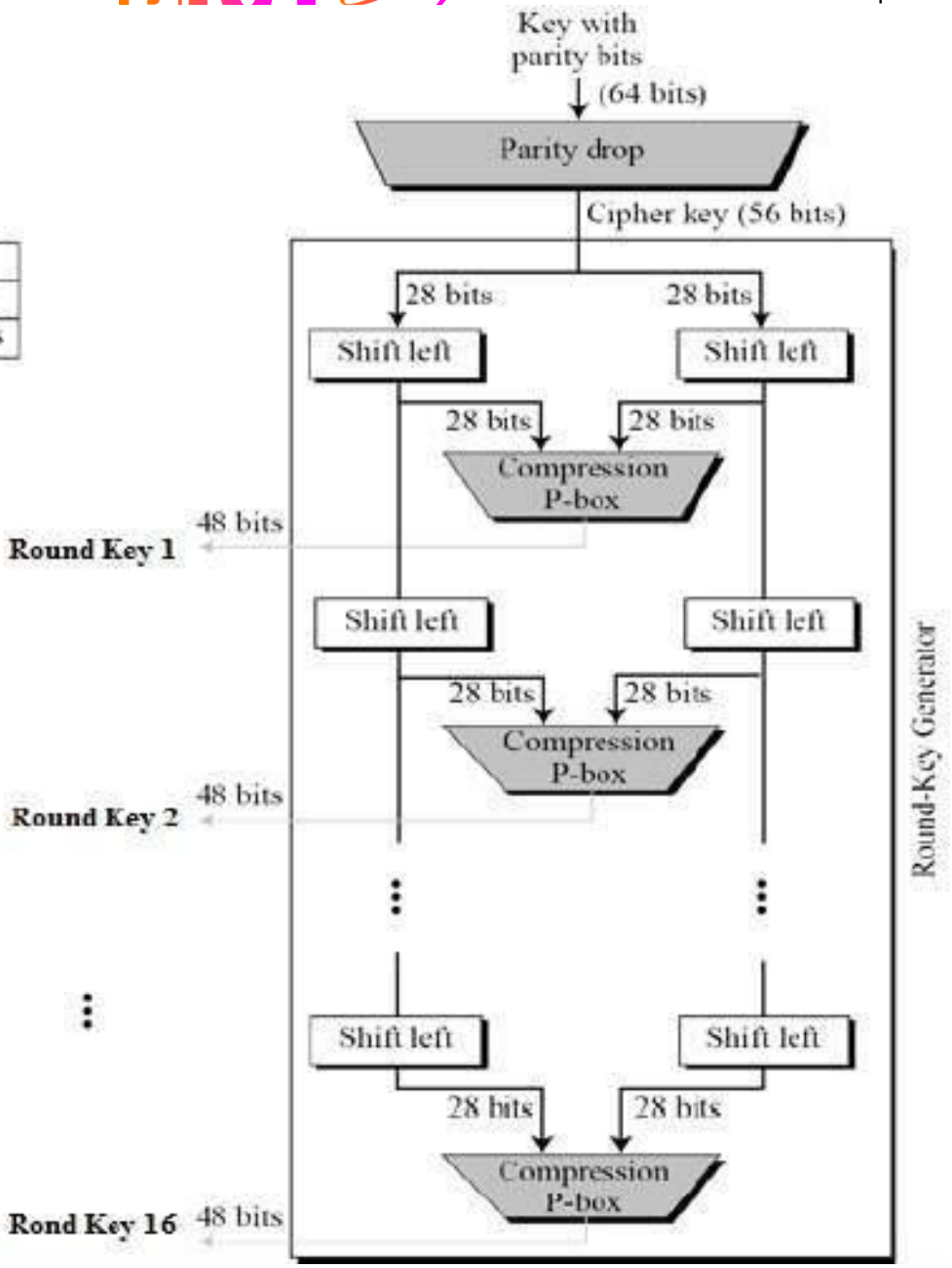

1 2 8 25 40 58 64
Initial Permutation
1 2 8 25 40 58 64
16 Rounds
1 2 8 25 40 58 64
Final Permutation
1 2 8 25 40 58 64

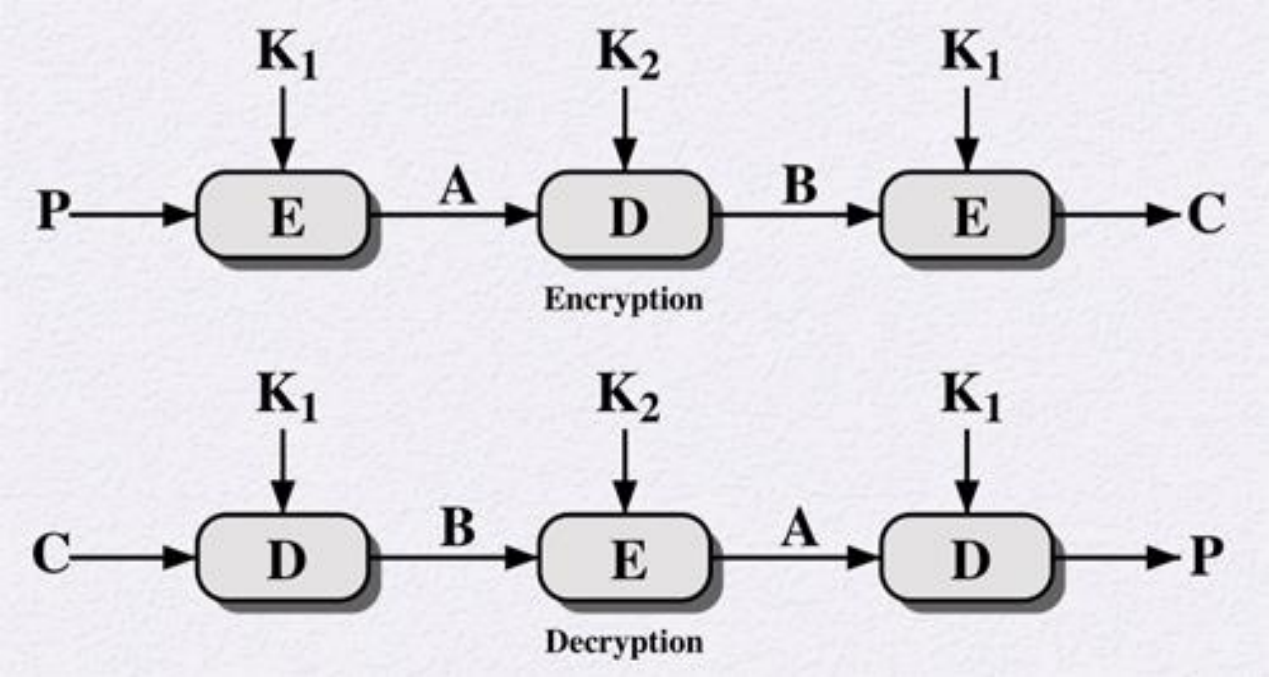

f (R I–1, K I)
In
32 bits
Expansion P-box
48 bits
XOR
K I (48 bits)
48 bits
S-Boxes
S S S S S S S S
32 bits
Straight P-box
32 bits
Out


From bit 32
32-bit input
From bit 1

Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits





Triple-DES with Three-Keys

- Disadvantage of DES is the key length size as it is 56 bit it is very easy for cryptanalyst to break the key, so to overcome this 3DES algorithm was introduce
- 3DES has effectively key length of 168 bit and 64 bit block size
- To encrypt : C = E(K3,D(k2,E(K1,P)))
- To Decrypt : p =D(K1,E(K2,D(K3,C)))

Origins of AES

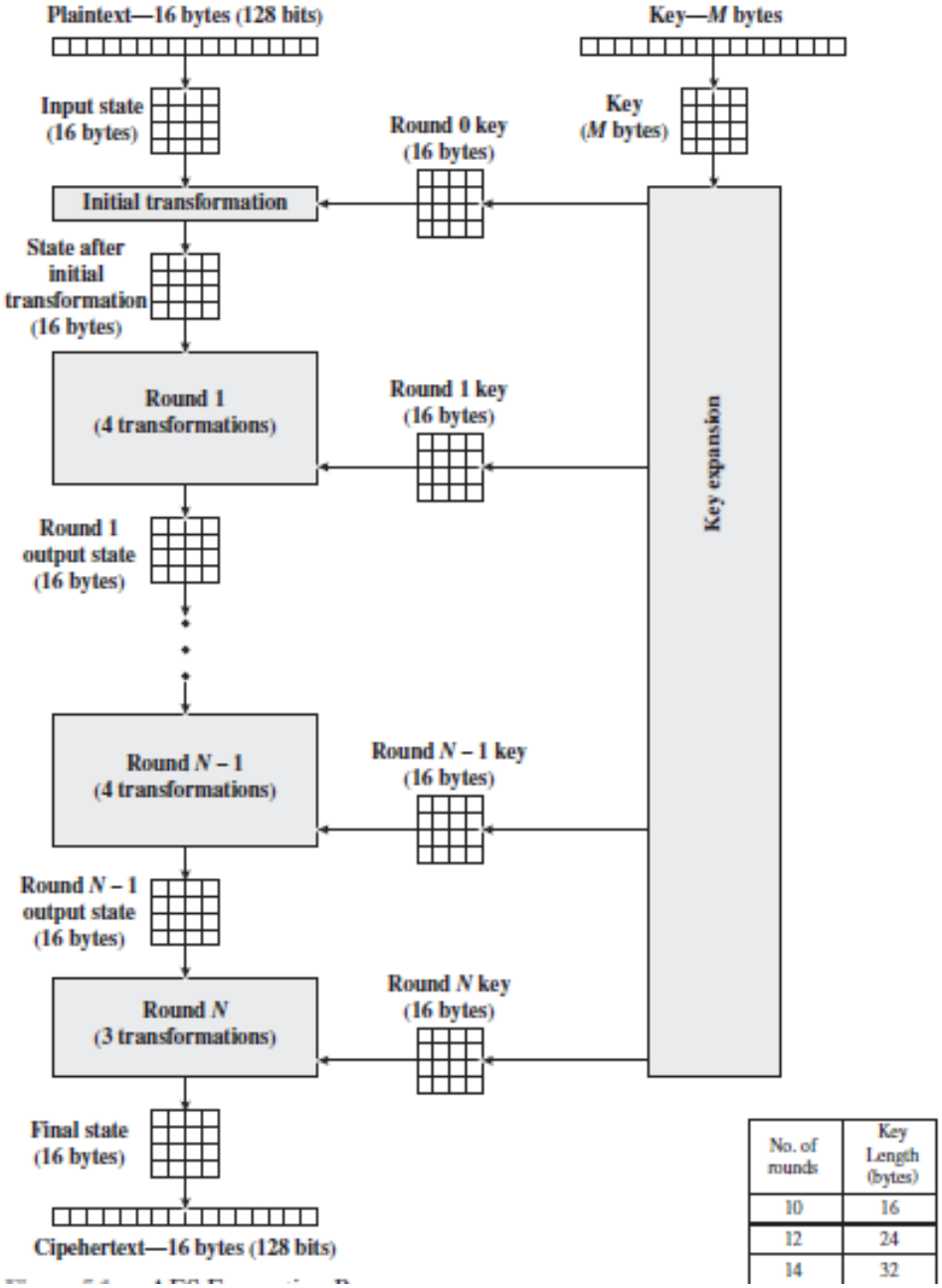
Clearly a replacement for DES was needed

- Have theoretical attacks that can break it
- Have demonstrate exhaustive key search attacks
- can use Triple-DES - but slow, has small blocks
- US NIST issued call for ciphers in 1997
- 15 candidates accepted in jun 98
- 5 were shortlisted in Aug 99
- Rijndael was selected as the AES in Oct-2000
- issued as FIPS PUB 197 standard in Nov -2001

The AES Cipher - Rijndael

- Design by Dr. Joan Daemen and Dr. Vincent Rijmen
- It has 128/192/256 bit keys, 128 bit block data
- it is an iterative rather than feistel cipher
- processes data as bock of 4 columns of 4 bytes
- operates on entire data block in every round
- it is designed to be
- resistant against known attacks
- speed and code compactness on many CPUs

- design simplicity



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

AES

Encryption Process

AES Structure

- Data block of 4 columns of 4 bytes is state
- key is expanded to array of words
- it has 9/11/13 rounds in which state undergoes
  - byte substitution (1 S-box used on every byte)
  - shift rows (permute bytes between groups/columns)
  - mix columns (subs using matrix multiply of groups)
  - add round key (XOR state with key material)
  - view as alternating XOR key & scramble data bytes

Cipher Block Modes of Operations

A block cipher processes the data blocks of fixed size. Usually, the size of a message is larger than the block size. Hence, the long message is divided into a series of sequential message blocks, and the cipher operates on these blocks one at a time.

- Electronic Code Book (ECB) Mode
- Cipher Block Chaining (CBC) Mode
- Cipher Feedback (CFB) Mode
- Output Feedback (OFB) Mode
- Counter (CTR) Mode

Electronic Code Book (ECB) Mode

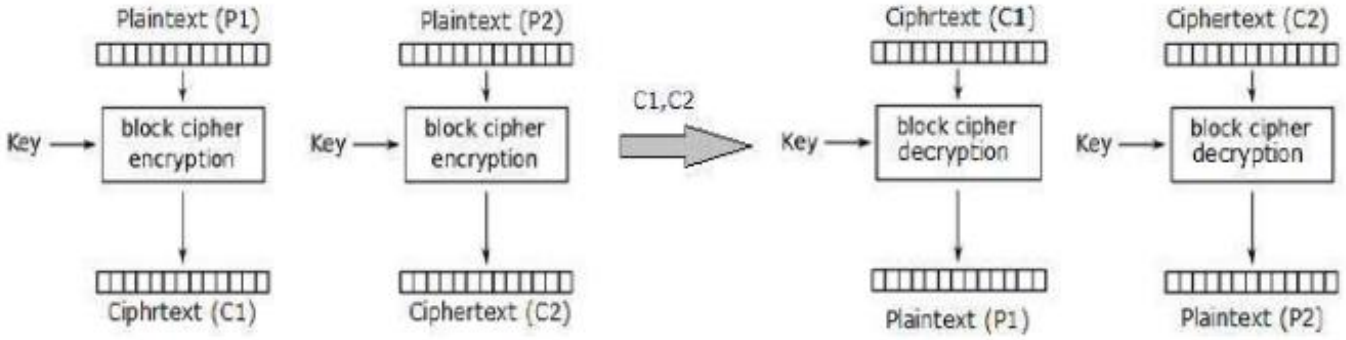
- This mode is a most straightforward way of processing a series of sequentially listed message blocks.

Operation

- The user takes the first block of plaintext and encrypts it with the key to produce the first block of ciphertext.
- He then takes the second block of plaintext and follows the same process with same key and so on so forth.
- The ECB mode is deterministic, that is, if plaintext block P1, P2,…, Pm are encrypted twice under the same key, the output ciphertext blocks will be the same.

In fact, for a given key technically we can create a codebook of ciphertexts for all possible plaintext blocks. Encryption would then entail only looking up for required plaintext and select the corresponding ciphertext. Thus, the operation is analogous to the assignment of code words in a codebook, and hence gets an official name − Electronic Codebook mode of operation (ECB).

It is illustrated as follows –



Analysis of ECB Mode

In reality, any application data usually have partial information which can be guessed. For example, the range of salary can be guessed. A ciphertext from ECB can allow an attacker to guess the plaintext by trial-and- error if the plaintext message is within predictable.

For example, if a ciphertext from the ECB mode is known to encrypt a salary figure, then a small number of trials will allow an attacker to recover the figure. In general, we do not wish to use a deterministic cipher, and hence the ECB mode should not be used in most applications.

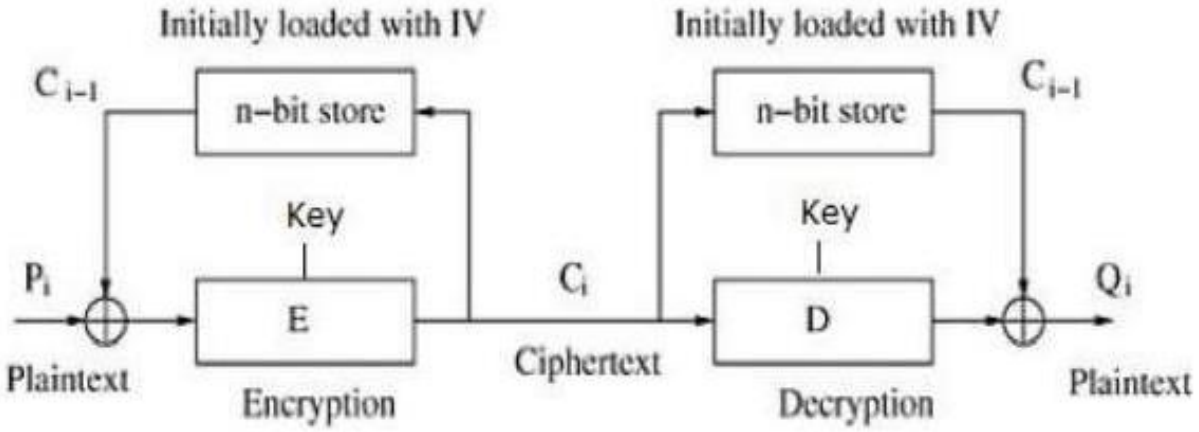
Cipher Block Chaining (CBC) Mode

CBC mode of operation provides message dependence for generating ciphertext and makes the system non- deterministic.

Operation :

- Load the n-bit Initialization Vector (IV) in the top register.

- XOR the n-bit plaintext block with data value in top register.
- Encrypt the result of XOR operation with underlying block cipher with key K.
- Feed ciphertext block into top register and continue the operation till all plaintext blocks are processed.
- For decryption, IV data is XORed with first ciphertext block decrypted. The first ciphertext block is also fed into to register replacing IV for decrypting next ciphertext block.



Analysis of CBC Mode

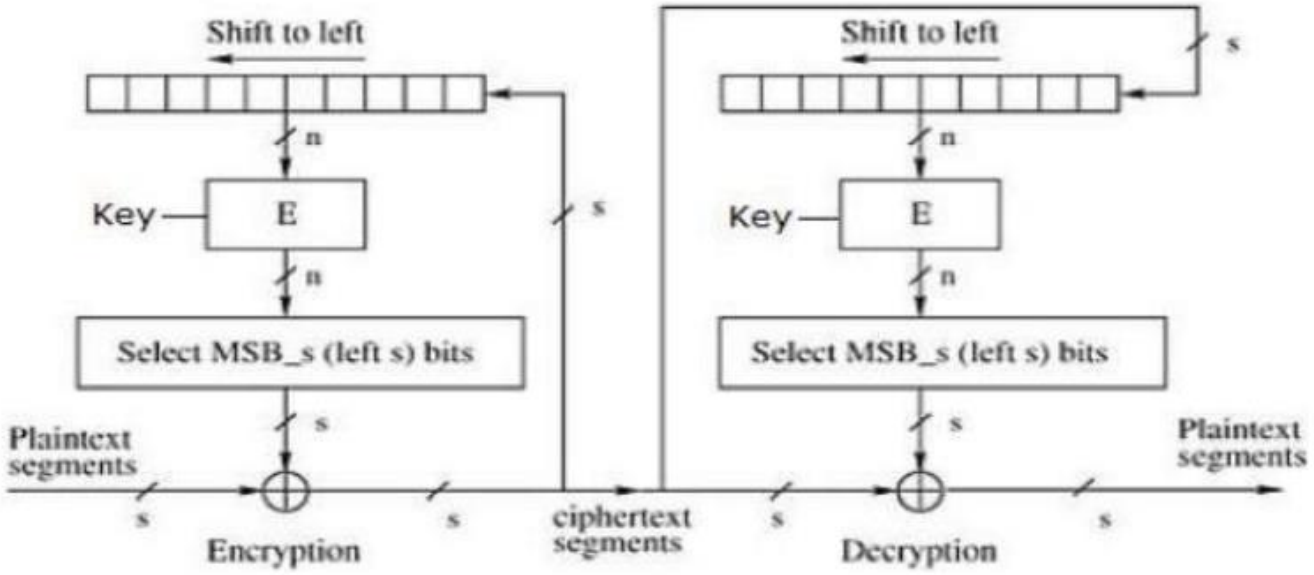
- In CBC mode, the current plaintext block is added to the previous ciphertext block, and then the result is encrypted with the key. Decryption is thus the reverse process, which involves decrypting the current ciphertext and then adding the previous ciphertext block to the result.
- Advantage of CBC over ECB is that changing IV results in different ciphertext for identical message. On the drawback side, the error in transmission gets propagated to few further block during decryption due to chaining effect.
- It is worth mentioning that CBC mode forms the basis for a well-known data origin authentication mechanism. Thus, it has an advantage for those applications that require both symmetric encryption and data origin authentication.

Cipher Feedback (CFB) Mode

In this mode, each ciphertext block gets ‘fed back’ into the encryption process in order to encrypt the next plaintext block.

Operation

- The operation of CFB mode is depicted in the following illustration. For example, in the present system, a message block has a size ‘s’ bits where 1 < s
- < n. The CFB mode requires an initialization vector (IV) as the initial random n-bit input block. The IV need not be secret. Steps of operation are −
- Load the IV in the top register.
- Encrypt the data value in top register with underlying block cipher with key K.
- Take only ‘s’ number of most significant bits (left bits) of output of encryption process and XOR them with ‘s’ bit plaintext message block to generate ciphertext block.
- Feed ciphertext block into top register by shifting already present data to the left and continue the operation till all plaintext blocks are processed.
- Essentially, the previous ciphertext block is encrypted with the key, and then the result is XORed to the current plaintext block.
- Similar steps are followed for decryption. Pre-decided IV is initially loaded at the start of decryption.



Analysis of CFB Mode

- CFB mode differs significantly from ECB mode, the ciphertext corresponding to a given plaintext block depends not just on that plaintext block and the key, but also on the previous ciphertext block. In other words, the ciphertext block is dependent of message.
- CFB has a very strange feature. In this mode, user decrypts the ciphertext using only the encryption process of the block cipher. The decryption algorithm of the underlying block cipher is never used.
- Apparently, CFB mode is converting a block cipher into a type of stream cipher. The encryption algorithm is used as a key-stream generator to produce key-stream that is placed in the bottom register. This key stream is then XORed with the plaintext as in case of stream cipher.
- By converting a block cipher into a stream cipher, CFB mode provides some of the advantageous properties of a stream cipher while retaining the advantageous properties of a block cipher.
- On the flip side, the error of transmission gets propagated due to changing of blocks.

Output Feedback (OFB) Mode

- It involves feeding the successive output blocks from the underlying block cipher back to it. These feedback blocks provide string of bits to feed the encryption algorithm which act as the key-stream generator as in case of CFB mode.
- The key stream generated is XOR-ed with the plaintext blocks. The OFB mode requires an IV as the initial random n-bit input block. The IV need not be secret.
- The operation is depicted in the following illustration

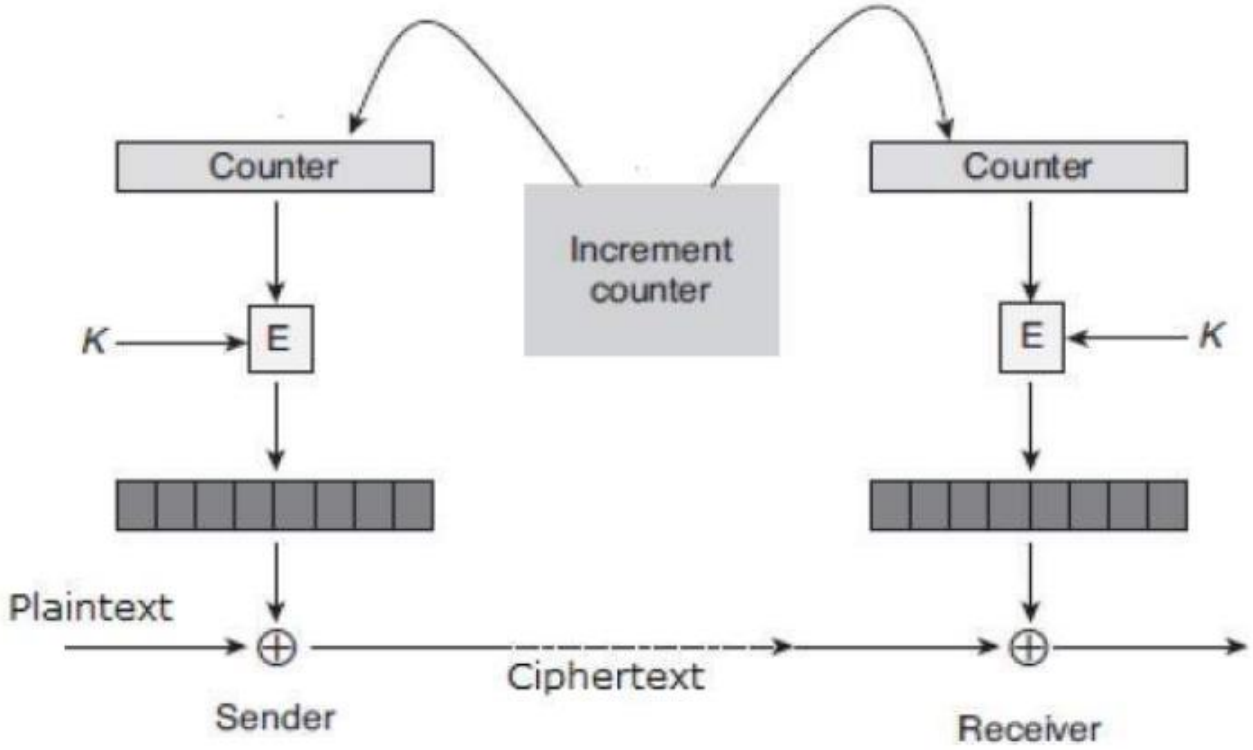
Counter (CTR) Mode

It can be considered as a counter-based version of CFB mode without the feedback. In this mode, both the sender and receiver need to access to a reliable counter, which computes a new shared value each time a ciphertext block is exchanged. This shared counter is not necessarily a secret value, but challenge is that both sides must keep the counter synchronized.

Operation

- Both encryption and decryption in CTR mode are depicted in the following

- illustration. Steps in operation are −
- Load the initial counter value in the top register is the same for both the sender and the receiver. It plays the same role as the IV in CFB (and CBC) mode.
- Encrypt the contents of the counter with the key and place the result in the bottom register.
- Take the first plaintext block P1 and XOR this to the contents of the bottom register. The result of this is C1. Send C1 to the receiver and update the counter. The counter update replaces the ciphertext feedback in CFB mode.
- Continue in this manner until the last plaintext block has been encrypted.
- The decryption is the reverse process. The ciphertext block is XORed with the output of encrypted contents of counter value. After decryption of each ciphertext block counter is updated as in case of encryption.



Analysis of Counter Mode

- It does not have message dependency and hence a ciphertext block does not depend on the previous plaintext blocks.
- Like CFB mode, CTR mode does not involve the decryption process of the block cipher. This is because the CTR mode is really using the block cipher to generate a key-stream, which is encrypted using the XOR function. In other words, CTR mode also converts a block cipher to a stream cipher.
- The serious disadvantage of CTR mode is that it requires a synchronous counter at sender and receiver. Loss of synchronization leads to incorrect recovery of plaintext.
- However, CTR mode has almost all advantages of CFB mode. In addition, it does not propagate error of transmission at all.

UNIT 3

Asymmetric Encryption / Public key Encryption and Message Digest

There are two types for encryption

- Symmetric key encryption and
- Public key encryption / Asymmetric Key encryption

Symmetric Key Encryption

- Symmetric key Encryption uses same key to encrypt data and decrypt data
- It is easiest to understand
- Faster compared to public key encryption
- Problem
- Key needs to be stored securely
- Secured channel is required to transfer the key

Public Key Encryption

- Public key encryption uses two key Private and Public key
- Slower as compare to symmetric key

A public-key encryption scheme

- Plaintext: This is the readable message or data that is fed into the algorithm as input.
- Encryption algorithm: The encryption algorithm performs various transformations on the plaintext.
- Public and private keys: This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.
- Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
- Decryption algorithm: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

Hash functions

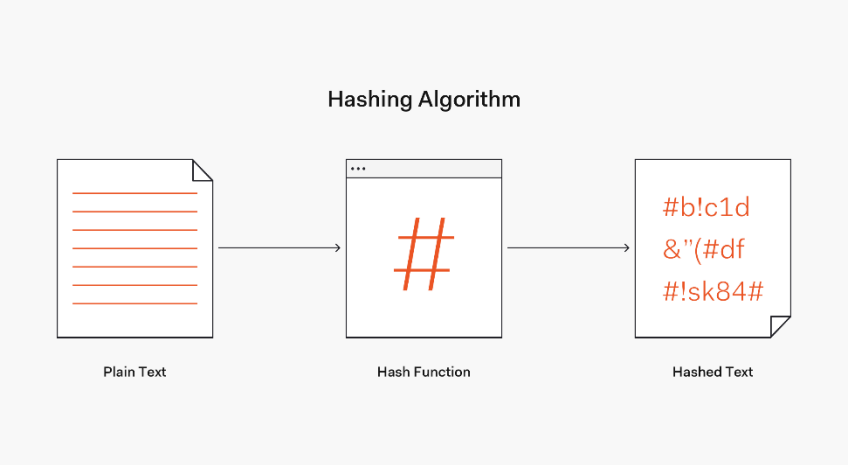
Similar to Message Authentication Code (MAC)

Takes in variable size message and produce a fixed size output

Hash functions are extremely useful and appear in almost all information security applications.

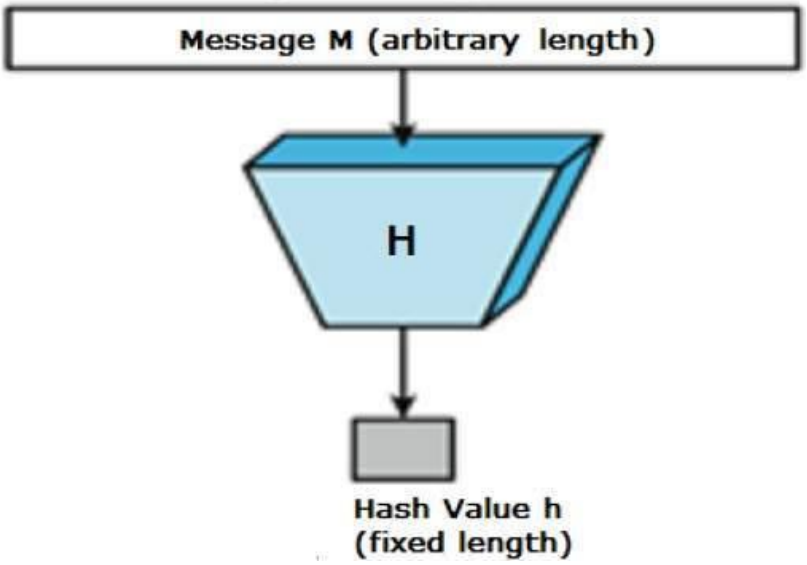
A hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The input to the hash function is of arbitrary length but output is always of fixed length.

Values returned by a hash function are called message digest or simply hash values. The following picture illustrated hash function −



Features of Hash Functions

- The typical features of hash functions are −
- Fixed Length Output (Hash Value)
  Efficiency of Operation



Fixed Length Output (Hash Value)

- Hash function coverts data of arbitrary length to a fixed length. This process is often referred to
- as hashing the data.
- In general, the hash is much smaller than the input data, hence hash functions are sometimes
- called compression functions.
- Since a hash is a smaller representation of a larger data, it is also referred to as a digest.Hash function with n bit output is referred to as an n-bit hash function. Popular hash functions generate values between 160 and 512 bits.

Efficiency of Operation

- Generally, for any hash function h with input x, computation of h(x) is a fast operation.
- Computationally hash functions are much faster than a symmetric encryption.

Properties of Hash Functions

● Pre-Image Resistance

 This property means that it should be computationally hard to reverse a hash function.

 In other words, if a hash function h produced a hash value z, then it should be a difficult process to find any input value x that hashes to z.

 This property protects against an attacker who only has a hash value and is trying to find the input.

● **Second Pre-Image Resistance**

 This property means given an input and its hash, it should be hard to find a different input with the same hash.

 In other words, if a hash function h for an input x produces hash value h(x), then it should be difficult to find any other input value y such that h(y) = h(x).

 This property of hash function protects against an attacker who has an input value and its hash, and wants to substitute different value as legitimate value in place of original input value.

● **Collision Resistance**

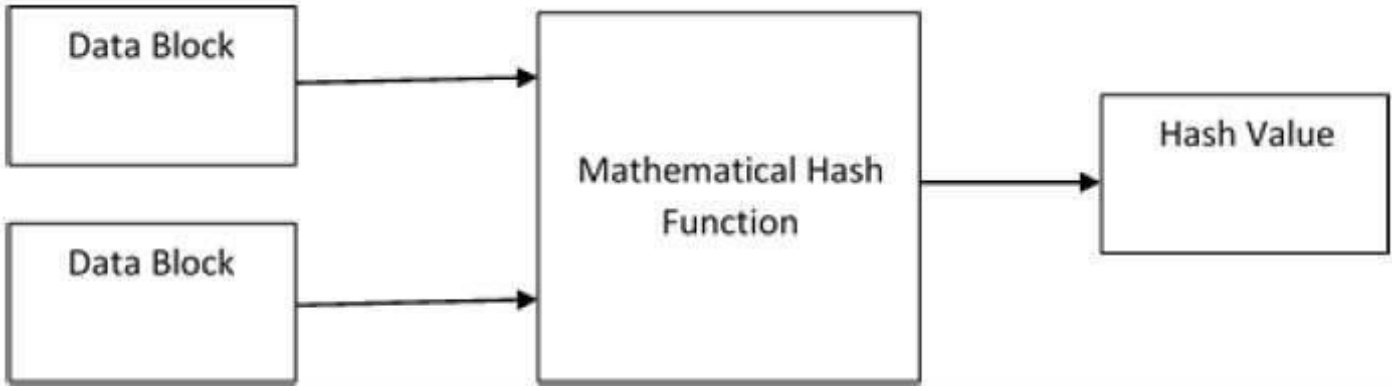
 This property means it should be hard to find two different inputs of any length that result in the same hash. This property is also referred to as collision free hash function.

 In other words, for a hash function h, it is hard to find any two different inputs x and y such that h(x) = h(y).

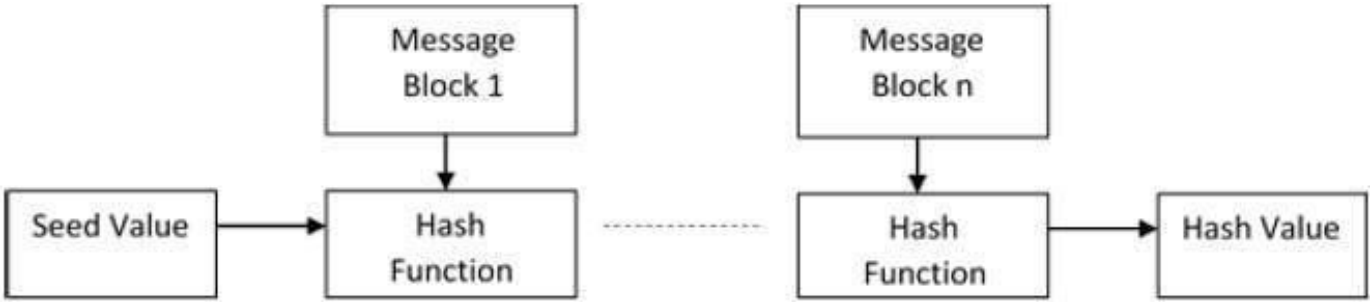
- Since, hash function is compressing function with fixed hash length, it is impossible for a hash function not to have collisions. This property of collision free only confirms that these collisions should be hard to find.
- This property makes it very difficult for an attacker to find two input values with the same hash.
- Also, if a hash function is collision-resistant then it is second pre- image resistant.

Design of Hashing Algorithms

- At the heart of a hashing is a mathematical function that operates on two fixed-size blocks of data to create a hash code. This hash function forms the part of the hashing algorithm.
- The size of each data block varies depending on the algorithm. Typically the block sizes are from 128 bits to 512 bits. The following illustration demonstrates hash function −



- Hashing algorithm involves rounds of above hash function like a block cipher. Each round takes an input of a fixed size, typically a combination of the most recent message block and the output of the last round.
- This process is repeated for as many rounds as are required to hash the entire message. Schematic of hashing algorithm is depicted in the following illustration –



Since, the hash value of first message block becomes an input to the second hash operation, output of which alters the result of the third operation, and so on. This effect, known as an avalanche effect of hashing.

Avalanche effect results in substantially different hash values for two messages that differ by even a single bit of data.

Popular Hash Functions

- MD5 was most popular and widely used hash function for quite some years.
- The MD family comprises of hash functions MD2, MD4, MD5 and MD6. It was adopted as Internet Standard RFC 1321. It is a 128-bit hash function.
- MD5 digests have been widely used in the software world to provide assurance about integrity of transferred file. For example, file servers often provide a pre-computed MD5

checksum for the files, so that a user can compare the checksum of the downloaded file to it.

- In 2004, collisions were found in MD5. An analytical attack was reported to be successful only in an hour by using computer cluster. This collision attack resulted in compromised MD5 and hence it is no longer recommended for use.

Secure Hash Function (SHA)

- Family of SHA comprise of four SHA algorithms; SHA-0, SHA-1, SHA-2, and SHA-3. Though from same family, there are structurally different.
- The original version is SHA-0, a 160-bit hash function, was published by the National Institute of Standards and Technology (NIST) in 1993. It had few weaknesses and did not become very popular. Later in 1995, SHA-1 was designed to correct alleged weaknesses of SHA-0.
- SHA-1 is the most widely used of the existing SHA hash functions. It is employed in several widely used applications and protocols including Secure Socket Layer (SSL) security.

- In 2005, a method was found for uncovering collisions for SHA-1 within practical time frame making long-term employability of SHA-1 doubtful.
- SHA-2 family has four further SHA variants, SHA-224, SHA-256, SHA-384, and SHA-512 depending up on number of bits in their hash value. No successful attacks have yet been reported on SHA-2 hash function.
- Though SHA-2 is a strong hash function. Though significantly different, its basic design is still follows design of SHA-1. Hence, NIST called for new competitive hash function designs.
- In October 2012, the NIST chose the Keccak algorithm as the new SHA-3 standard. Keccak offers many benefits, such as efficient performance and good resistance for attacks.

Applications of Hash Functions

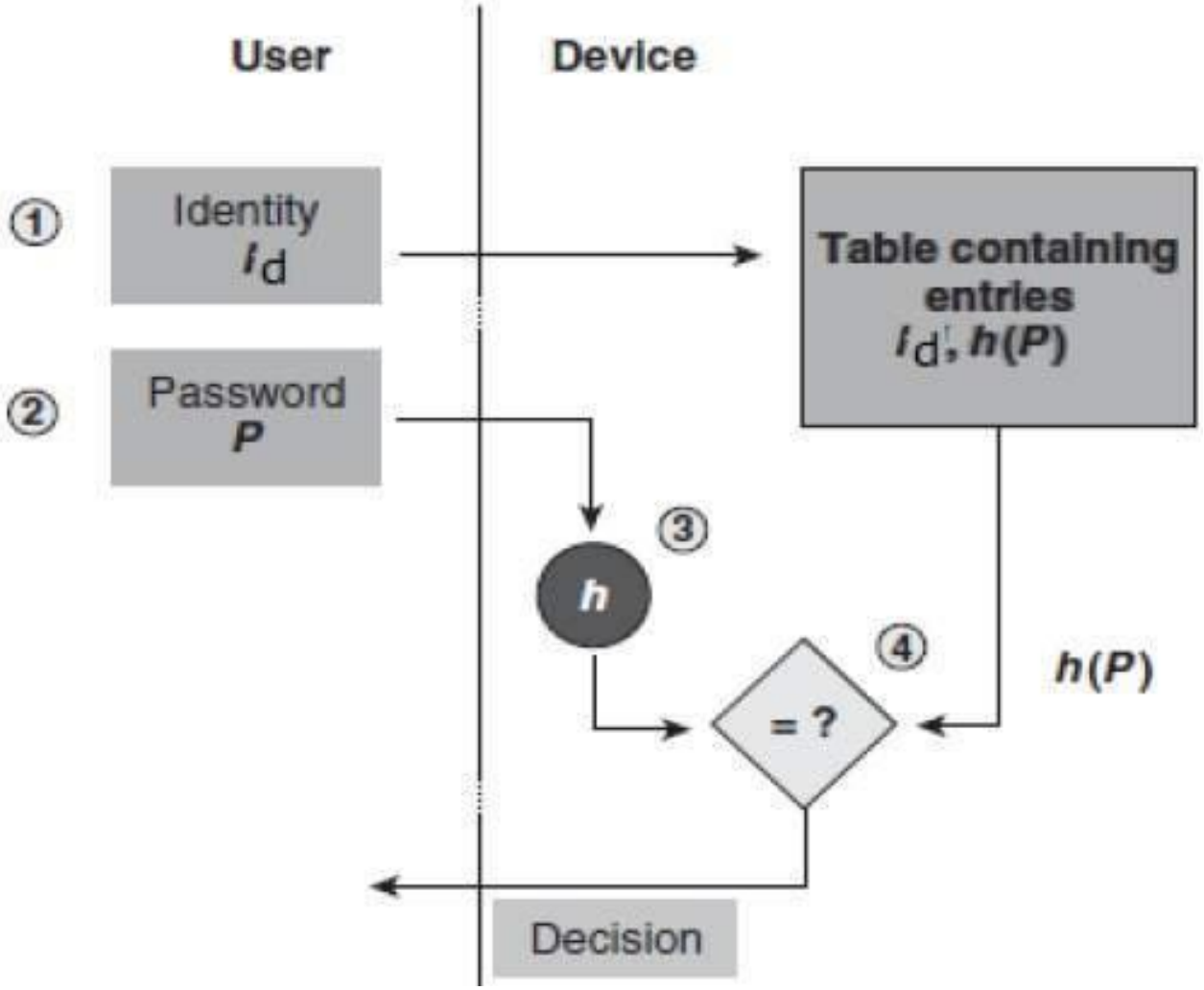
There are two direct applications of hash function based on its cryptographic properties.

- Password Storage
- Data Integrity Check

Password Storage

- Instead of storing password in clear, mostly all logon processes store the hash values of passwords in the file.
- The Password file consists of a table of pairs which are in the form (user id, h(P)).
- The process of logon is depicted in the following

illustration –



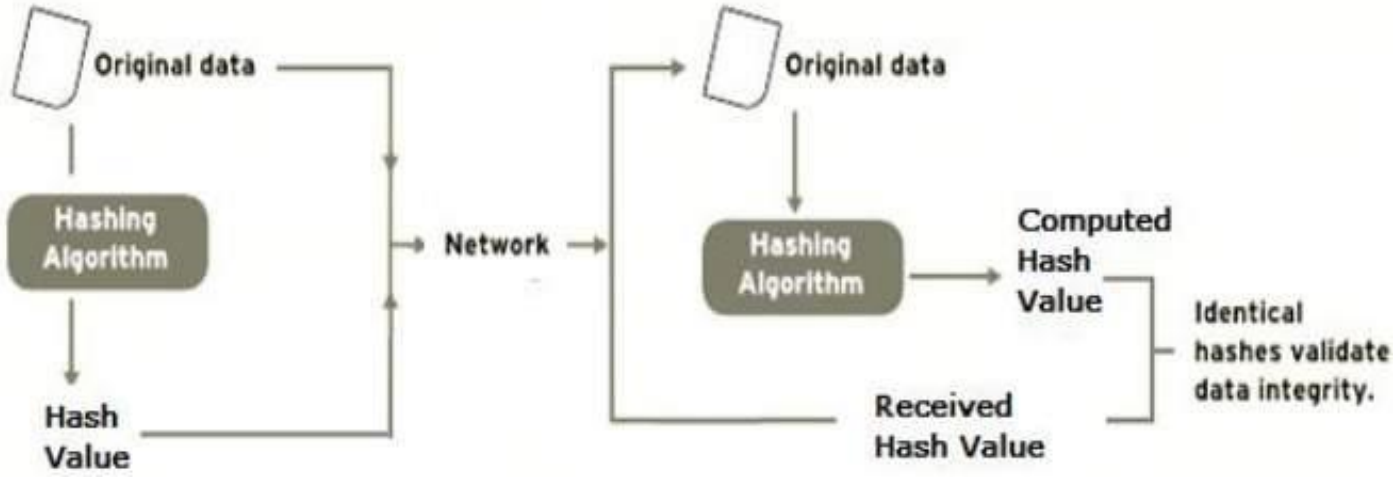
An intruder can only see the hashes of passwords, even if he accessed the password. He can neither logon using hash nor can he derive the password from hash value since hash function possesses the property of pre-image resistance.

Data Integrity Check

Data integrity check is a most common application of the hash functions. It is used to generate the checksums on data files. This application provides assurance to the user about correctness of the data.

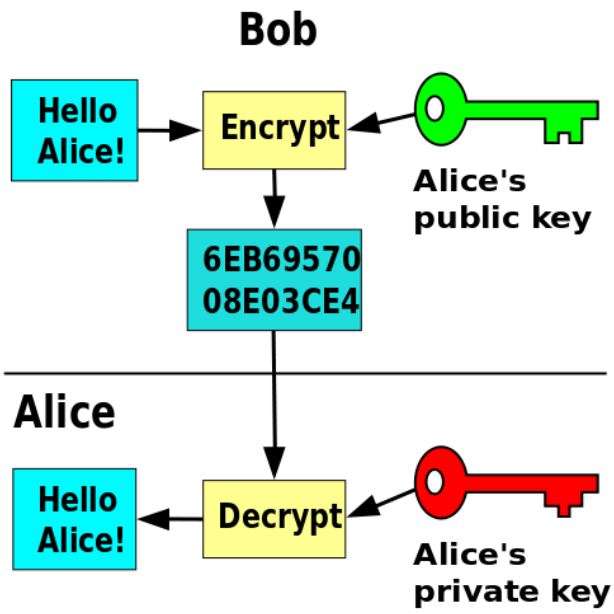
The process is depicted in the following illustration −

The integrity check helps the user to detect any changes made to original file. It however, does not provide any assurance about originality. The attacker, instead of modifying file data, can change the entire file and compute all together new hash and send to the receiver. This integrity check application is useful only if the user is sure about the originality of file.



Public-Key Cryptography Principles

- The most important properties of public key encryption scheme are −
- Different keys are used for encryption and decryption. This is a property which set this scheme different than symmetric encryption scheme.
- Each receiver possesses a unique decryption key, generally referred to as his private key.
- Receiver needs to publish an encryption key, referred to as his public key.
- Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.
- Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from the ciphertext and the encryption (public) key.
- Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key. In fact, intelligent part of any public-key cryptosystem is in designing a relationship between two keys.



The essential steps are the following:

1.Each user generates a pair of keys to be used for the encryption and decryption of messages.
2.Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private. As Figure suggests, each user maintains a collection of public keys obtained from others.
3.If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.
4.When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

Public-key systems are characterized by the use of a cryptographic algorithm with two keys, one held private and one available publicly.
Depending on the application, the sender uses either the sender's private key or the receiver's public key, or both, to perform some type of cryptographic function.

Applications for Public-Key Cryptosystems
In broad terms, we can classify the use of public-key cryptosystems into three categories:
- Encryption/decryption: The sender encrypts a message with the recipient's public key.
- Digital signature: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
- Key exchange: Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties. Some algorithms are suitable for all three applications, whereas others can be used only for one or two of these applications.

Applications for Public-Key Cryptosystems

Table 9.2. Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

Requirements of public key
It is computationally easy for a party B to generate a pair (public key PUb, private key PRb)
It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext:
C= E(PUb, M)
It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:
M= D(PRb, C) = D(PRb, E(PUb, M))
It is computationally infeasible for an opponent, knowing the public key, PUb, to determine the private key, PRb.
It is computationally infeasible for an opponent, knowing the public key, PUb, and a ciphertext, C, to recover the original message

Asymmetric Encryption
- **Two most popular algorithms are RSA & El Gamal**
- RSA (Rivest, Shamir, and Adelman)
- Developed by Ron Rivest, Adi Shamir, Len Adelman

- Both public and private key are interchangeable
- Variable Key Size (512, 1024, or 2048 buts)
- Most popular public key algorithm

El Gamal

- Developed by Taher ElGamal
- Variable key size (512 or 1024 bits)
- Less common than RSA, used in protocols like PGP

Asymmetric Encryption – RSA Algorithm

- Choose two large prime numbers p & q
- Compute n=pq and z=(p-1)(q-1)
- Choose number e, less than n, which has no common factor (other than 1) with z
- Find number d, such that ed – 1 is exactly divisible by z Keys are generated using n, d, e
  - Public key is (n,e)
  - Private key is (n, d)
- Encryption: c = me mod n
  - m is plain text
  - c is cipher text
- Decryption: m = cd mod n
- Public key is shared and the private key is hidden

Asymmetric Encryption - RSA

- P=5 & q=7
- n=5*7=35 and z=(4)*(6) = 24
- e = 5
- d = 29 , (29x5 –1) is exactly divisible by 24

(d=(K*z+1)/e (for some integer k lets suppose k=6)

- Keys generated are
  - Public key: (35,5)
  - Private key is (35, 29)
- Encrypt the word love using (c = me mod n)
  - Assume that the alphabets are between 1 & 26

Plain Text	Numeric Representation	m^e	Cipher Text ($c = m^e$ mod n)
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

Asymmetric Encryption - RSA

- Decrypt the word love using (m = cd mod n)
  - n = 35, c=29

Cipher Text	c^d	$(m = m^e \bmod n)$	Plain Text
17	481968572106750915091411825223072000	17	l
15	127834039488589391112327575683594 00	15	o
22	852643319086537701956194499721110000000	22	v
10	100000000000000000000000000000	10	e

Asymmetric Encryption – Key Agreement

- Key agreement is a method to create secret key by exchanging only public keys.
- Example
  - Bob sends Alice his public key
  - Alice sends Bob her public key
  - Bob uses Alice's public key and his private key to generate a session key
  - Alice uses Bob's public key and her private key to generate a session key
  - Using a key agreement algorithm both will generate same key
  - Bob and Alice do not need to transfer any key

Key Generation

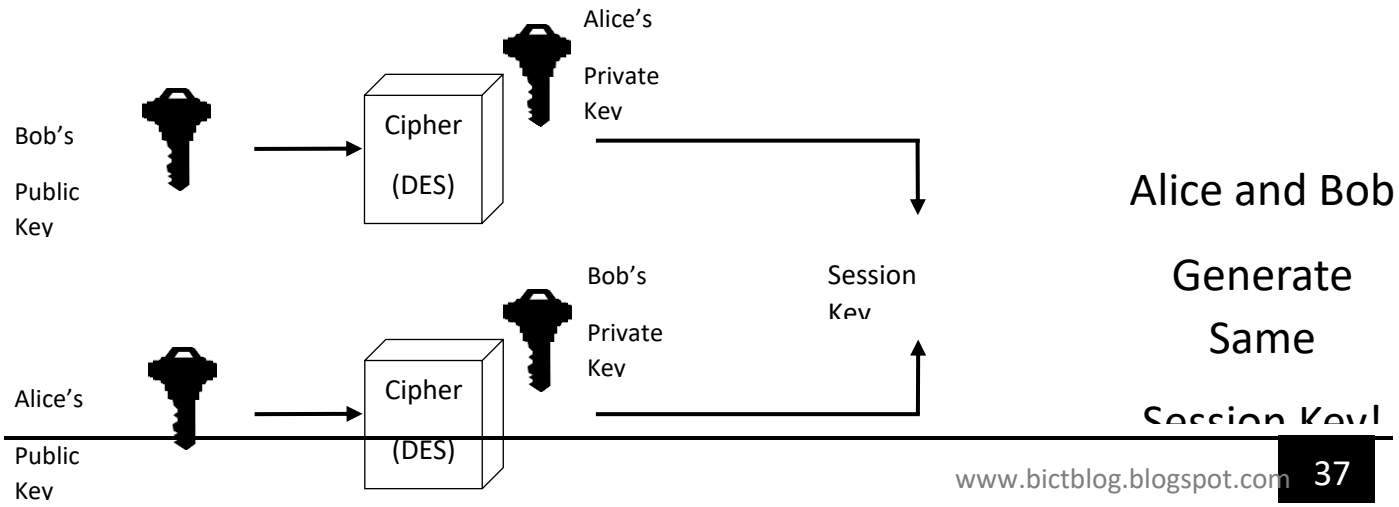
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$



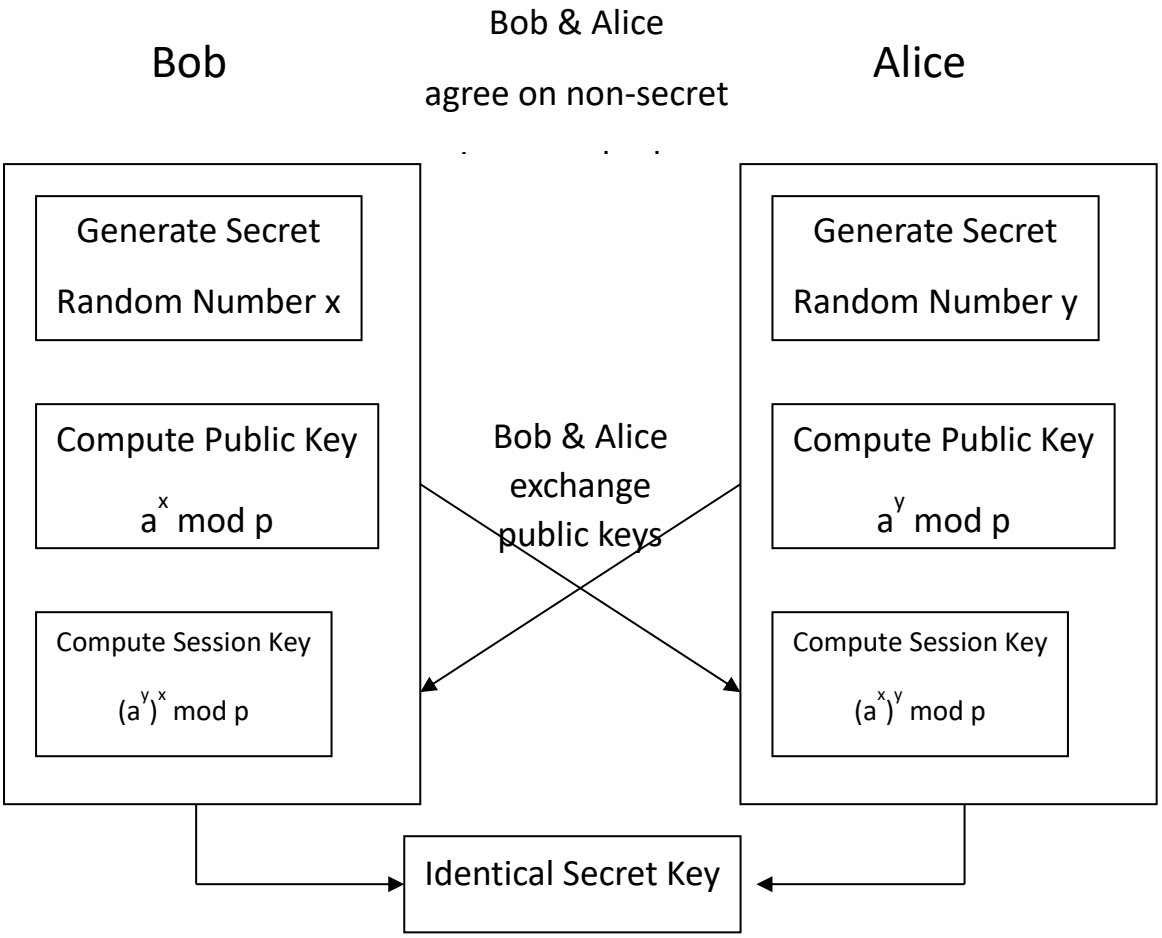
Session Key

Alice and Bob Generate Same Session Key!

Asymmetric Encryption – Key Agreement contd.

- Diffie-Hellman is the first key agreement algorithm
- Invented by Whitfield Diffie & Martin Hellman
- Provided ability for messages to be exchanged securely without having to have shared some secret information previously
- Inception of public key cryptography which allowed keys to be exchanged in the open
- No exchange of secret keys
- Man-in-the middle attack avoided

Diffie-Hellman Mathematical Analysis



Diffie-Hellman Key Exchange

- The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values.
- The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Global Public Elements

q	prime number
α	$\alpha < q$ *and* α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Calculation of Secret Key by User A

$K = (Y_B)^{X_A} \bmod q$

Calculation of Secret Key by User B

$K = (Y_A)^{X_B} \bmod q$

Primitive root of a prime number n modulo n

Given a prime number n, the task is to find its primitive root under modulo n. Primitive root of a prime number n is an integer r between[1, n-1] such that the values of r^x(mod n) where x is in range[0, n-2] are different. Return -1 if n is a non-prime number.

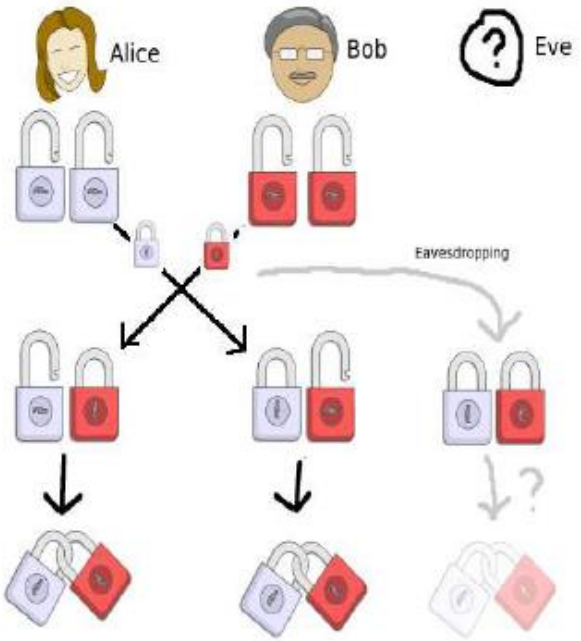
Examples:

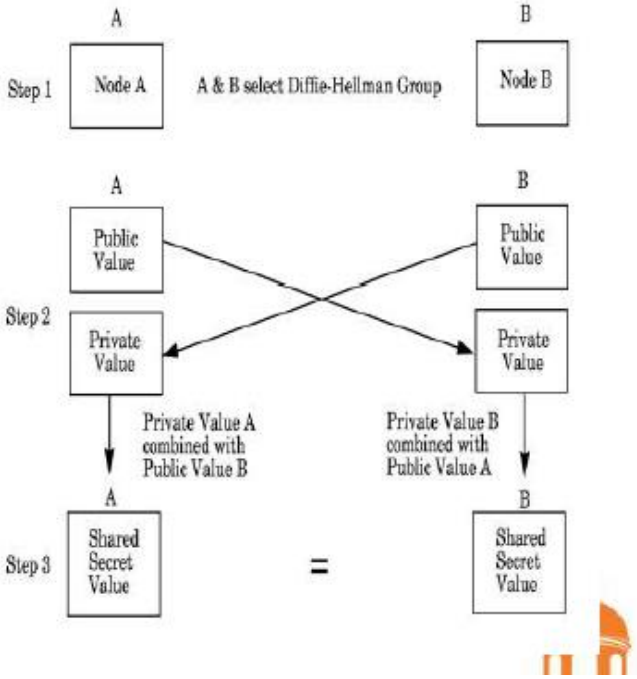
```
Input : 7
Output : Smallest primitive root = 3
Explanation: n = 7
3^0(mod 7) = 1
3^1(mod 7) = 3
3^2(mod 7) = 2
3^3(mod 7) = 6
3^4(mod 7) = 4
3^5(mod 7) = 5

Input : 761
Output : Smallest primitive root = 6
```

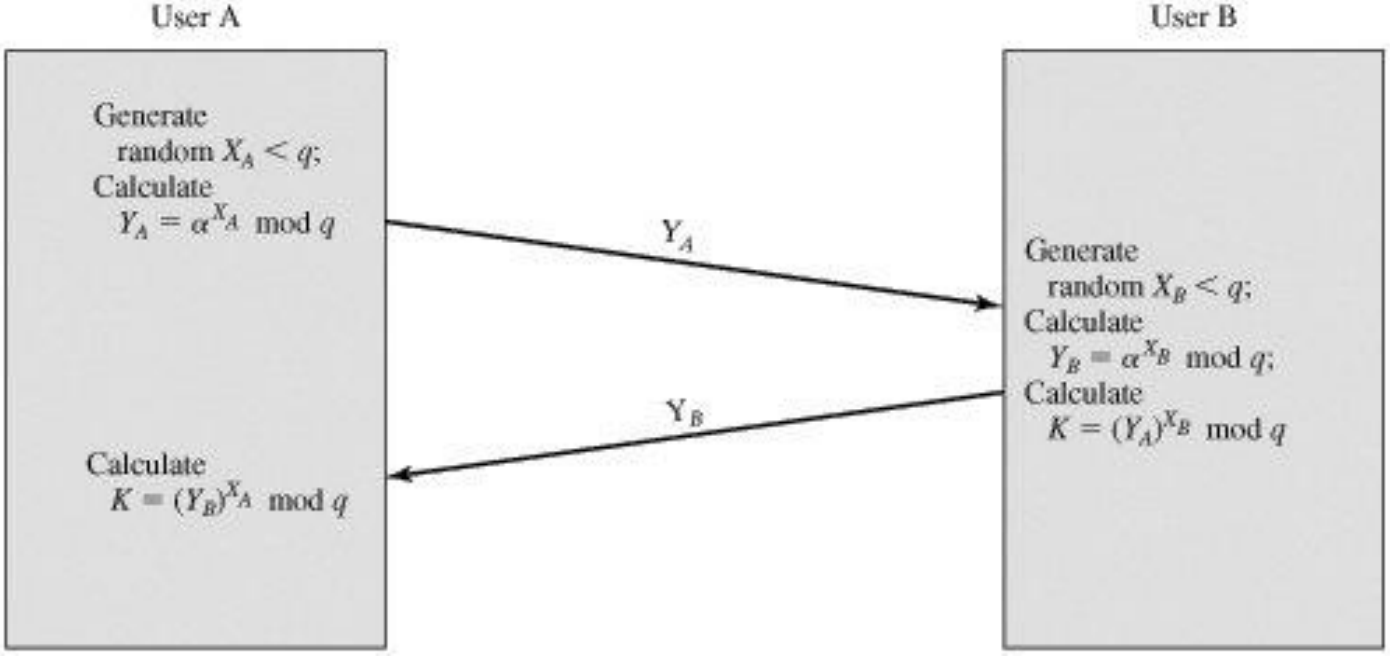
- users Alice & Bob who wish to swap keys:
- agree on prime q=**353** and α=**3**
- select random secret keys:
  - A chooses x_A=**97**, B chooses x_B=**233**
- compute respective public keys:
  - $y_A = \mathbf{3}^{97} \bmod \mathbf{353} = \mathbf{40}$ (Alice)
  - $y_B = \mathbf{3}^{233} \bmod \mathbf{353} = \mathbf{248}$ (Bob)
- compute shared session key as:
  - $K_{AB} = y_B^{xA} \bmod \mathbf{353} = \mathbf{248}^{97} = 160$ (Alice)
  - $K_{AB} = y_A^{xB} \bmod \mathbf{353} = \mathbf{40}^{233} = 160$ (Bob)

Diffie –Hellman Key Exchange





Diffie-Hellman Key Exchange

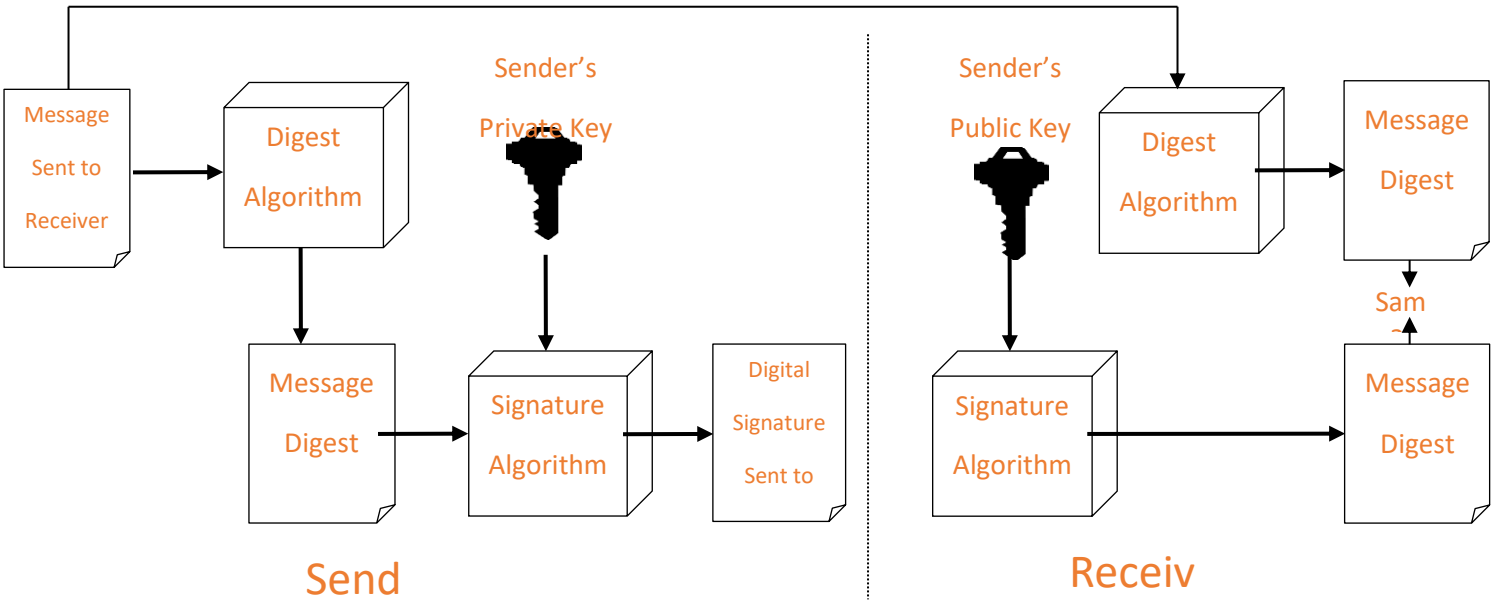


Message Signature

An authentication mechanism that enables the creator of a message to attach a code that acts as a signature. The signature is formed by taking the hash of the message and encrypting the message with the creator's private key. The signature guarantees the source and integrity of the message.

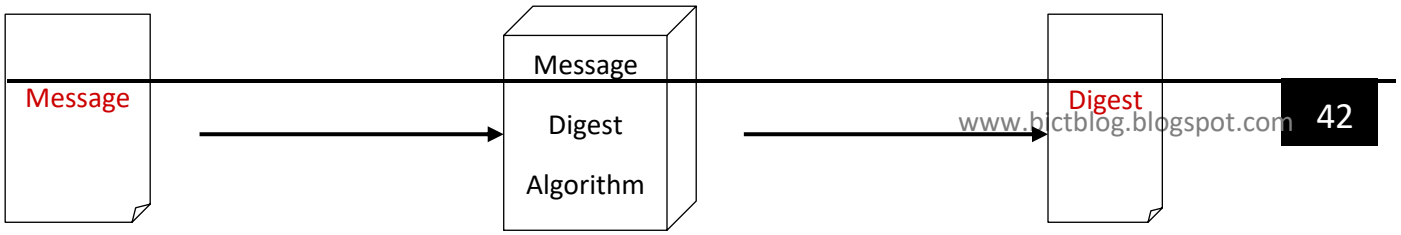
Authentication – Digital Signatures

- A digital signature is a data item which accompanies or is logically associated with a digitally encoded message.
- It has two goals
  - A guarantee of the source of the data
  - Proof that the data has not been tampered with



Authentication – Message Digests

- A message digest is a fingerprint for a document
- Purpose of the message digest is to provide proof that a document has not been tampered with.
- Hash functions used to generate message digests are one way functions that have following properties
  - It must be computationally infeasible to reverse the function
  - It must be computationally infeasible to construct two messages which which hash to the same digest
- Some of the commonly used hash algorithms are
  - MD5 – 128 bit hashing algorithm by Ron Rivest of RSA
  - SHA & SHA-1 – 162 bit hashing algorithm developed by NIST

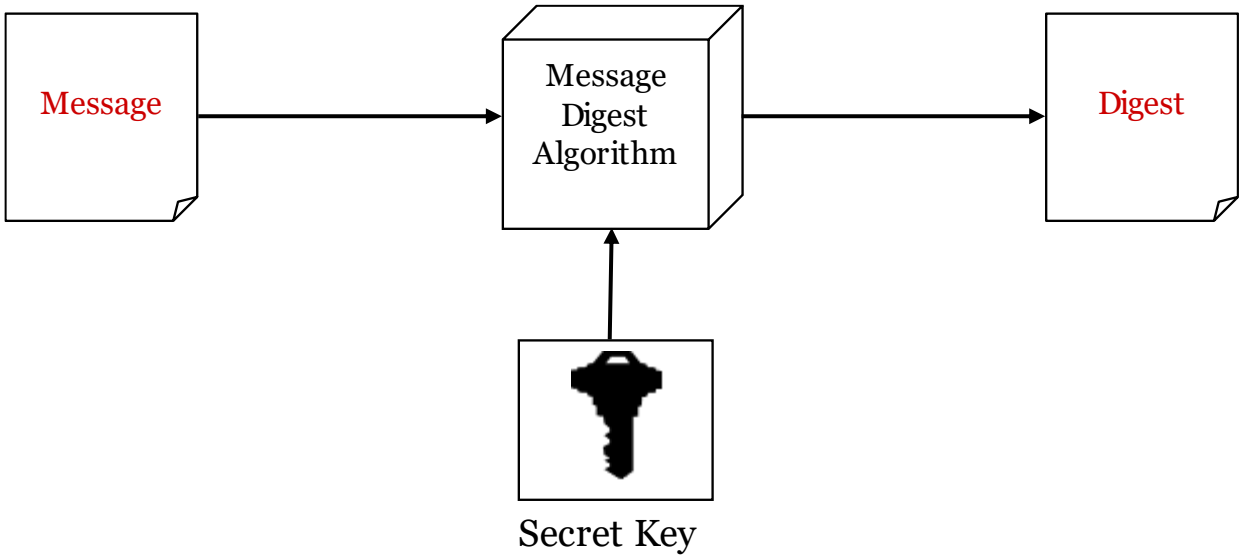


Message

Message Digest Algorithm

Message Authentication Codes

- A message digest created with a key
- Creates security by requiring a secret key to be possesses by both parties in order to retrieve the message
- Some of the commonly used hash algorithms are
  - MD5 – 128 bit hashing algorithm by Ron Rivest of RSA
  - SHA & SHA-1 – 162 bit hashing algorithm developed by NIST

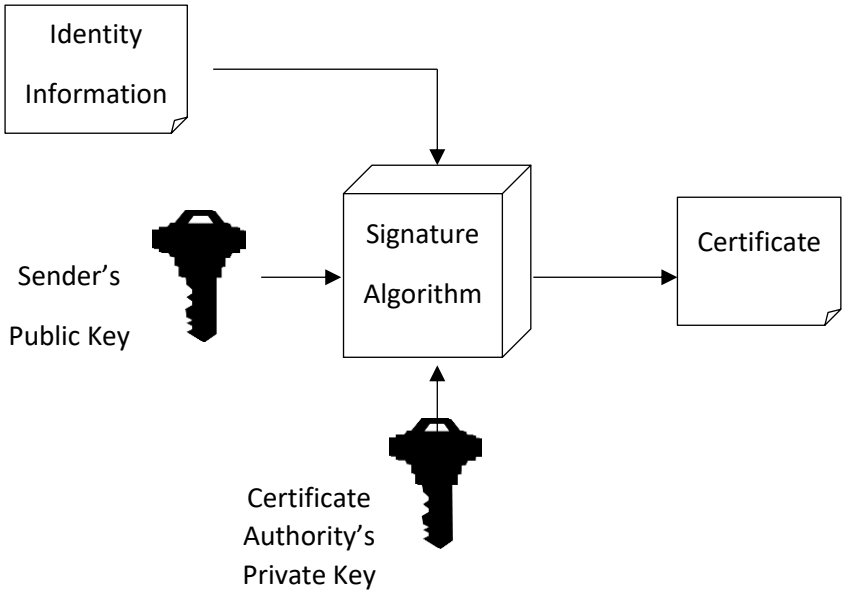


Message
Message Digest Algorithm

Secret Key
Authentication – Digital Certificates

- A digital certificate is a signed statement by a trusted party that another party's public key belongs to them.

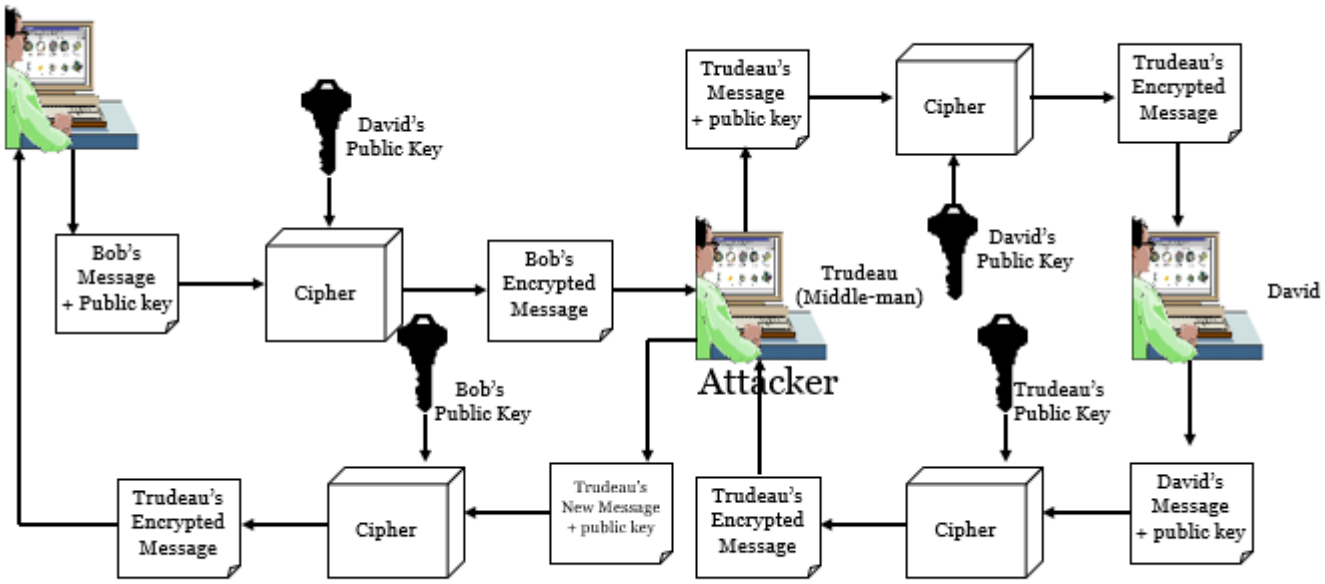
- This allows one certificate authority to be authorized by a different authority (root CA)
- Top level certificate must be self signed
- Any one can start a certificate authority
- Name recognition is key to some one recognizing a certificate authority
- Verisign is industry standard certificate authority



Identity Information
Sender's Public Key
Certificate Authority's Private Key

Asymmetric Encryption -Weaknesses

- Slow compared to symmetric Encryption
- It is problematic to get the key pair generated for the encryption.
- Vulnerable to man-in-the-middle attack
- Hacker could generate a key pair, give the public key away and tell everybody, that it belongs to somebody else. Now, everyone believing it will use this key for encryption, resulting in the hacker being able to read the messages. If he encrypts the messages again with the public key of the real recipient, he will not be recognized easily.



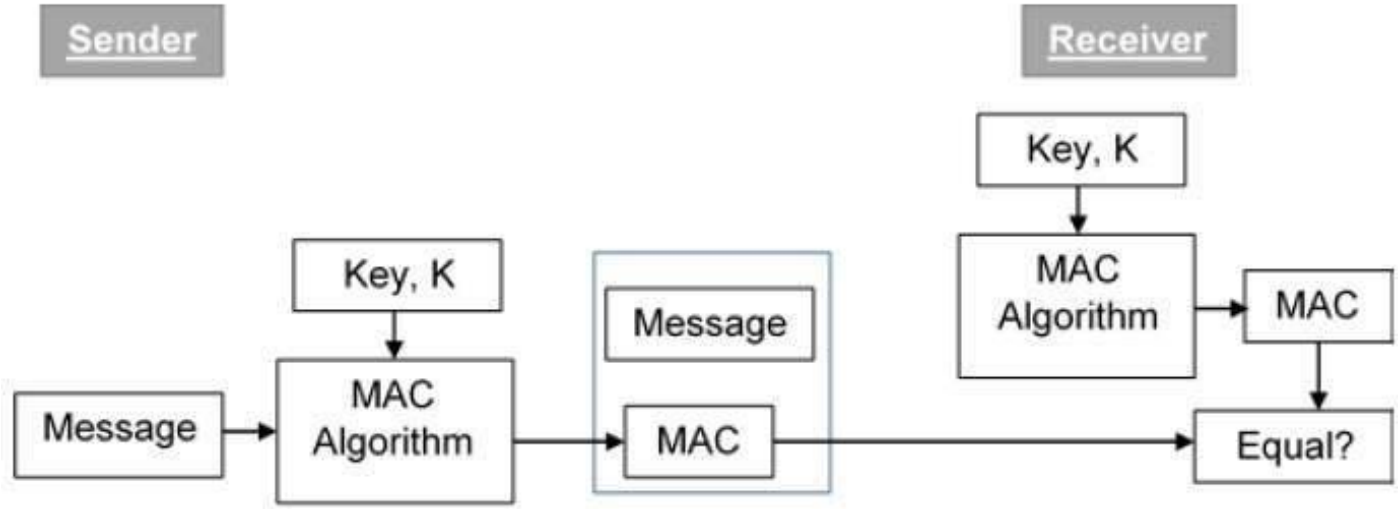
Trudeau's Encrypted Message

Asymmetric Encryption - Weaknesses

- Efficiency is lower than Symmetric Algorithms
- A 1024-bit asymmetric key is equivalent to 128-bit symmetric key
- Potential for eavesdropping attack during transmission of key
- It is problematic to get the key pair generated for the encryption

Message Authentication Code (MAC)

- 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.



- The sender uses some publicly known MAC algorithm, inputs the message and the secret key K and produces a MAC value.

- Similar to hash, MAC function also compresses an arbitrary long input into a fixed length output. The major difference between hash and MAC is that MAC uses secret key during the compression.
- The sender forwards the message along with the MAC. Here, we assume that the message is sent in the clear, as we are concerned of providing message origin authentication, not confidentiality. If confidentiality is required then the message needs encryption.
- On receipt of the message and the MAC, the receiver feeds the received message and the shared secret key K into the MAC algorithm and re-computes the MAC value.
- The receiver now checks equality of freshly computed MAC with the MAC received from the sender. If they match, then the receiver accepts the message and assures himself that the message has been sent by the intended sender.
- If the computed MAC does not match the MAC sent by the sender, the receiver cannot determine whether it is the message that has been altered or it is the origin that has been falsified. As a bottom-line, a receiver safely assumes that the message is not the genuine.

Limitations of MAC

- Establishment of Shared Secret.
- It can provide message authentication among pre- decided legitimate users who have shared key.
- This requires establishment of shared secret prior to use of MAC.

● Inability to Provide Non-Repudiation

● Non-repudiation is the assurance that a message originator cannot deny any previously sent messages and commitments or actions.

● MAC technique does not provide a non-repudiation service. If the sender and receiver get involved in a dispute over message origination, MACs cannot provide a proof that a message was indeed sent by the sender.

● Though no third party can compute the MAC, still sender could deny having sent the message and claim that the receiver forged it, as it is impossible to determine which of the two parties computed the MAC.

UNIT 4

Network Security Applications

Network security is any activity designed to protect the usability and integrity of your network and data.
It includes both hardware and software technologies. Effective network security manages access to the network.
It targets a variety of threats and stops them from entering or spreading on your network.
Various business services are now offered online though client-server applications. The most popular forms are web application and e-mail. In both applications, the client communicates to the designated server and obtains services.
While using a service from any server application, the client and server exchange a lot of information on the underlying intranet or Internet. We are aware of fact that these information transactions are vulnerable to various attacks.
Network security entails securing data against attacks while it is in transit on a network. To achieve this goal, many real-time security protocols have been designed

Public-Key Infrastructure

- The most distinct feature of Public Key Infrastructure (PKI) is that it uses a pair of keys to achieve the underlying security service. The key pair comprises of private key and public key.
- Since the public keys are in open domain, they are likely to be abused. It is, thus, necessary to establish and maintain some kind of trusted infrastructure to manage these keys.

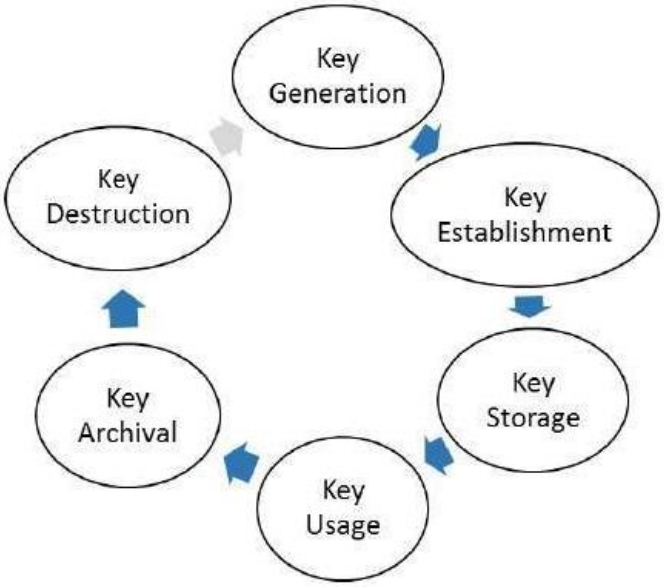
Key Management

- It goes without saying that the security of any cryptosystem depends upon how securely its keys are managed. Without secure procedures for the handling of cryptographic keys, the benefits of the use of strong cryptographic schemes are potentially lost.
- It is observed that cryptographic schemes are rarely compromised through weaknesses in their design. However, they are often compromised through poor key management.

There are some important aspects of key
management which are as follows −

- Cryptographic keys are nothing but special pieces of data. Key management refers to the secure administration of cryptographic keys.
- Key management deals with entire key lifecycle as

depicted in the following illustration –



- There are two specific requirements of key management for public key cryptography.
- Secrecy of private keys. Throughout the key lifecycle, secret keys must remain secret from all parties except those who are owner and are authorized to use them.
- Assurance of public keys. In public key cryptography, the public keys are in open domain and seen as public pieces of data. By default there are no assurances of whether a public key is correct, with whom it can be associated, or what it can be used for. Thus key management of public keys needs to focus much more explicitly on assurance of purpose of public keys.

Public Key Infrastructure (PKI)

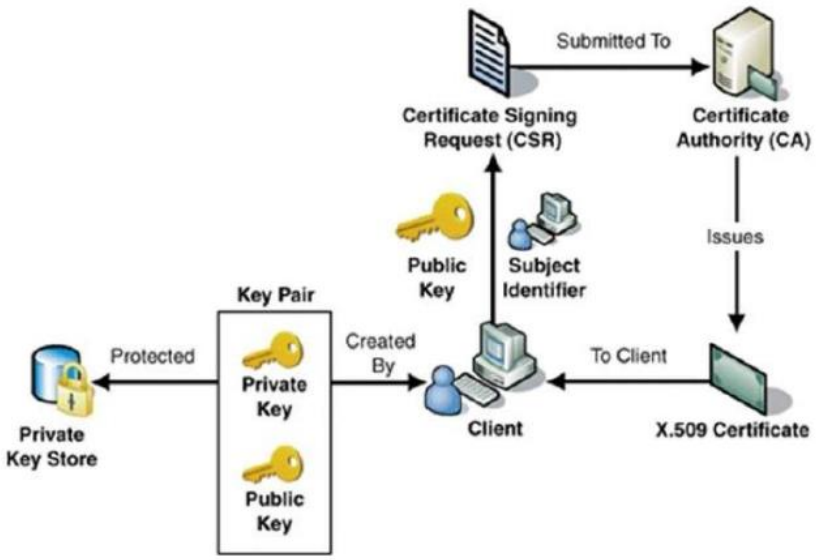
- PKI provides assurance of public key. It provides the identification of public keys and their distribution. An anatomy of PKI comprises of the following components.
- Public Key Certificate, commonly referred to as 'digital certificate'.
- Private Key tokens.
- Certification Authority.
- Registration Authority.
- Certificate Management System.

Digital Certificate

- For analogy, a certificate can be considered as the ID card issued to the person. People use ID cards such as a driver's license, passport to prove their identity. A digital certificate does the same basic thing in the electronic world, but with one difference.
- Digital Certificates are not only issued to people but they can be issued to computers, software packages or anything else that need to prove the identity in the electronic world.

- Digital certificates are based on the ITU standard X.509 which defines a standard certificate format for public key certificates and certification validation. Hence digital certificates are sometimes also referred to as X.509 certificates.
- Public key pertaining to the user client is stored in digital certificates by The Certification Authority (CA) along with other relevant information such as client information, expiration date, usage, issuer etc.
- CA digitally signs this entire information and includes digital signature in the certificate.
- Anyone who needs the assurance about the public key and associated information of client, he carries out the signature validation process using CA's public key. Successful validation assures

that the public key given in the certificate belongs to the person whose details are given in the certificate.



As shown in the illustration, the CA accepts the application from a client to certify his public key. The CA, after duly verifying identity of client, issues a digital certificate to that client.

Certifying Authority (CA)

- As discussed above, the CA issues certificate to a client and assist other users to verify the certificate.
- The CA takes responsibility for identifying correctly the identity of the client asking for a certificate to be issued, and ensures that the information contained within the certificate is correct and digitally signs it.

Key Functions of CA

- Generating key pairs − The CA may generate a key pair independently or jointly with the client.
- Issuing digital certificates − The CA could be thought of as the PKI equivalent of a passport agency − the CA issues a certificate after client provides the credentials to confirm his identity. The CA then signs the certificate to prevent modification of the details contained in the certificate.
- Publishing Certificates − The CA need to publish certificates so that users can find them. There are two ways of achieving this. One is to publish certificates in the equivalent of an electronic telephone directory. The other is to send your certificate out to those people you think might need it by one means or another.
- Verifying Certificates − The CA makes its public key available in environment to assist verification of his signature on clients' digital certificate.
- Revocation of Certificates − At times, CA revokes the certificate issued due to some reason such as compromise of private key by user or loss of trust in the client. After revocation, CA maintains the list of all revoked certificate that is available to the environment.

Classes of Certificates

- There are four typical classes of certificate −
- Class 1 − These certificates can be easily acquired by

supplying an email address.

- Class 2 − These certificates require additional personal information to be supplied.
- Class 3 − These certificates can only be purchased after checks have been made about the requestor's identity.
- Class 4 − They may be used by governments and financial organizations needing very high levels of trust.

Registration Authority (RA)

- CA may use a third-party Registration Authority (RA) to perform the necessary checks on the person or company requesting the certificate to confirm their identity. The RA may appear to the client as a CA, but they do not actually sign the certificate that is issued.

Certificate Management System (CMS)

- It is the management system through which certificates are published, temporarily or permanently suspended, renewed, or revoked. Certificate management systems do not normally delete certificates because it may be necessary to prove their status at a point in time, perhaps for legal reasons. A CA along with associated RA runs certificate management systems to be able to track their responsibilities and liabilities.

Private Key Tokens

- While the public key of a client is stored on the certificate, the associated secret private key can be stored on the key owner's computer. This method is generally not adopted. If an attacker gains access to the computer, he can easily gain access to private key. For this reason, a private key is stored on secure removable storage token access to which is protected through a password.
- Different vendors often use different and sometimes proprietary storage formats for storing keys. For example, Entrust uses the proprietary .epf format, while Verisign, GlobalSign, and Baltimore use the standard .p12 format.

- Network security entails securing data against attacks while it is in transit on a network. To achieve this goal, many real-time security protocols have been designed. There are popular standards for real-time network security protocols such as S/MIME, SSL/TLS, SSH, and IPsec. As mentioned earlier, these protocols work at different layers of networking model.
- In the last chapter, we discussed some popular protocols that are designed to provide application layer security. In this chapter, we will discuss the process of achieving network security at Transport Layer and associated security protocols.
- For TCP/IP protocol based network, physical and data link layers are typically implemented in the user terminal and network card hardware. TCP and IP layers are implemented in the operating system. Anything above TCP/IP is implemented as user process.

Need for Transport Layer Security

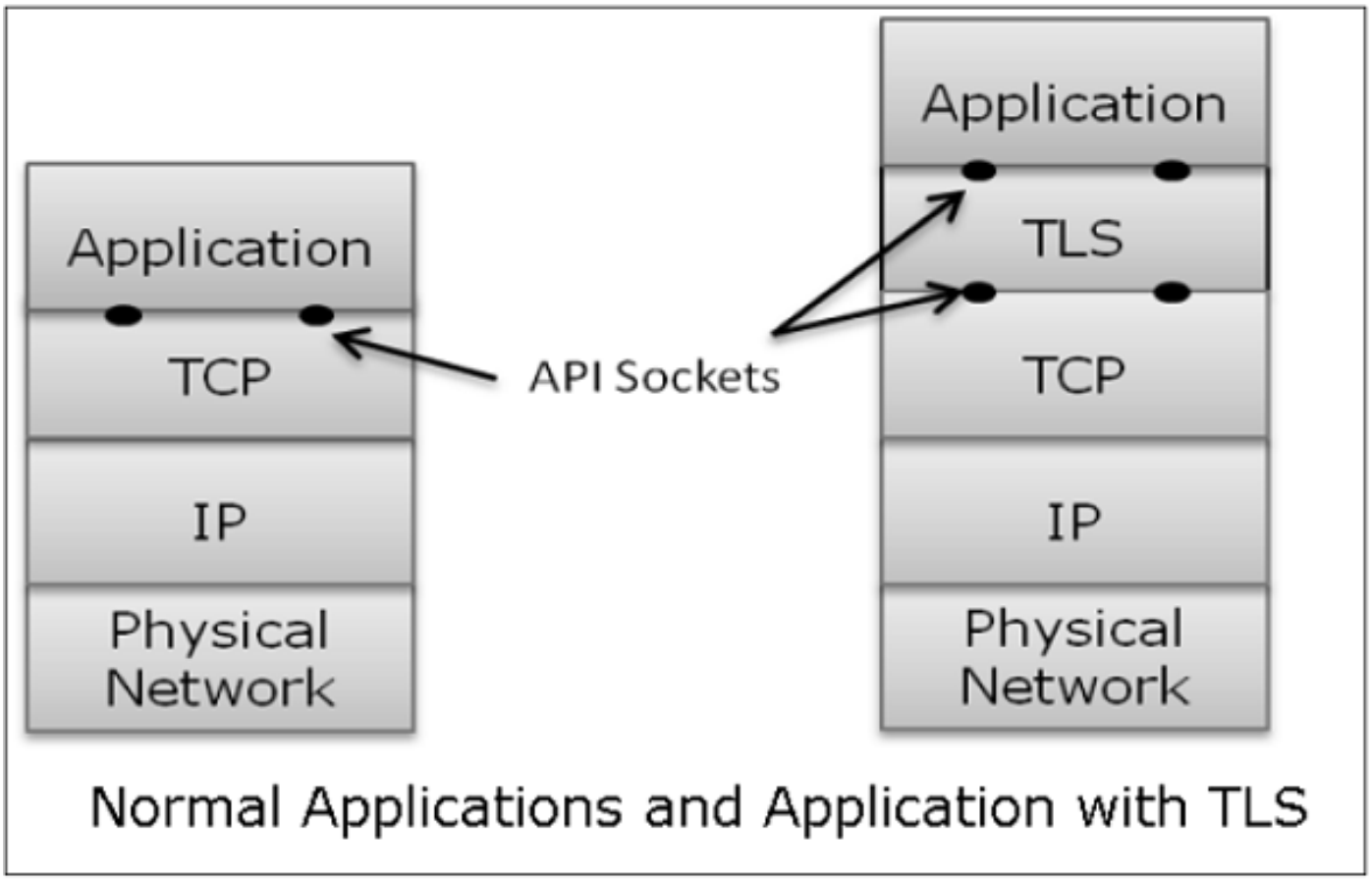
- If transactions did not use confidentiality (encryption), an attacker could obtain his payment card information. The attacker can then make purchases at Bob's expense.
- If no data integrity measure is used, an attacker could modify Bob's order in terms of type or quantity of goods.
- Lastly, if no server authentication is used, a server could display Alice's famous logo but the site could be a malicious site maintained by an attacker, who is masquerading as Alice. After

receiving Bob's order, he could take Bob's money and flee. Or he could carry out an identity theft by collecting Bob's name and credit card details.

- Transport layer security schemes can address these problems by enhancing TCP/IP based network communication with confidentiality, data integrity, server authentication, and client authentication.
- The security at this layer is mostly used to secure HTTP based web transactions on a network. However, it can be employed by any application running over TCP.

Philosophy of TLS Design

- Transport Layer Security (TLS) protocols operate above the TCP layer. Design of these protocols use popular Application Program Interfaces (API) to TCP, called “sockets" for interfacing with TCP layer.
- Applications are now interfaced to Transport Security Layer instead of TCP directly. Transport Security Layer provides a simple API with sockets, which is similar and analogous to TCP's API.



- In the above diagram, although TLS technically resides between application and transport layer, from the common perspective it is a transport protocol that acts as TCP layer enhanced with security services.
- TLS is designed to operate over TCP, the reliable layer 4 protocol (not on UDP protocol), to make design of TLS much simpler, because it doesn't have to worry about ‘timing out’ and ‘retransmitting lost data’. The TCP layer continues doing that as usual which serves the need of TLS.

Why TLS is Popular?

- The reason for popularity of using a security at Transport Layer is simplicity.

- Design and deployment of security at this layer does not require any change in TCP/IP protocols that are implemented in an operating system.
- Only user processes and applications needs to be designed/modified which is less complex.

Secure Socket Layer (SSL)

- We discuss the family of protocols designed for TLS. The family includes SSL versions 2 and 3 and TLS protocol. SSLv2 has been now replaced by SSLv3, so we will focus on SSL v3 and TLS.

Brief History of SSL

- In year 1995, Netscape developed SSLv2 and used in Netscape Navigator 1.1. The SSL version1 was never published and used. Later, Microsoft improved upon SSLv2 and introduced another similar protocol named Private Communications Technology (PCT).
- Netscape substantially improved SSLv2 on various security issues and deployed SSLv3 in 1999. The Internet Engineering Task Force (IETF) subsequently, introduced a similar TLS (Transport Layer Security) protocol as an open standard. TLS protocol is non- interoperable with SSLv3.
- TLS modified the cryptographic algorithms for key expansion and authentication. Also, TLS suggested use of open crypto Diffie- Hellman (DH) and Digital Signature Standard (DSS) in place of patented RSA crypto used in SSL. But due to expiry of RSA patent in 2000, there existed no strong reasons for users to shift away from the widely deployed SSLv3 to TLS.

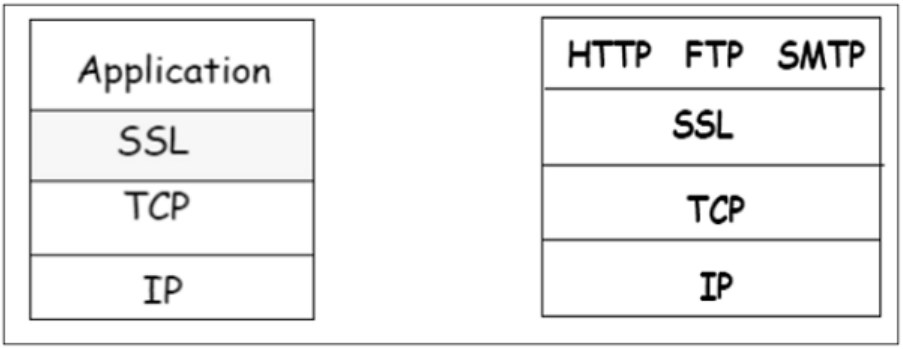
Salient Features of SSL

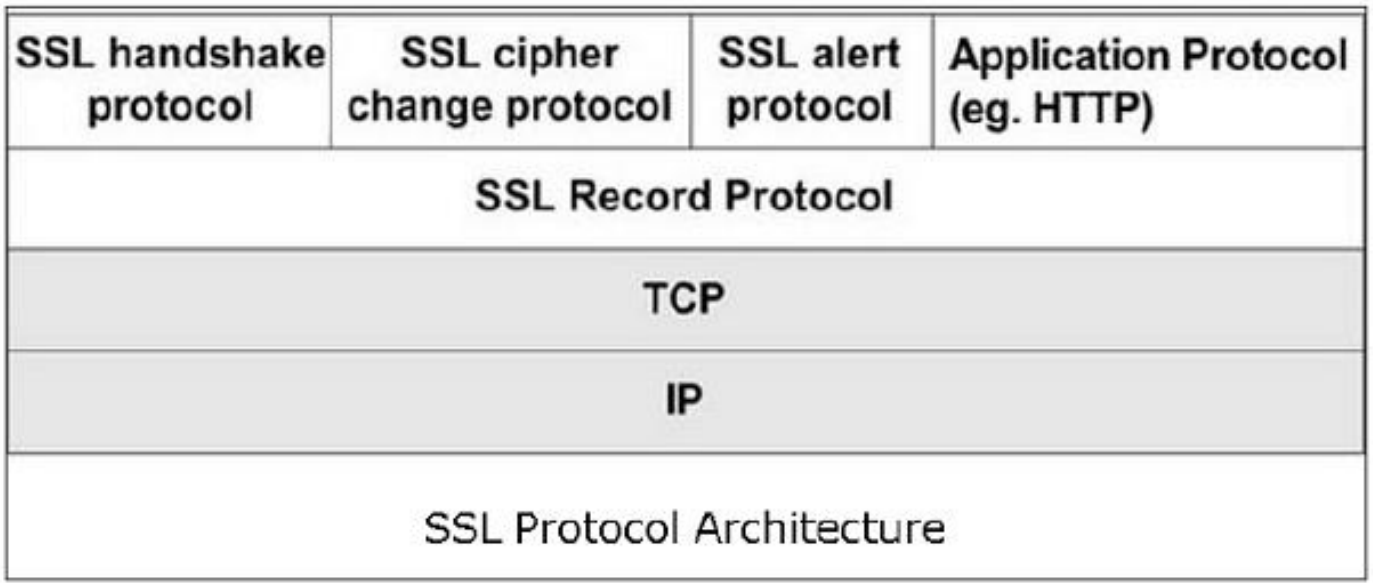
- SSL provides network connection security through −
  - Confidentiality − Information is exchanged in an encrypted form.
  - Authentication − Communication entities identify each other through the use of digital certificates. Web-server authentication is mandatory whereas client authentication is kept optional.
  - Reliability − Maintains message integrity checks.
- SSL is available for all TCP applications.
- Supported by almost all web browsers.
- Provides ease in doing business with new online entities.
- Developed primarily for Web e-commerce.

Architecture of SSL

- SSL is specific to TCP and it does not work with UDP. SSL provides Application Programming Interface (API) to applications. C and Java SSL libraries/classes are readily available.
- SSL protocol is designed to interwork between application and transport layer as shown in the following image −
- SSL itself is not a single layer protocol as depicted in the image; in fact it is composed of two sub-layers.
- Lower sub-layer comprises of the one component of SSL protocol called as SSL Record Protocol. This component provides integrity and confidentiality services.
- Upper sub-layer comprises of three SSL-related protocol components and an application protocol. Application component provides the information transfer service between client/server interactions. Technically, it can operate on top of SSL layer as well. Three SSL related protocol components are
  - SSL Handshake Protocol

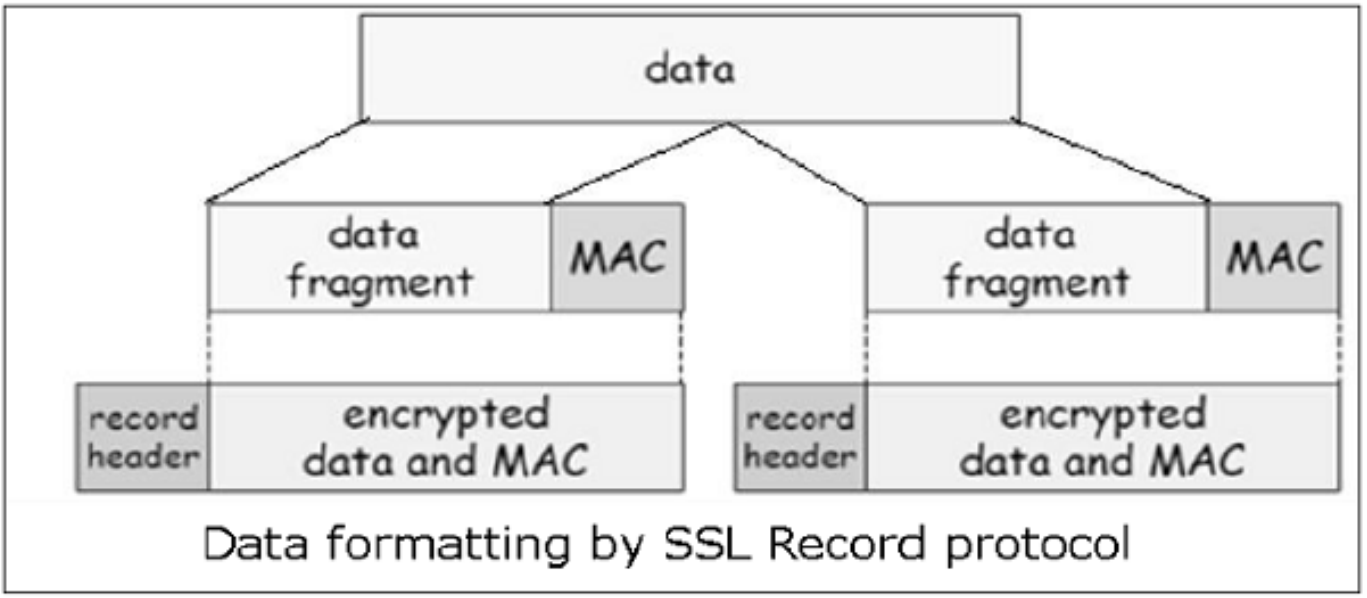
- Change Cipher Spec Protocol
- Alert Protocol.





Functions of SSL Protocol Components

- The four sub-components of the SSL protocol handle various tasks for secure communication between the client machine and the server.
- **Record Protocol**
  - The record layer formats the upper layer protocol messages.
  - It fragments the data into manageable blocks (max length 16 KB). It optionally compresses the data.
  - Encrypts the data.
  - Provides a header for each message and a hash (Message Authentication Code (MAC)) at the end.
  - Hands over the formatted blocks to TCP layer for transmission.



- **SSL Handshake Protocol**
- It is the most complex part of SSL. It is invoked before any application data is transmitted. It creates SSL sessions between the client and the server.
- Establishment of session involves Server authentication, Key and algorithm negotiation, Establishing keys and Client authentication (optional).
- A session is identified by unique set of cryptographic security parameters.
- Multiple secure TCP connections between a client and a server can share the same session.
- Handshake protocol actions through four phases.

- **Change CipherSpec Protocol**
- Simplest part of SSL protocol. It comprises of a single message exchanged between two communicating entities, the client and the server.
- As each entity sends the Change CipherSpec message, it changes its side of the connection into the secure state as agreed upon.
- The cipher parameters pending state is copied into the current state.
- Exchange of this Message indicates all future data exchanges are encrypted and integrity is protected.

- **SSL Alert Protocol**
- This protocol is used to report errors – such as unexpected message, bad record MAC, security parameters negotiation failed, etc.
- It is also used for other purposes – such as notify closure of the TCP connection, notify receipt of bad or unknown certificate, etc.

- All four phases, discussed above, happen within the establishment of TCP session. SSL session establishment starts after TCP SYN/ SYNACK and finishes before TCP

TLS Protocol

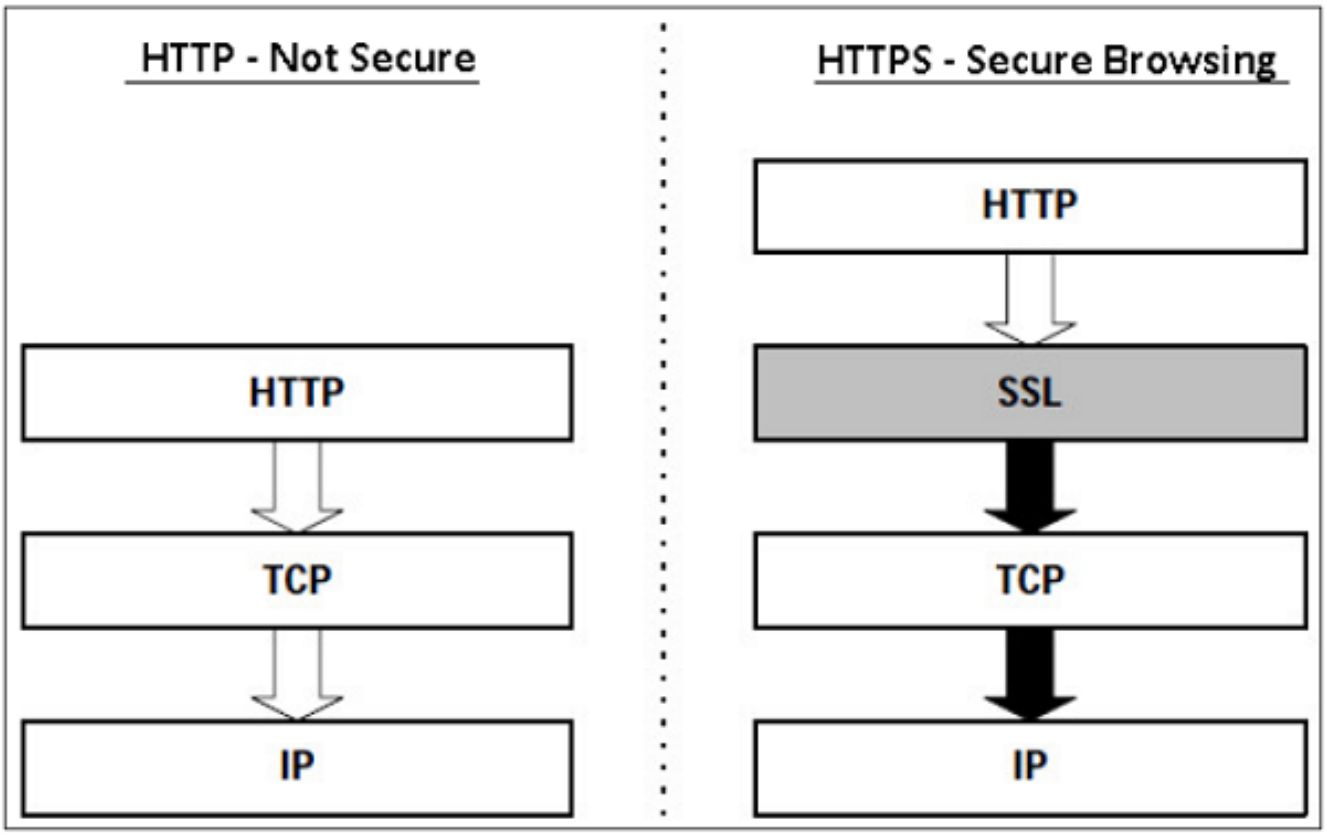
- In order to provide an open Internet standard of SSL, IETF released The Transport Layer Security (TLS) protocol in January 1999. TLS is defined as a proposed Internet Standard in RFC 5246.

Salient Features of TLS Protocol

- TLS protocol has same objectives as SSL.
- It enables client/server applications to communicate in a secure manner by authenticating, preventing eavesdropping and resisting message modification.
- TLS protocol sits above the reliable connection-oriented transport TCP layer in the networking layers stack.
- The architecture of TLS protocol is similar to SSLv3 protocol. It has two sub protocols: the TLS Record protocol and the TLS Handshake protocol.
- Though SSLv3 and TLS protocol have similar architecture, several changes were made in architecture and functioning particularly for the handshake protocol.

HTTPS Defined

- Hyper Text Transfer Protocol (HTTP) protocol is used for web browsing. The function of HTTPS is similar to HTTP.
- The only difference is that HTTPS provides “secure” web browsing.
- HTTPS stands for HTTP over SSL.
- This protocol is used to provide the encrypted and authenticated connection between the client web browser and the website server.



- The secure browsing through HTTPS ensures that the following content are encrypted −
- URL of the requested web page.
- Web page contents provided by the server to the user client.
- Contents of forms filled in by user.
- Cookies established in both directions.

Working of HTTPS

- You request a HTTPS connection to a webpage by entering https:// followed by URL in the browser address bar.
- Web browser initiates a connection to the web server. Use of https invokes the use of SSL protocol.
- An application, browser in this case, uses the system port 443 instead of port 80 (used in case of http).
- The SSL protocol goes through a handshake protocol for establishing a secure session as discussed in earlier sections.
- The website initially sends its SSL Digital certificate to your browser. On verification of certificate, the SSL handshake progresses to exchange the shared secrets for the session.
- When a trusted SSL Digital Certificate is used by the server, users get to see a padlock icon in the browser address bar. When an Extended Validation Certificate is installed on a website, the address bar turns green.
- Once established, this session consists of many secure connections between the web server and the browser.

Use of HTTPS

- Use of HTTPS provides confidentiality, server authentication and message integrity to the user. It enables safe conduct of e-commerce on the Internet.
- Prevents data from eavesdropping and denies identity theft which are common attacks on HTTP.
- Present day web browsers and web servers are equipped with HTTPS support. The use of HTTPS over HTTP, however, requires more computing power at the client and the server end to carry out encryption and SSL handshake.

Secure Shell Protocol (SSH)

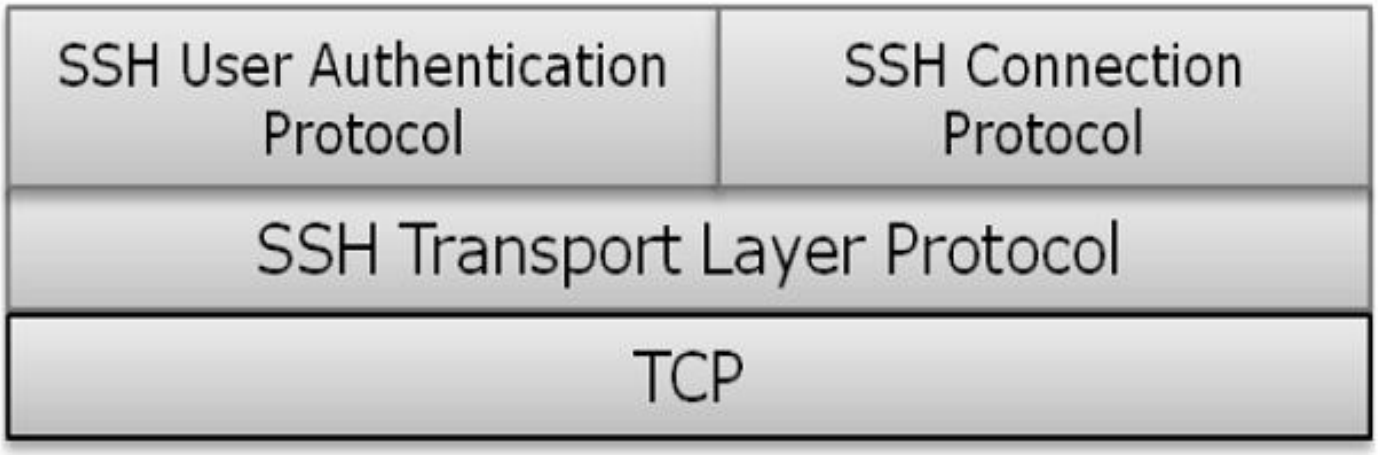
The salient features of SSH are as follows −

- SSH is a network protocol that runs on top of the TCP/IP layer. It is designed to replace the TELNET which provided unsecure means of remote logon facility.
- SSH provides a secure client/server communication and can be used for tasks such as file transfer and e- mail.
- SSH2 is a prevalent protocol which provides improved network communication security over earlier version SSH1.

SSH is organized as three sub-protocols

SSH is organized as three sub-protocols.

- Transport Layer Protocol
- User Authentication Protocol
- Connection Protocol



- **Transport Layer Protocol** − This part of SSH protocol provides data confidentiality, server (host) authentication, and data integrity. It may optionally provide data compression as well.

Server Authentication − Host keys are asymmetric like public/private keys. A server uses a public key to prove its identity to a client. The client verifies that contacted server is a “known” host from the database it maintains. Once the server is authenticated, session keys are generated.

Session Key Establishment − After authentication, the server and the client agree upon cipher to be used. Session keys are generated by both the client and the server. Session keys are generated before user authentication so that usernames and passwords can be sent encrypted. These keys are generally replaced at regular intervals (say, every hour) during the session and are destroyed immediately after use.

Data Integrity − SSH uses Message Authentication Code (MAC) algorithms to for data integrity check. It is an improvement over 32 bit CRC used by SSH1.

User Authentication Protocol − This part of SSH authenticates the user to the server. The server verifies that access is given to intended users only. Many authentication methods are currently used such as, typed passwords, Kerberos, public-key authentication, etc.

Connection Protocol − This provides multiple logical channels over a single underlying SSH connection.

SSH Services

SSH provides three main services that enable provision of many secure solutions. These services are briefly described as follows −

- **Secure Command-Shell (Remote Logon)**
- **Secure File Transfer**
- **Port Forwarding (Tunneling)**

Secure Command-Shell (Remote Logon) − It allows the user to edit files, view the contents of directories, and access applications on connected device. Systems administrators can remotely start/view/stop services and processes, create user accounts, and change file/directories permissions and so on. All tasks that are feasible at a machine’s command prompt can now be performed securely from the remote machine using secure remote logon.

Secure File Transfer − SSH File Transfer Protocol (SFTP) is designed as an extension for SSH-2 for secure file transfer. In essence, it is a separate protocol layered over the Secure Shell protocol to handle file transfers. SFTP encrypts both the username/password and the file data being transferred. It uses the same port as the Secure Shell server, i.e. system port no 22.

Port Forwarding (Tunneling) − It allows data from unsecured TCP/IP based applications to be secured. After port forwarding has been set up, Secure Shell reroutes traffic from a program (usually a client) and sends it across the encrypted tunnel to the program on the other side (usually a server).

Multiple applications can transmit data over a single multiplexed secure channel, eliminating the need to open many ports on a firewall or router.

Benefits

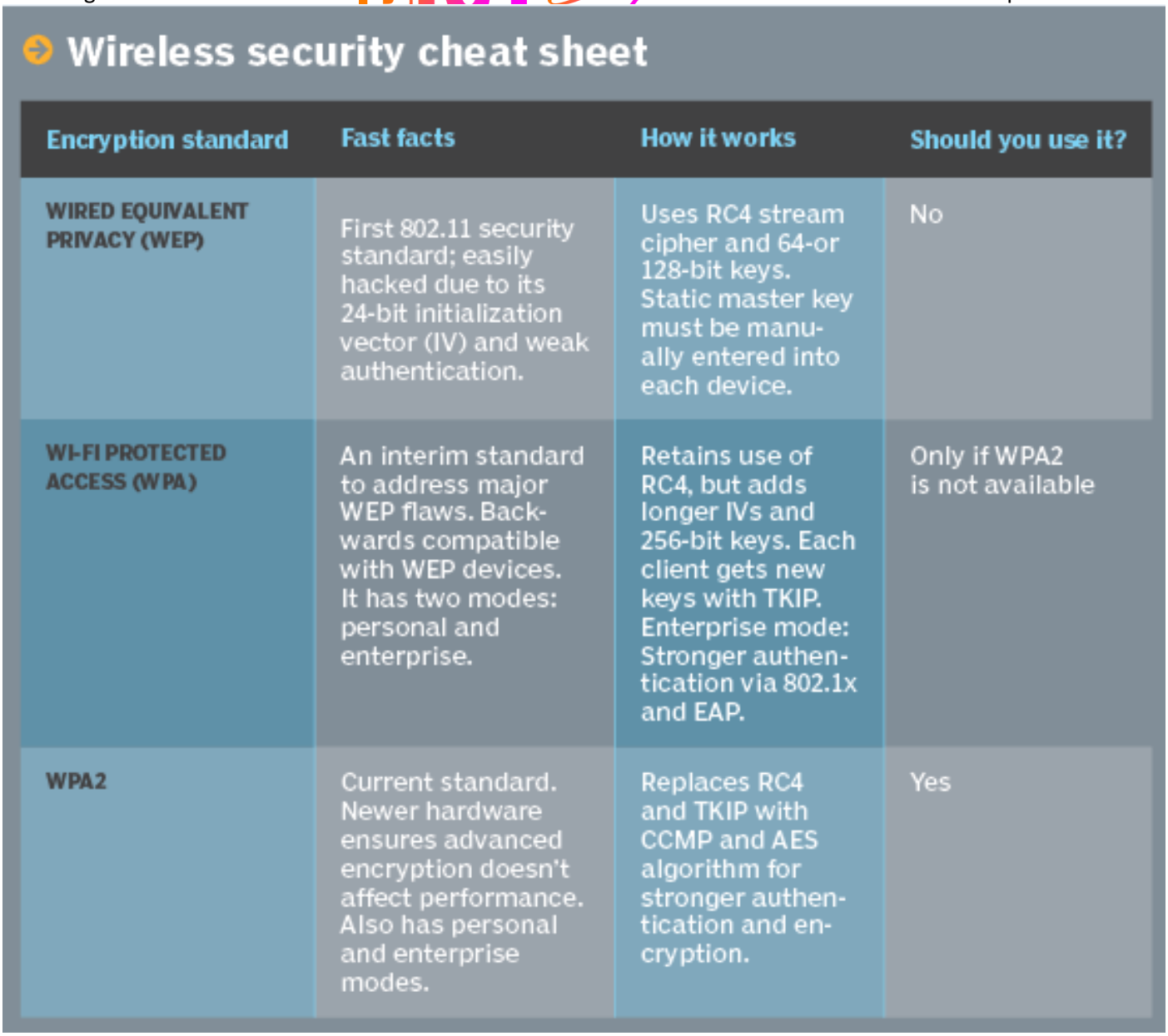
- Transport Layer Security is transparent to applications.
- Server is authenticated.
- Application layer headers are hidden.
- It is more fine-grained than security mechanisms at layer 3 (IPsec) as it works at the transport connection level.

Limitations

- Applicable to TCP-based applications only (not UDP).
- TCP/IP headers are in clear.
- Suitable for direct communication between the client and the server. Does not cater for secure applications using chain of servers (e.g. email)
- SSL does not provide non-repudiation as client authentication is optional.
- If needed, client authentication needs to be implemented above SSL.

Wireless security protocols

- In wireless security, passwords are only half the battle. Choosing the proper level of encryption is just as vital, and the right choice will determine whether your wireless LAN is a house of straw or a shielded fortress.
- Most wireless access points come with the ability to enable one of three wireless encryption standards: Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA) or WPA2. Explore the chart below to get a basic understanding of the differences between WPA, WEP and WPA2, as well as the uses and mechanisms of each one of these wireless security protocols, and to find out whether WPA, WEP or WPA2 is the best choice for your environment.



Wireless security cheat sheet

Encryption standard	Fast facts	How it works	Should you use it?
WIRED EQUIVALENT PRIVACY (WEP)	First 802.11 security standard; easily hacked due to its 24-bit initialization vector (IV) and weak authentication.	Uses RC4 stream cipher and 64-or 128-bit keys. Static master key must be manually entered into each device.	No
WI-FI PROTECTED ACCESS (WPA)	An interim standard to address major WEP flaws. Backwards compatible with WEP devices. It has two modes: personal and enterprise.	Retains use of RC4, but adds longer IVs and 256-bit keys. Each client gets new keys with TKIP. Enterprise mode: Stronger authentication via 802.1x and EAP.	Only if WPA2 is not available
WPA2	Current standard. Newer hardware ensures advanced encryption doesn't affect performance. Also has personal and enterprise modes.	Replaces RC4 and TKIP with CCMP and AES algorithm for stronger authentication and encryption.	Yes

Wired Equivalency Privacy (WEP)

- Developed in the late 1990s as the first encryption algorithm for the 802.11 standard, WEP was designed with one main goal in mind: to prevent hackers from snooping on wireless data as it was transmitted between clients and access points (APs). From the start, however, WEP lacked the strength necessary to accomplish this.
- Cybersecurity experts identified several severe flaws in WEP in 2001, eventually leading to industrywide recommendations to phase out the use of WEP in both enterprise and consumer devices. After a large-scale cyberattack executed against T.J. Maxx in 2009 was traced back to vulnerabilities exposed by WEP, the Payment Card Industry Data Security
Standard prohibited retailers and other entities that processed credit card data from using WEP.

WEP uses the RC4 stream cipher for authentication and encryption. The standard originally specified a 40-bit, preshared encryption key -- a 104-bit key was later made available after a set of restrictions from the U.S. government was lifted. The key must be manually entered and updated by an administrator.
The key is combined with a 24-bit initialization vector (IV) in an effort to strengthen the encryption.

However, the small size of the IV increases the likelihood that keys will be reused, which, in turn, makes them easier to crack. This characteristic, along with several other vulnerabilities -- including problematic authentication mechanisms -- makes WEP a risky choice for wireless security.

Wi-Fi Protected Access (WPA)

- The numerous flaws in WEP revealed the urgent need for an alternative, but the deliberately slow and careful processes required to write a new security specification posed a conflict. In response, in 2003, the Wi-Fi Alliance released WPA as an interim standard, while the Institute of Electrical and Electronics Engineers (IEEE) worked to develop a more advanced, long-term replacement for WEP.
- WPA has discrete modes for enterprise users and for personal use. The enterprise mode, WPA-EAP, uses more stringent 802.1x authentication with the Extensible Authentication Protocol, or EAP. The personal mode, WPA-PSK, uses presaged keys for simpler implementation and management among consumers and small offices. Enterprise mode requires the use of an authentication server.
- Although WPA is also based on the RC4 cipher, it introduced several enhancements to encryption -- namely, the use of the Temporal Key Integrity Protocol (TKIP). The protocol contains a set of functions to improve wireless LAN security: the use of 256-bit keys, per-packet key mixing -- the generation of a unique key for each packet -- automatic broadcast of updated keys, a message integrity check, a larger IV size (48 bits) and mechanisms to reduce IV reuse.
- WPA was designed to be backward-compatible with WEP to encourage quick, easy adoption. Network security professionals were able to support the new standard on many WEP-based devices with a simple firmware update. This framework, however, also meant the security it provided was not as robust as it could be.

Wi-Fi Protected Access 2 (WPA2)

- As the successor to WPA, the WPA2 standard was ratified by the IEEE in 2004 as 802.11i. Like its predecessor, WPA2 also offers enterprise and personal modes. Although WPA2 still has vulnerabilities, it is considered the most secure wireless security standard available.
- WPA2 replaces the RC4 cipher and TKIP with two stronger encryption and authentication mechanisms: the Advanced Encryption Standard (AES) and Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP), respectively. Also meant to be backward-compatible, WPA2 supports TKIP as a fallback if a device cannot support CCMP.

- Developed by the U.S. government to protect classified data, AES is composed of three symmetric block ciphers. Each encrypts and decrypts data in blocks of 128 bits using 128-, 192- and 256-bit keys. Although the use of AES requires more computing power from APs and clients, ongoing improvements in computer and network hardware have mitigated performance concerns.
- CCMP protects data confidentiality by allowing only authorized network users to receive data, and it
uses cipher block chaining message authentication code to ensure message integrity.

- WPA2 also introduced more seamless roaming, allowing clients to move from one AP to another on the same network without having to reauthenticate, through the use of Pairwise Master Key caching or reauthentication.

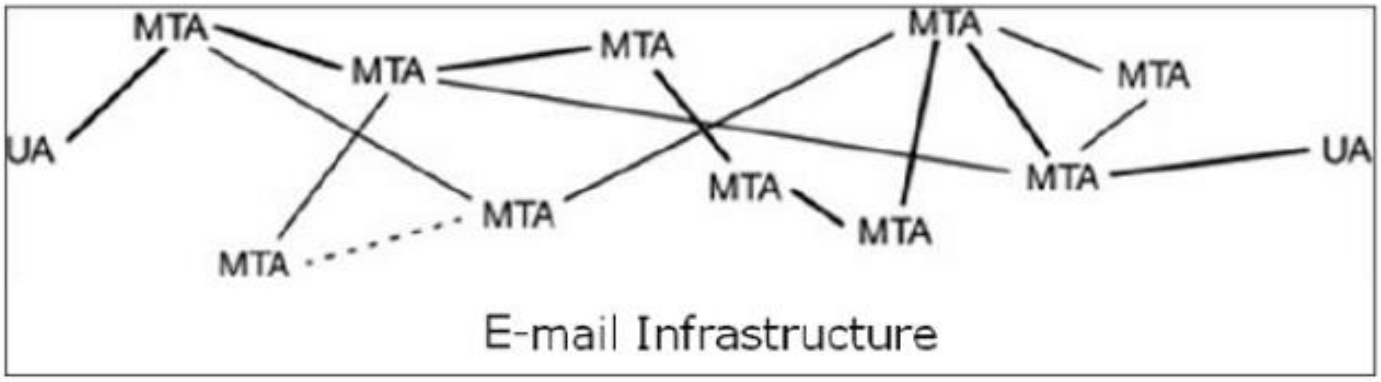
E-mail Security

- Nowadays, e-mail has become very widely used network application. Let's briefly discuss the e-mail infrastructure before proceeding to know about e- mail security protocols.

E-mail Infrastructure

- The simplest way of sending an e-mail would be sending a message directly from the sender's machine to the recipient's machine. In this case, it is essential for both the machines to be running on the network simultaneously. However, this setup is impractical as users may occasionally connect their machines to the network.
- Hence, the concept of setting up e-mail servers arrived. In this setup, the mail is sent to a mail server which is permanently available on the network. When the recipient's machine connects to the network, it reads the mail from the mail server.

- In general, the e-mail infrastructure consists of a mesh of mail servers, also termed as Message Transfer Agents (MTAs) and client machines running an e-mail program comprising of User Agent (UA) and local MTA.
- Typically, an e-mail message gets forwarded from its UA, goes through the mesh of MTAs and finally reaches the UA on the recipient's machine.



- **The protocols used for e-mail are as follows −**
- Simple mail Transfer Protocol (SMTP) used for forwarding e-mail messages.
- Post Office Protocol (POP) and Internet Message Access Protocol (IMAP) are used to retrieve the messages by recipient from the server.

MIME

- Basic Internet e-mail standard was written in 1982 and it describes the format of e-mail message exchanged on the Internet. It mainly supports e-mail message written as text in basic Roman alphabet.
- By 1992, the need was felt to improve the same. Hence, an additional standard Multipurpose Internet Mail

Extensions (MIME) was defined. It is a set of extensions to the basic Internet E-mail standard. MIME provides an ability to send e-mail using characters other than those of the basic Roman alphabet such as Cyrillic alphabet (used in Russian), the Greek alphabet, or even the ideographic characters of Chinese.

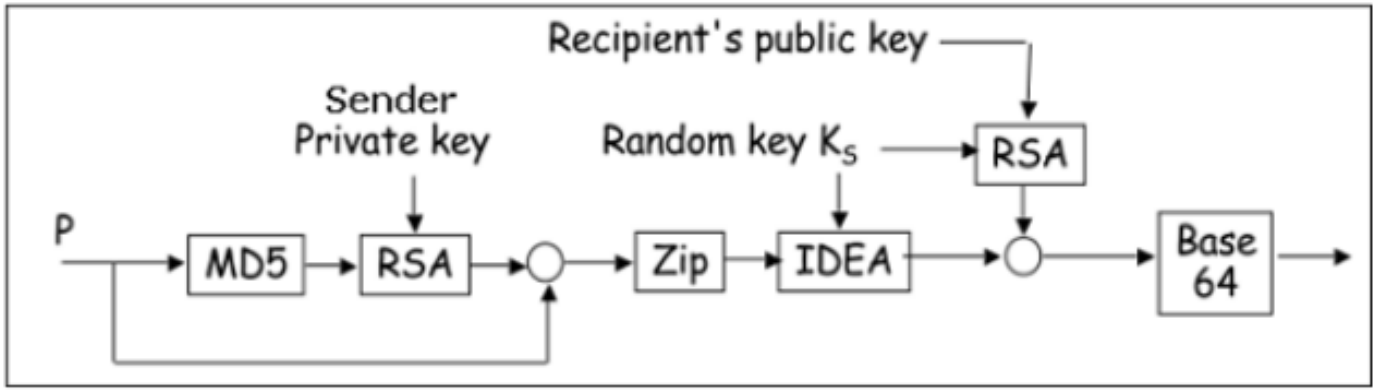
- Another need fulfilled by MIME is to send non-text contents, such as images or video clips. Due to this features, the MIME standard became widely adopted with SMTP for e-mail communication.

E-Mail Security Services

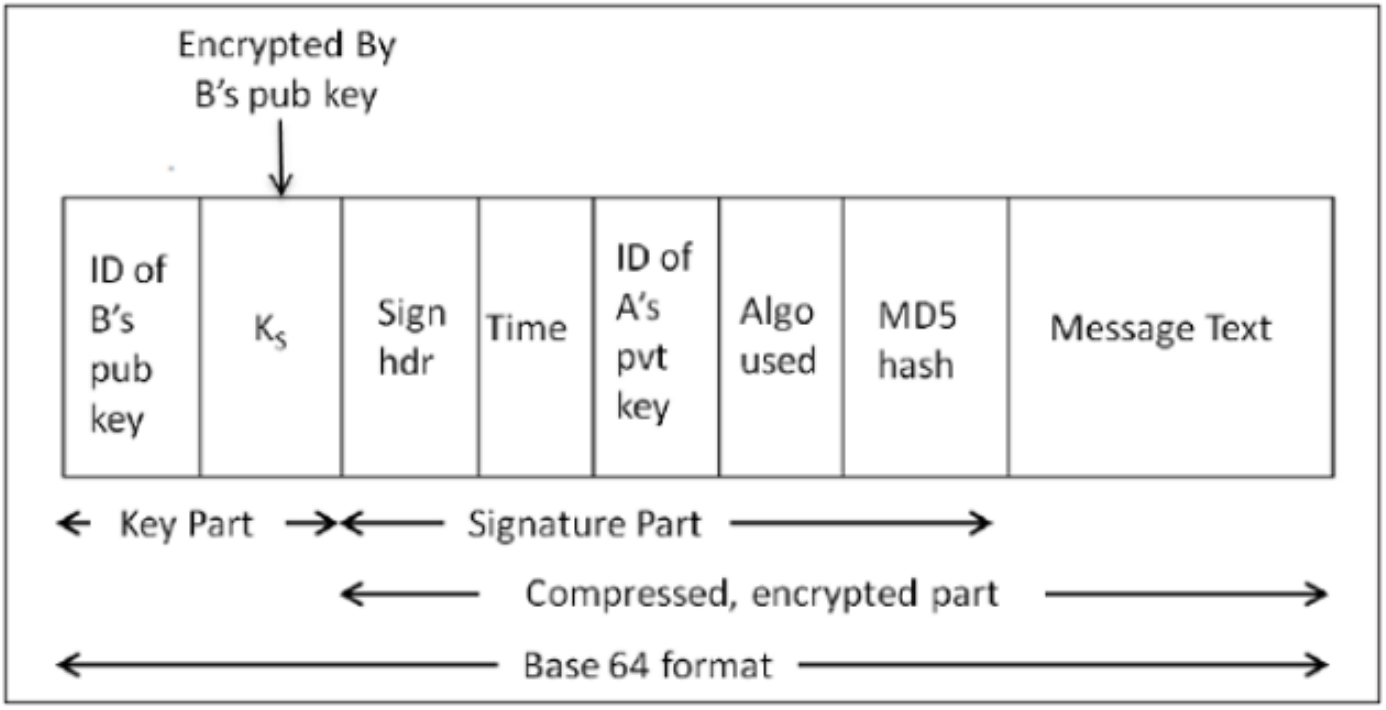
- Confidentiality − E-mail message should not be read by anyone but the intended recipient.
- Authentication − E-mail recipient can be sure of the identity of the sender.
- Integrity − Assurance to the recipient that the e-mail message has not been altered since it was transmitted by the sender.
- Non-repudiation − E-mail recipient is able to prove to a third party that the sender really did send the message.
- Proof of submission − E-mail sender gets the confirmation that the message is handed to the mail delivery system.
- Proof of delivery − Sender gets a confirmation that the recipient received the message.

PGP

- Pretty Good Privacy (PGP) is an e-mail encryption scheme. It has become the de-facto standard for providing security services for e-mail communication.
- As discussed above, it uses public key cryptography, symmetric key cryptography, hash function, and digital signature. It provides −
- Privacy
- Sender Authentication
- Message Integrity
- Non-repudiation



- Along with these security services, it also provides data compression and key management support. PGP uses existing cryptographic algorithms such as RSA, IDEA, MD5, etc., rather than inventing the new ones.



Working of PGP

- Hash of the message is calculated. (MD5 algorithm)
- Resultant 128 bit hash is signed using the private key of the sender (RSA Algorithm).
- The digital signature is concatenated to message, and the result is compressed.
- A 128-bit symmetric key, KS is generated and used to encrypt the compressed message with IDEA.
- KS is encrypted using the public key of the recipient using RSA algorithm and the result is appended to the encrypted message.
- The format of PGP message is shown in the following diagram. The IDs indicate which key is used to encrypt KS and which key is to be used to verify the signature on the hash.

- In PGP scheme, a message in signed and encrypted, and then MIME is encoded before transmission

PGP Certificate

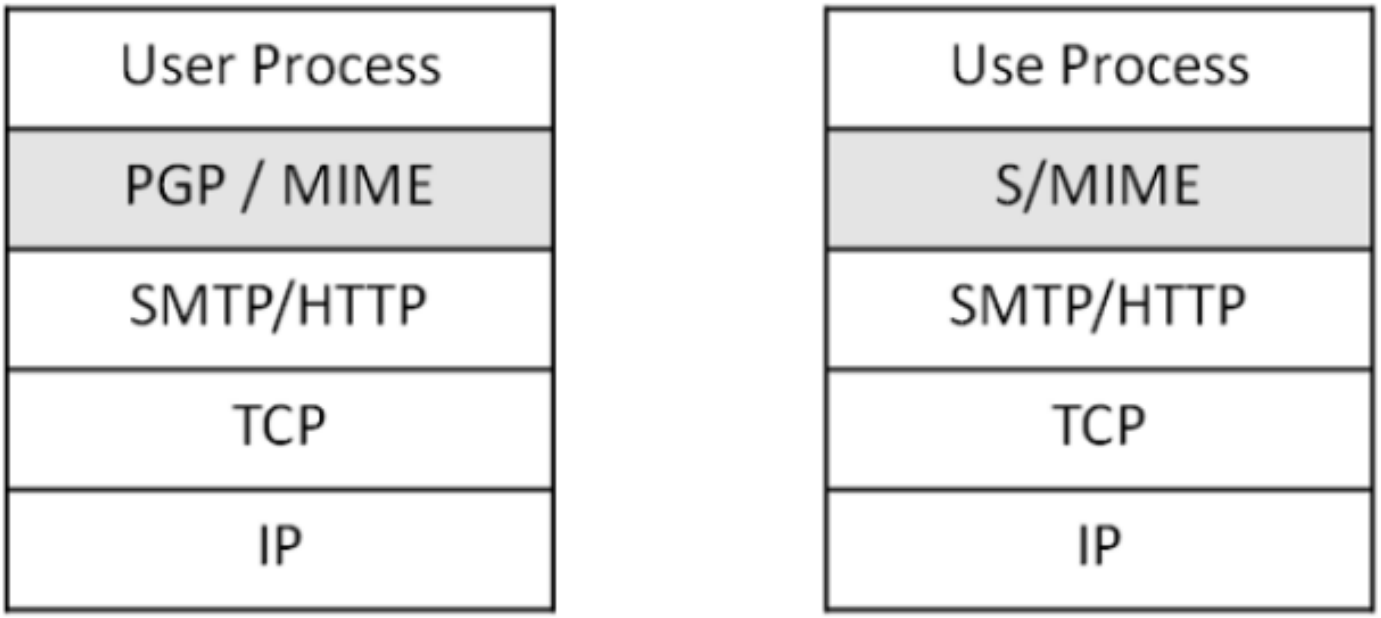
- PGP key certificate is normally established through a chain of trust. For example, A's public key is signed by B using his public key and B's public key is signed by C using his public key. As this process goes on, it establishes a web of trust.
- In a PGP environment, any user can act as a certifying authority. Any PGP user can certify another PGP user's public key. However, such a certificate is only valid to another user if the user recognizes the certifier as a trusted introducer.
- Several issues exist with such a certification method. It may be difficult to find a chain leading from a known and trusted public key to desired key. Also, there might be multiple chains which can lead to different keys for desired user.
- PGP can also use the PKI infrastructure with certification authority and public keys can be certified by CA (X.509 certificate).

S / MIME

- S/MIME stands for Secure Multipurpose Internet Mail Extension. S/MIME is a secure e-mail standard. It is based on an earlier non-secure e-mailing standard called MIME.
- S/MIME approach is similar to PGP. It also uses public key cryptography, symmetric key cryptography, hash functions, and digital signatures. It provides similar security services as PGP for e-mail communication.
- The most common symmetric ciphers used in S/MIME are RC2 and TripleDES. The usual public key method is RSA, and the hashing algorithm is SHA-1 or MD5.
- S/MIME specifies the additional MIME type, such as “application/pkcs7-mime”, for data enveloping after encrypting. The whole MIME entity is encrypted and packed into an object. S/MIME has standardized cryptographic message formats (different from PGP). In fact, MIME is extended with some keywords to identify the encrypted and/or signed parts in the message.
- S/MIME relies on X.509 certificates for public key distribution. It needs top-down hierarchical PKI for certification support.

Employability of S/MIME

- Due to the requirement of a certificate from certification authority for implementation, not all users can take advantage of S/MIME, as some may wish to encrypt a message, with a public/private key pair. For example, without the involvement or administrative overhead of certificates.
- In practice, although most e-mailing applications implement S/MIME, the certificate enrollment process is complex. Instead PGP support usually requires adding a plug-in and that plug-in comes with all that is needed to manage keys. The Web of Trust is not really used. People exchange their public keys over another medium. Once obtained, they keep a copy of public keys of those with whom e-mails are usually exchanged.
- Implementation layer in network architecture for PGP and S/MIME schemes is shown in the following image. Both these schemes provide application level security of for e-mail communication.



One of the schemes, either PGP or S/MIME, is used depending on the environment. A secure e-email communication in a captive network can be provided by adapting to PGP. For e-mail security over Internet, where mails are exchanged with new unknown users very often, S/MIME is considered as a good option.

IP Security Overview

- The standard Internet communication protocol is completely unprotected, allowing hosts to inspect or modify data in transit. Adding IPSec to the system will resolve this limitation by providing strong encryption, integrity, authentication and replay protection.

What Security Problem

- Todays Internet is primarily comprised of :
- Public
- Un-trusted
- Unreliable IP networks

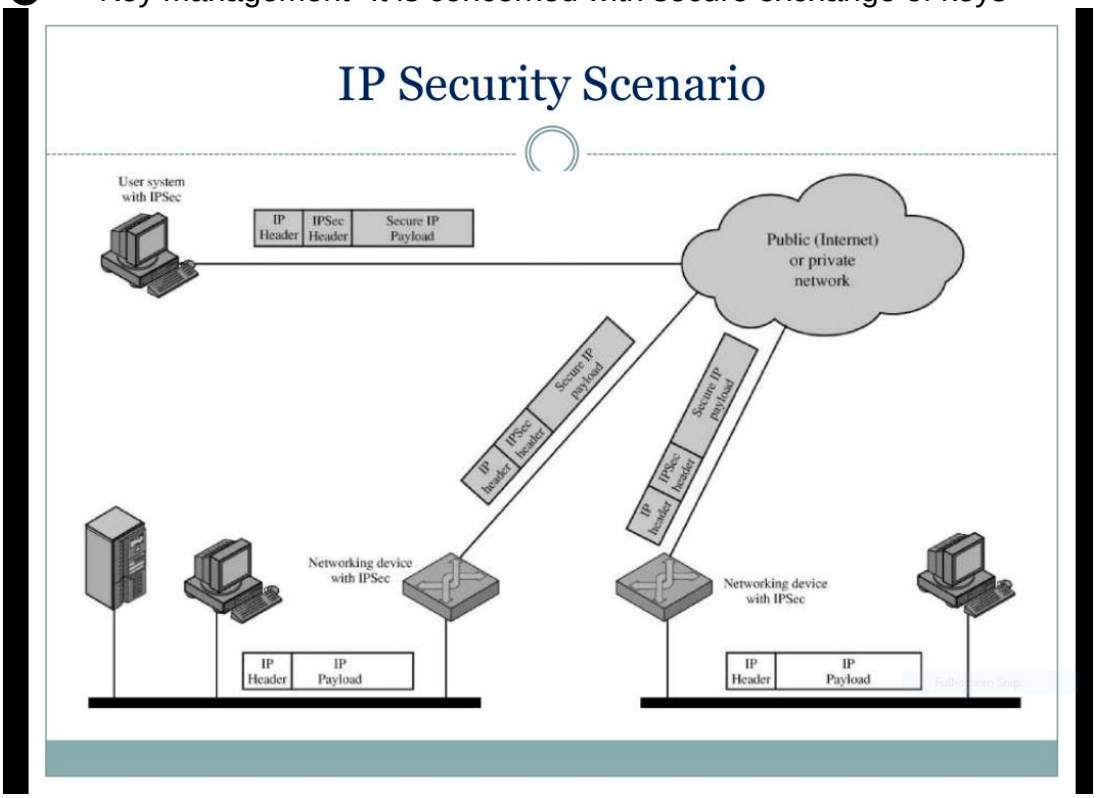
- Because of this inherent lack of security, the Internet is subject to various types of threats…

Internet Threats

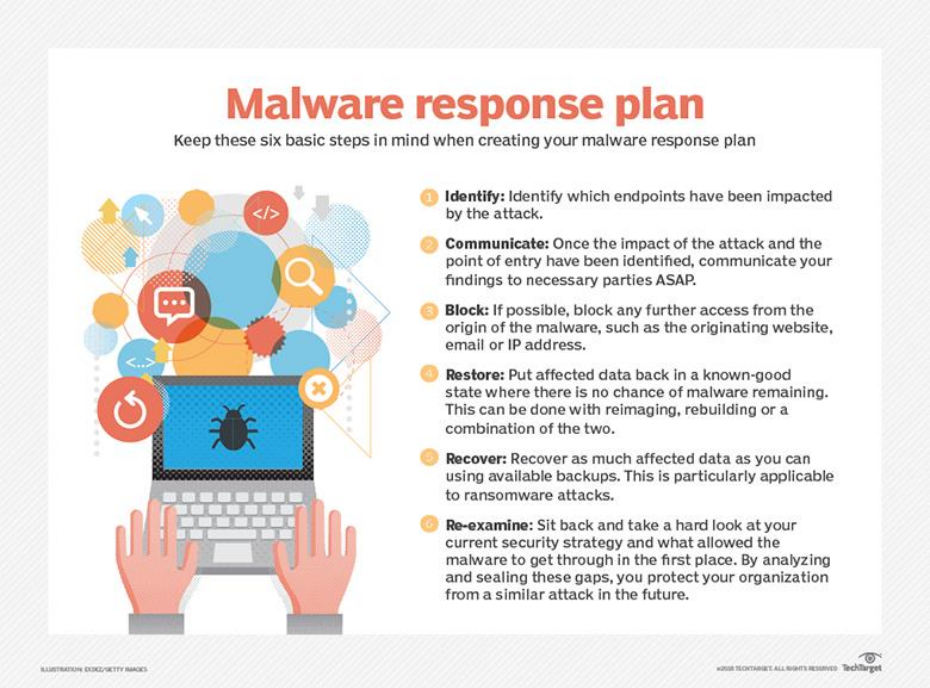
- Data integrity
  - The contents of a packet can be accidentally or deliberately modified.
- Identity spoofing
  - The origin of an IP packet can be forged.
- Anti-reply attacks
  - Unauthorized data can be retransmitted.
- Loss of privacy
  - The contents of a packet can be examined in transit.

IP SECURITY

- Authentication- The authentication mechanism ensures that the received packet was sent by the identified source. It also assures that the packet has not been altered in transit.
- Confidentiality- The confidentiality facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties.
- Key management- It is concerned with secure exchange of keys



Malware



Perhaps the most sophisticated types of threats to computer systems are presented by programs that exploit vulnerabilities in computing systems. Such threats are referred to as malicious software, or malware.
Malicious software can be divided into two categories: those that need a host program, and those that are independent. The former, referred to as parasitic, are essentially fragments of programs that cannot exist independently of some actual application program, utility, or system program. Viruses, logic bombs, and backdoors are examples. Independent malware is a self-contained program that can be scheduled and run by the operating system. Worms and bot programs are examples.

Malicious software can be classified into two types: Replicating and Non- Replicating
Non- Replicating are programs or fragments of programs that are activated by a trigger. Examples are logic bombs, backdoors, and bot programs. Replicating consist of either a program fragment or an independent program that, when executed, may produce one or more copies of itself to be computer virus
A computer virus is a piece of software that can “infect” other programs by modifying them; the modification includes injecting the original program with a routine to make copies of the virus program, which can then go on to infect other programs.
Like its biological counterpart, a computer virus carries in its instructional code the recipe for making perfect copies of itself. The typical virus becomes embedded in a program on a computer.
Then, whenever the infected computer comes into contact with an uninfected piece of software, a fresh copy of the virus passes into the new program. Thus, the infection can be spread from computer to computer by unsuspecting users who either swap disks or send programs to one another over a network. In a network environment, the ability to access applications and system services on other computers provides a perfect culture for the spread of a virus.

- A virus can do anything that other programs do.The difference is that a virus attaches itself to another program and executes secretly when the host program is run. Once a virus is executing, it can perform any function, such as erasing files and programs that is allowed by the privileges of the current user.

Backdoor

- A backdoor, also known as a trapdoor, is a secret entry point into a program that allows someone who is aware of the backdoor to gain access without going through the usual security access procedures. Programmers have used backdoors legitimately for many years to debug and test programs; such a backdoor is called a maintenance hook.
- This usually is done when the programmer is developing an application that has an authentication procedure, or a long setup, requiring the user to enter many different values to run the application. To debug the program, the developer may wish to gain special privileges or to avoid

all the necessary setup and authentication. The programmer may also want to ensure that there is a method of activating the program should something be wrong with the authentication procedure that is being built into the application. The backdoor is code that recognizes some special sequence of input or is triggered by being run from a certain user ID or by an unlikely sequence of events. Backdoors become threats when unscrupulous programmers use them to gain unauthorized access.

Logic Bomb

One of the oldest types of program threat, predating viruses and worms, is the bomb.
The logic bomb is code embedded in some legitimate program that is set to "explode" when certain conditions are met.
Examples of conditions that can be used as triggers for a logic bomb are the presence or absence of certain files, a particular day of the week or date, or a particular user running the application.
Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage.

Trojan horse

A Trojan horse is a useful, or apparently useful, program or command procedure containing hidden code that, when invoked, performs some unwanted or harmful function. Trojan horse programs can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly.

- For example, to gain access to the files of another user on a shared system, a user could create a Trojan horse program that, when executed, changes the invoking user's file permissions so that the files are readable by any user. The author could then induce users to run the program by placing it in a common directory and naming it such that it appears to be a useful utility program or application.

Trojan horses fit into one of three models:

Continuing to perform the function of the original program and additionally performing a separate malicious activity
Continuing to perform the function of the original program but modifying the function to perform malicious activity (e.g., a Trojan horse version of a login program that collects passwords) or to disguise other malicious activity (e.g., a Trojan horse version of a process listing program that does not display certain processes that are malicious)
Performing a malicious function that completely replaces the function of the original program

Mobile Code

Mobile code refers to programs (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics .
The term also applies to situations involving a large homogeneous collection of platforms (e.g., Microsoft Windows).
Mobile code is transmitted from a remote system to a local system and then executed on the local system without the user's explicit instruction.
Mobile code often acts as a mechanism for a virus, worm, or Trojan horse to be transmitted to the user's workstation. In other cases, mobile code takes advantage of vulnerabilities to perform its own exploits, such as unauthorized data access or root compromise.
Popular vehicles for mobile code include Java applets, ActiveX, JavaScript, and VBScript.
The most common ways of using mobile code for malicious operations on local system are cross-site scripting, interactive and dynamic Web sites, e-mail attachments, and downloads from untrusted sites or of untrusted software.

Spyware/Adware

- Spyware secretly records information about a user and forwards it to third parties. The information gathered may cover files accessed on the computer, a user's online activities or even user's keystrokes.
- Adware as the name interprets displays advertising banners while a program is running. Adware can also work like spyware, it is deployed to gather confidential information. Basically, to spy on and gather information from a victim's computer.

Rootkit

A rootkit is a malicious software that alters the regular functionality of an OS on a computer in a stealthy manner. The altering helps the hacker to take full control of the system and the hacker acts as the system administrator on the victim's system. Almost all the rootkits are designed to hide their existence.

Keyloggers

Keyloggers, also called system monitors, are used to see nearly everything a user does on their computer. This includes emails, opened web-pages, programs and keystrokes.

A firewall is a system designed to prevent unauthorized access to or from a private network.You can implement a firewall in either hardware or software form, or a combination of both. Firewalls prevent unauthorized internetusers from accessing private networks connected to the internet, especially intranets. All messages entering or leaving the intranet (the local network to which you are connected) must pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria.
A firewall may act as a packet filter. It can operate as a positive filter, allowing to pass only packets that meet specific criteria, or as a negative filter, rejecting any packet that meets certain criteria.
Depending on the type of firewall, it may examine one or more protocol headers in each packet, the payload of each packet, or the pattern generated by a sequence of packets

Characteristics of Firwall

1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall.
2. Only authorized traffic, as defined by the local security policy, will be allowed to pass.Various types of firewalls are used, which implement various types of security policies.
3. The firewall itself is immune to penetration. This implies the use of a hardened system with a secured operating system. Trusted computer systems are suitable for hosting a firewall and often required in government applications.

Limitation of Firewall

- Firewalls have their limitations, including the following:
- 1.The firewall cannot protect against attacks that bypass the firewall. Internal systems may have dial-out capability to connect to an ISP. An internal LAN may support a modem pool that provides dial-in capability for traveling employees and telecommuters.
- 2. The firewall may not protect fully against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.
- 3. An improperly secured wireless LAN may be accessed from outside the organization. An internal firewall that separates portions of an enterprise network cannot guard against wireless communications between local systems on different sides of the internal firewall.
- 4. A laptop, PDA, or portable storage device may be used and infected outside the corporate network, and then attached and used internally.

- **Types of Firewalls**

- Packet filtering firewall
- Application proxy firewall
- Stateful inspection firewall
- Circuit – level proxy firewall

Packet Filtering Firewall

A packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet (Figure 11.1b).The firewall is typically configured to filter packets going in both directions (from and to the internal network). Filtering rules are based on information contained in a network packet: Source IP address, Destination IP address, Source and destination transport-level address, IP protocol field, Interface

The packet filter is typically set up as a list of rules based on matches to fields in the IP or TCP header. If there is a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet. If there is no match to any rule, then a default action is taken.Two default policies are possible:

- Default = discard: That which is not expressly permitted is prohibited.
- Default = forward: That which is not expressly prohibited is permitted.

Packet filtering works well for small networks but when applied to larger networks can quickly become very complex and difficult to configure. Packet filtering also cannot be used for content-based filtering and cannot, for instance, remove e-mail attachments. This type of firewall has little or no logging capability, making it difficult to determine if it's been attacked.

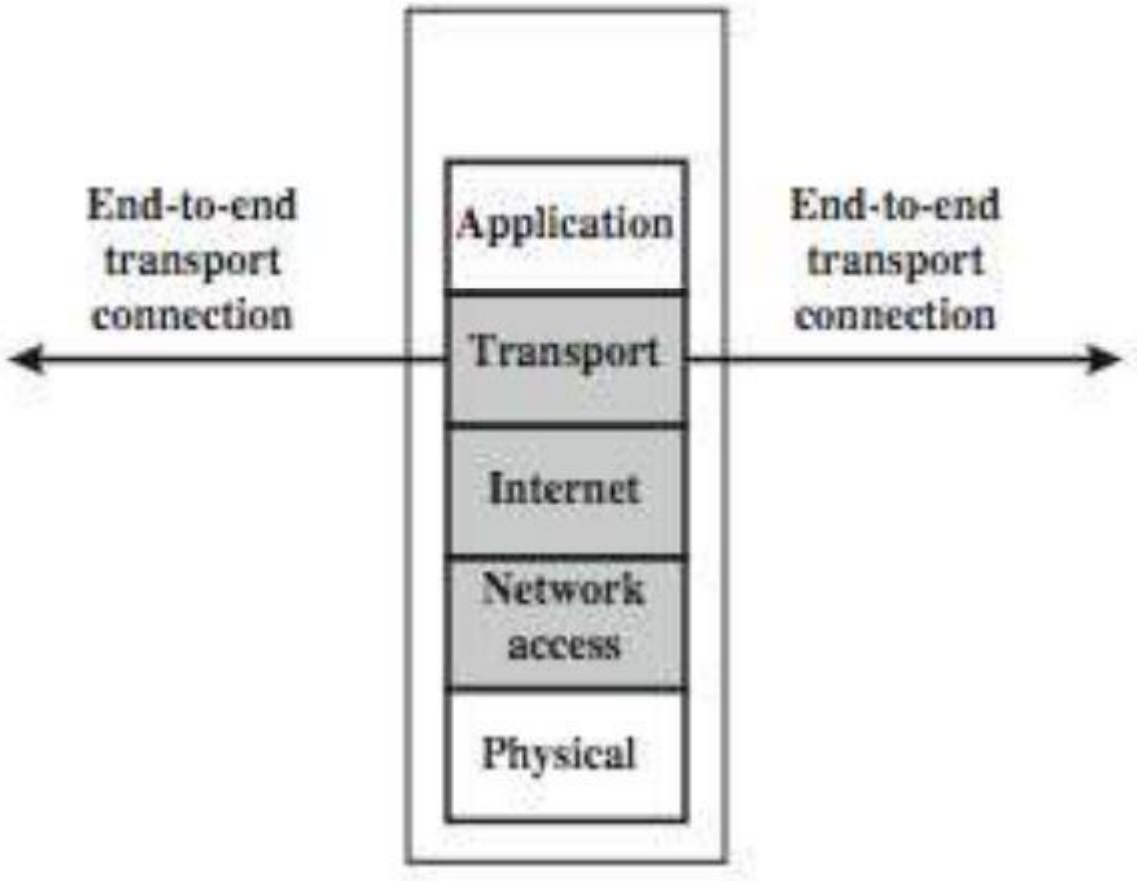
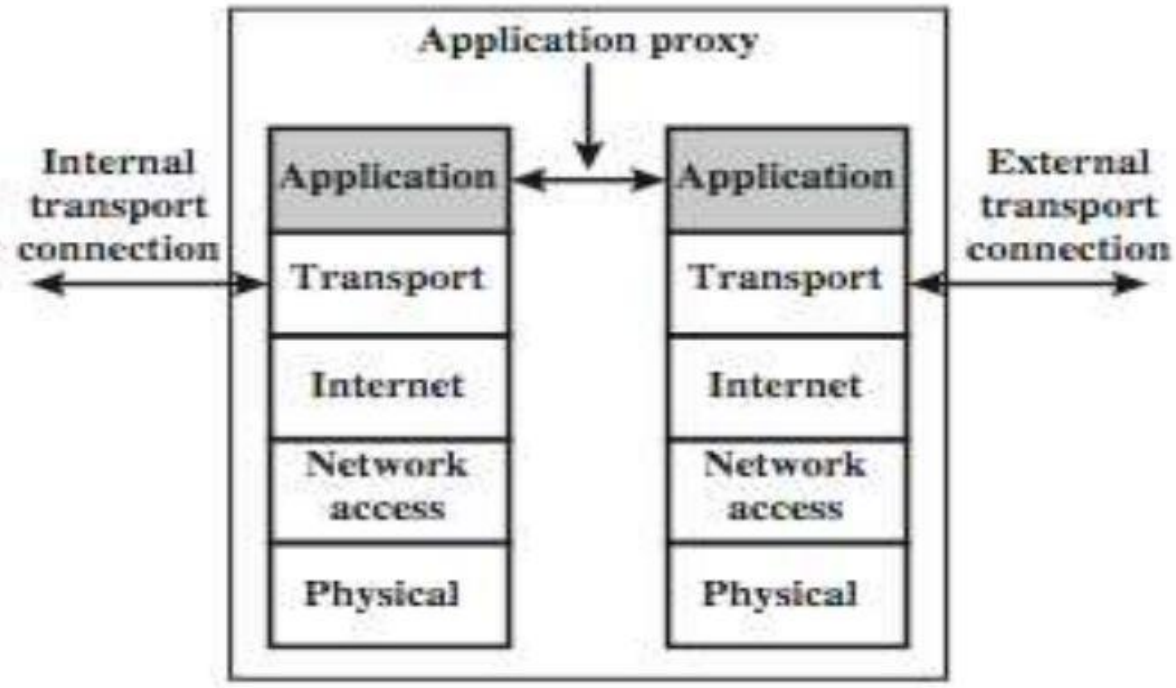


Figure packet filtering

Application proxy Firewall

The more sophisticated proxy or application layer firewalls deal with network traffic by passing all packets through a separate "proxy" application that examines data at an application level.

A proxy firewall doesn't allow a direct connection between your network and the Internet. Instead it accepts requests and executes them on behalf of the user.
For instance, if you're behind a proxy firewall and
type http://www.blackbox.co.uk, the request goes to the firewall, which gets the page on your behalf and passes it to you. This process is transparent to users
This proxy system enables you to set a firewall to accept or reject packets based on addresses, port information and application information. For instance, you can set the firewall to filter out all incoming packets belonging to EXE files, which are often infected with viruses and worms. Proxy firewalls generally keep very detailed logs, including information on the data portions of packets.
Proxy firewalls are slower and require more hardware than packet filtering; however, their greater versatility enables you to enforce tighter security policies.



Stateful Inspection Firewalls

- It keeps track of the state of active connections and uses this information to decide which packets to allow through it, i.e., it adapts itself to the current exchange of information, unlike the normal packet filters/stateless packet filters, which have hardcoded routing rules.
- This firewall tracks each communications session from start to end and enforces set rules based on protocol, port and source and destination addresses. By maintaining all session data, the firewall can quickly verify that new incoming packets meet the criteria for authorized traffic. Packets that aren't part of an authorized session are rejected.
- Stateful inspection firewalls have the advantage of being both smart and fast.

Circuit-Level Gateways –

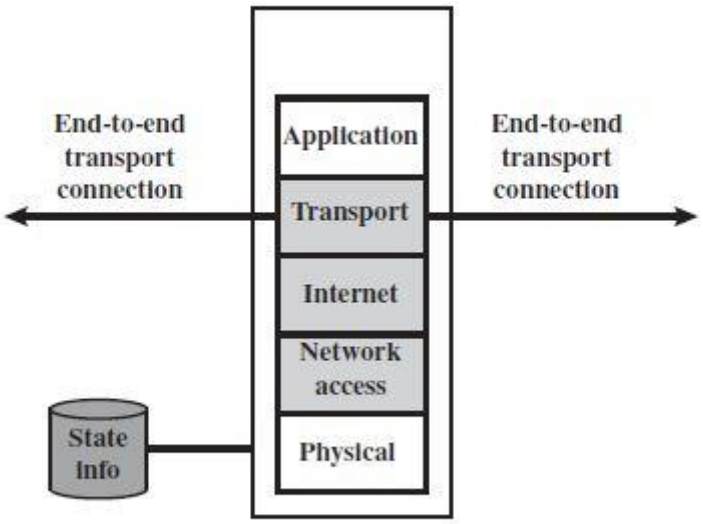
- It works at the session layer of the OSI Model. It is the advanced variation of Application Gateway. It acts as a virtual connection between the remote host and the internal users by creating a new connection between itself and the remote host.

- It also changes the source IP address in the packet and puts its own address at the place of source IP address of the packet from end users.
- This way, the IP addresses of the internal users are hidden and secured from the outside world.

Firewall Basing

It is common to base a firewall on a stand-alone machine running a common operating system, such as UNIX or Linux.

Firewall functionality can also be implemented as a software module in a router or LAN switch. In this section, we look at some additional firewall basing considerations.



(c) Stateful inspection firewall

Bastion Host

A bastion host is a system identified by the firewall administrator as a critical strong point in the network's security.

It serves as a platform for an application – level or circuit – level gateway.

Some common characteristics :

i. Executes a secure version of its OS, making it a hardened system.

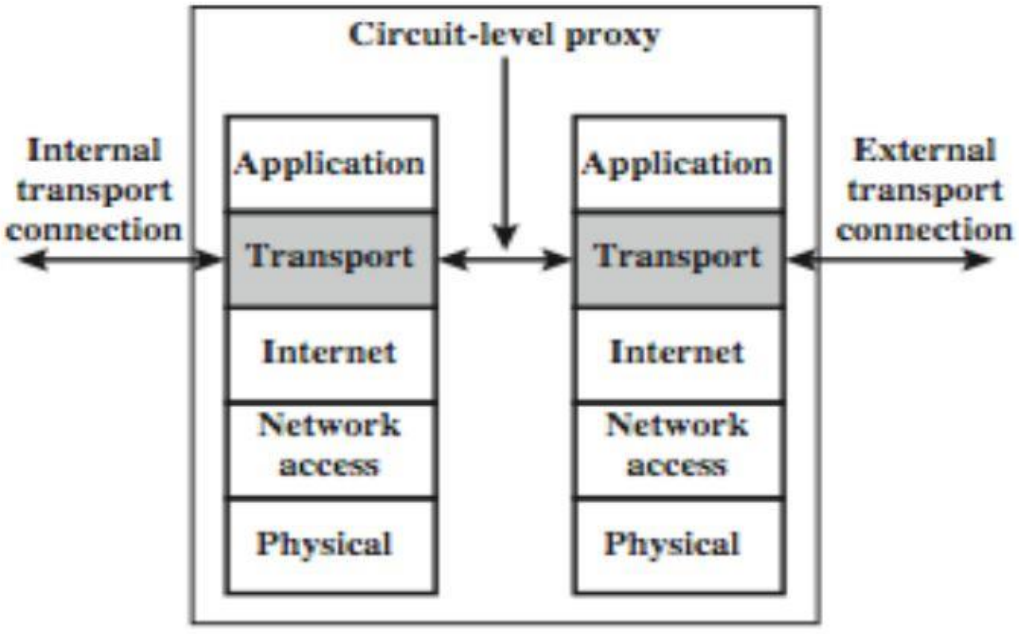
ii. Only the services that the network administrator considers essential are installed on the bastion host.

iii. It may require some additional authentication before user is allowed access to the proxy services.

iv. Each proxy is configured to support only a subset of the standard application's command set.

v. Each proxy is configured to allow access only to specific host systems.

vi. Each proxy maintains detailed audit information by logging all traffic.



- Each proxy module is a very small software package specifically designed for network security.

- Each proxy is independent of other proxies on the bastion host.
- A proxy generally performs no disk access other than to read its initial configuration file.
- Each proxy runs as a nonprivilaged user in a private and secured dictionary on the bastion host.

Host-Based Firewalls

A host-based firewall is a software module used to secure an individual host. Such modules are available in many operating systems or can be provided as an add-on package. Like conventional stand-alone firewalls, host-resident firewalls filter and restrict the flow of packets. A common location for such firewalls is a server. There are several advantages to the use of a server-based or workstation based firewall:

- Filtering rules can be tailored to the host environment. Specific corporate security policies for servers can be implemented, with different filters for servers used for different application.
- Protection is provided independent of topology. Thus both internal and external attacks must pass through the firewall.
- Used in conjunction with stand-alone firewalls, the host-based firewall provides an additional layer of protection. A new type of server can be added to the network, with its own firewall, without the necessity of altering the network firewall configuration.

Personal Firewall

A personal firewall controls the traffic between a personal computer or workstation on one side and the Internet or enterprise network on the other side. Personal firewall functionality can be used in the home environment and on corporate intranets.

Typically, the personal firewall is a software module on the personal computer. In a home environment with multiple computers connected to the Internet, firewall functionality can also be housed in a router that connects all of the home computers to a DSL, cable modem, or other Internet interface.

Personal firewalls are typically much less complex than either server-based firewalls or stand-alone firewalls. The primary role of the personal firewall is to deny unauthorized remote access to the computer. The firewall can also monitor outgoing activity in an attempt to detect and block worms and other malware.

An example of a personal firewall is the capability built in to the Mac OS X operating system.When the user enables the personal firewall in Mac OS X, all inbound connections are denied except for those the user explicitly permits. The list of inbound services that can be selectively reenabled, with their port numbers, includes the following:

The list of inbound services that can be selectively reenabled, with their port numbers, includes the following:

- Personal file sharing (548, 427)
- Windows sharing (139)
- Personal Web sharing (80, 427)
- Remote login - SSH (22)
- FTP access (20-21, 1024-64535 from 20-21)
- Remote Apple events (3031)
- Printer sharing (631, 515)
- IChat Rendezvous (5297, 5298)
- ITunes Music Sharing (3869)
- CVS (2401)

Symmetric Encryption Principles

- Symmetric Cryptography (DES, Triple DES, AES, Key distribution)
- Chiper Lock Mode(Electronic Codebook Mode

Symmetric Encryption Principles

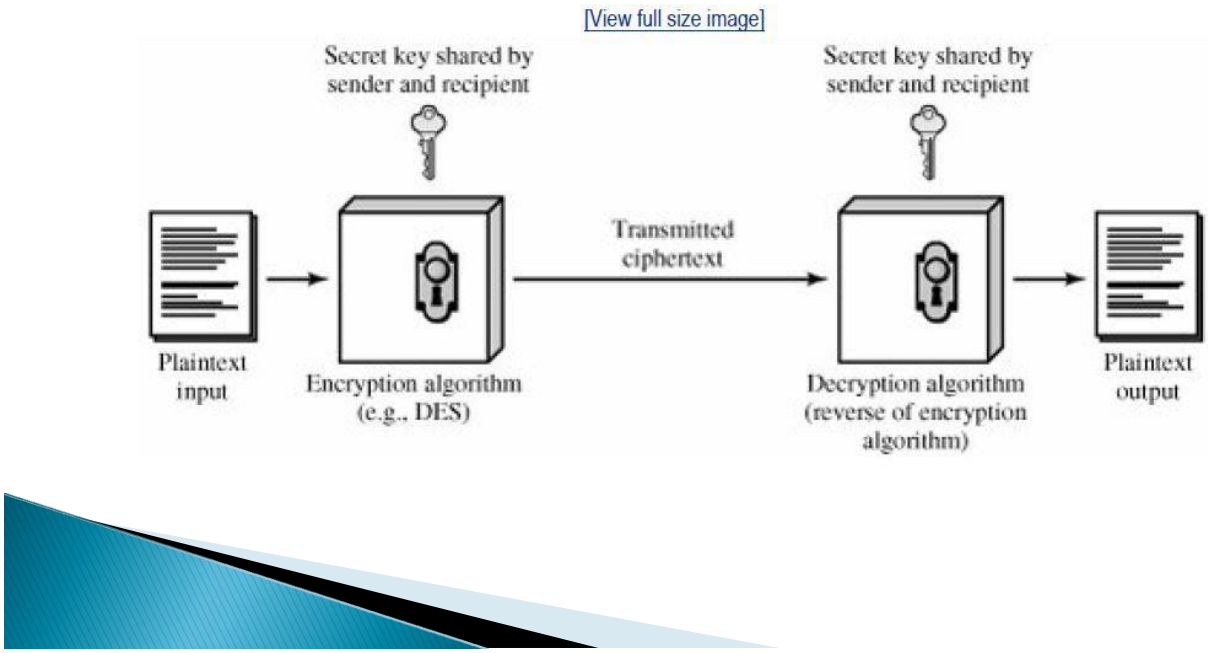
- Symmetric Cipher Model
- Plaintext
- Encryption Algorithm
- Secret Key (known to sender and receiver)
- Ciphertext
- Decryption Algorithm

Symmetric Encryption Principles

Symmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption.

- Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the cip

Figure 2.1. Simplified Model of Conventional Encryption

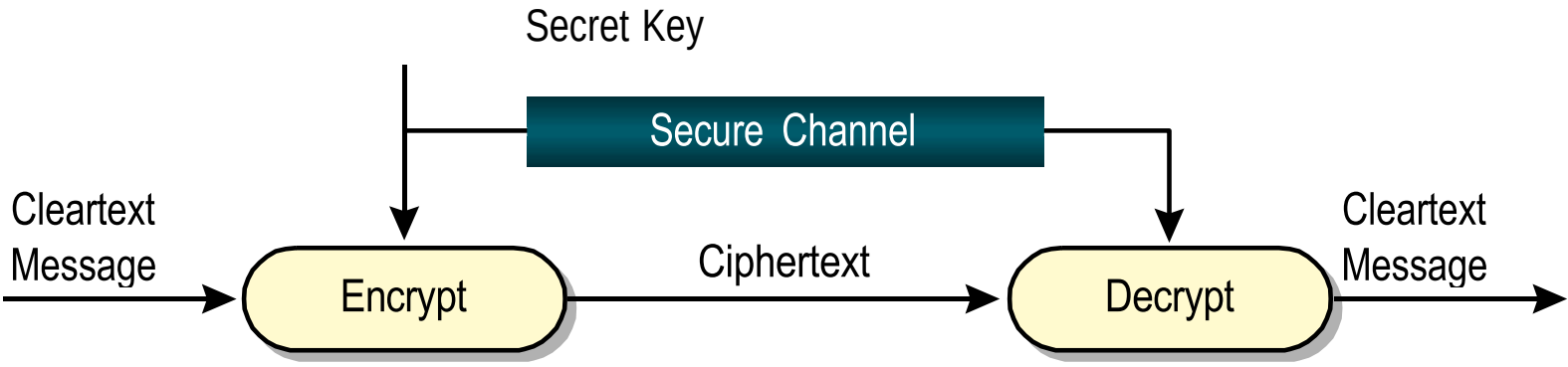


Traditional symmetric ciphers use substitution and/or transposition techniques.

- Rotor machines are sophisticated precomputer hardware devices that use substitution techniques.

Symmetric Key

- A common secret that all parties must know
- Difficult to distribute key securely
- Used by DES, 3DES, AES, Twofish, Blowfish, IDEA, RC5



Atributes of Strong Encryption

Confusion

– Change key values each round
– Performed through substitution
– Complicates plaintext/key relationship
– Interceptor should not be able to predict how ciphertext will change by changing one character

• Diffusion

– Change location of plaintext in ciphertext
– Done through transposition
– Cipher should spread information from plaintext over cipher text

Types of Encryption

- **Block cipher**

Encrypts blocks of data, often 128 bits

- **Stream cipher**

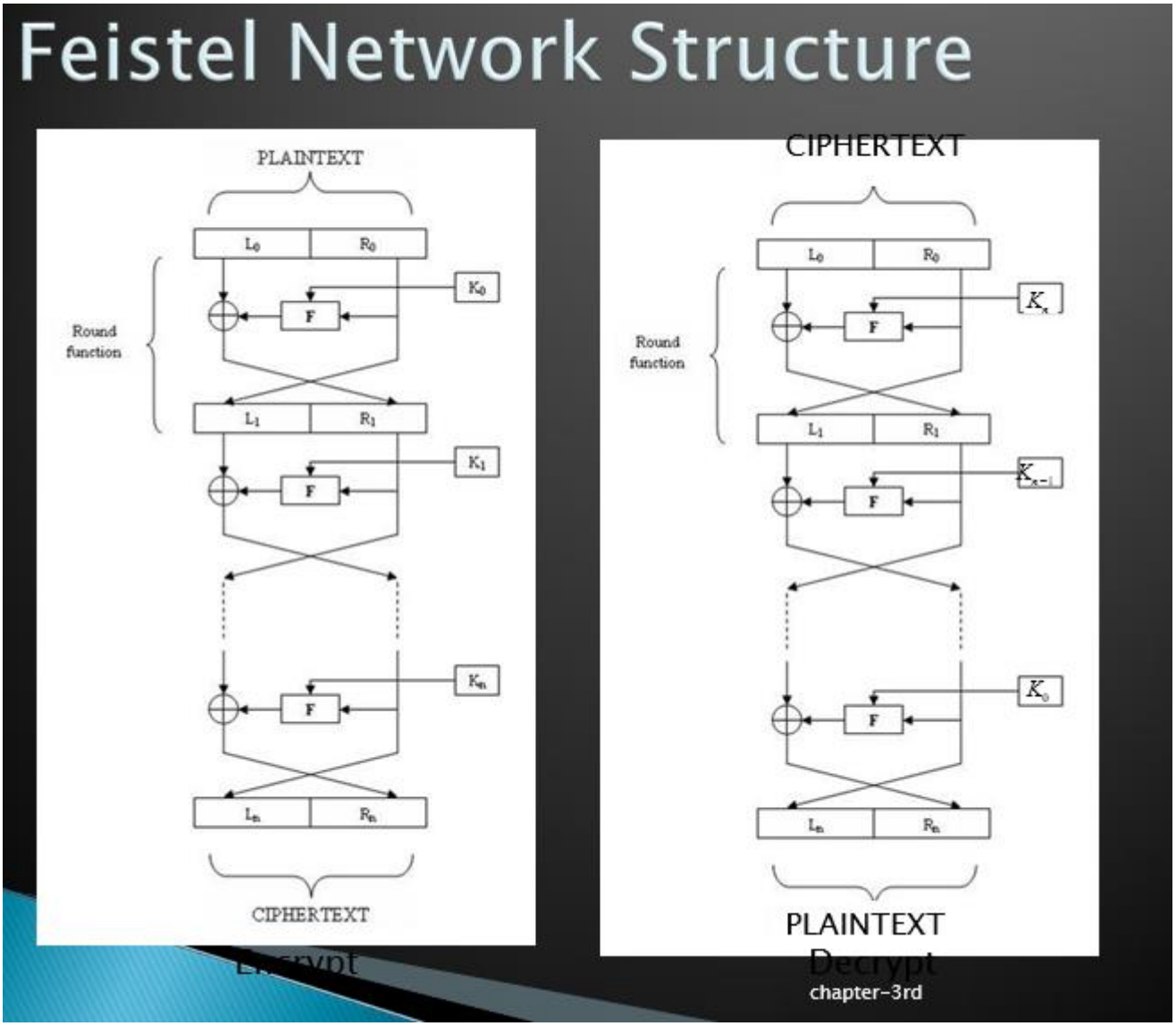
Operates on a continuous stream of data

A **block cipher** is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.

● Many block ciphers have a Feistel structure. Such a structure consists of a number of identical rounds of processing. In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round.

● The **Data Encryption Standard (DES)** has been the most widely used encryption algorithm until recently. It exhibits the classic Feistel structure. DES uses a 64- bit block and a 56-bit key.

- A **Feistel block cipher** operates on a plaintext block of n bits to produce a ciphertext block of n bits. There are 2^n possible different plaintext blocks and, for the encryption to be reversible (i.e., for decryption to be possible), each must produce a unique ciphertext block. Such a transformation is called reversible, or nonsingular.



Decryption of Piestel Cipher

- The process of decryption with a Feistel cipher is essentially the same as the encryption process. The rule is as follows: Use the ciphertext as input to the algorithm, but use the subkeys Ki in reverse order.
- That is, use Kn in the first round, Kn-1 in the second round, and so on until K1 is used in the last round.This is a nice feature because it means we need not implement two different algorithms, one for encryption and one for decryption.

Data Encryption Standard (DES)

The Data Encryption Standard (DES) is a symmetric- key block cipher published by the National Institute of Standards and Technology (NIST).

- In 1973, NIST published a request for proposals for a national symmetric-key cryptosystem. A proposal from IBM, a modification of a project called Lucifer, was accepted as DES. DES was

published in the Federal Register in March 1975 as a draft of the Federal Information Processing Standard (FIPS).

- There has been considerable controversy over the design, particularly in the choice of a 56-bit key.

Symmetric Encryption

- Uses same "secret key" to encipher and decipher message
  - Encryption methods can be extremely efficient, requiring minimal processing
  - Both sender and receiver must possess encryption key
  - If either copy of key is compromised, an intermediate can decrypt and read messages
  - Key distribution problem

Modern Block Ciphers

- Block ciphers are among the most widely used types of cryptographic algorithms
- Provide secrecy and/or authentication services
- In particular will introduce DES (Data Encryption Standard)
- Most symmetric block ciphers are based on a Feistel Cipher Structure
- Needed since must be able to decrypt ciphertext to recover messages efficiently
- Block ciphers look like an extremely large substitution
- Would need table of 264 entries for a 64- bit block
- Instead create from smaller building blocks

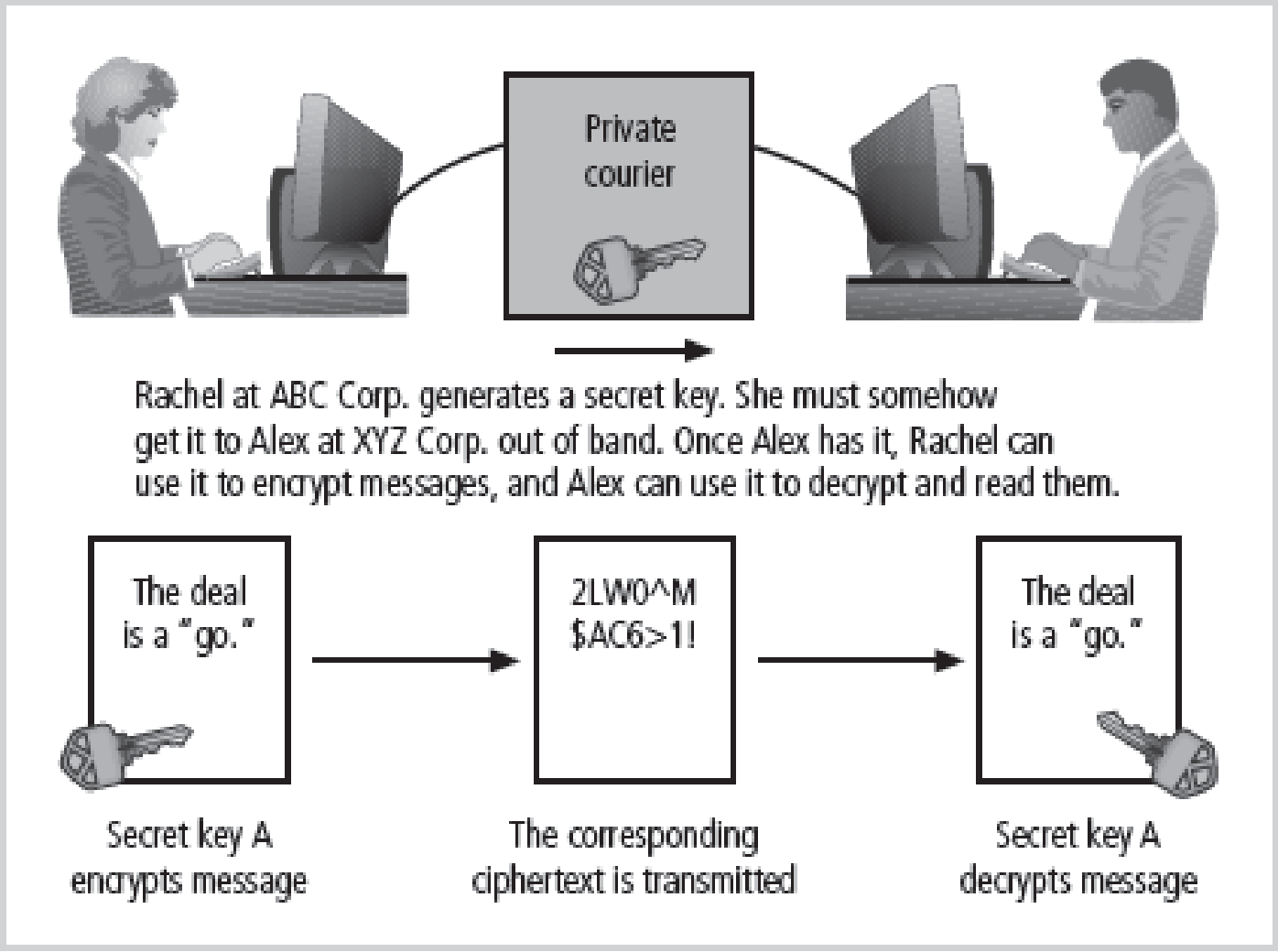


Figure 8-5 Example of Symmetric Encryption

- Data Encryption Standard (DES): one of most popular symmetric encryption cryptosystems
  - 64-bit block size; 56-bit key
  - Adopted by NIST in 1976 as federal standard for encrypting non-classified information
- Triple DES (3DES): created to provide security far beyond DES
- Advanced Encryption Standard (AES): developed to replace both DES and 3DES

DES

- **A block cipher:**
  - encrypts blocks of 64 bits using a 56 bit key
  - outputs 64 bits of ciphertext
- A product cipher
  - basic unit is the bit
  - performs both substitution (S-box) and transposition (permutation) (P-box) on the bits
- Cipher consists of 16 rounds (iterations) of processing
- From the original 56-bit key, 16 subkeys are generated (256)

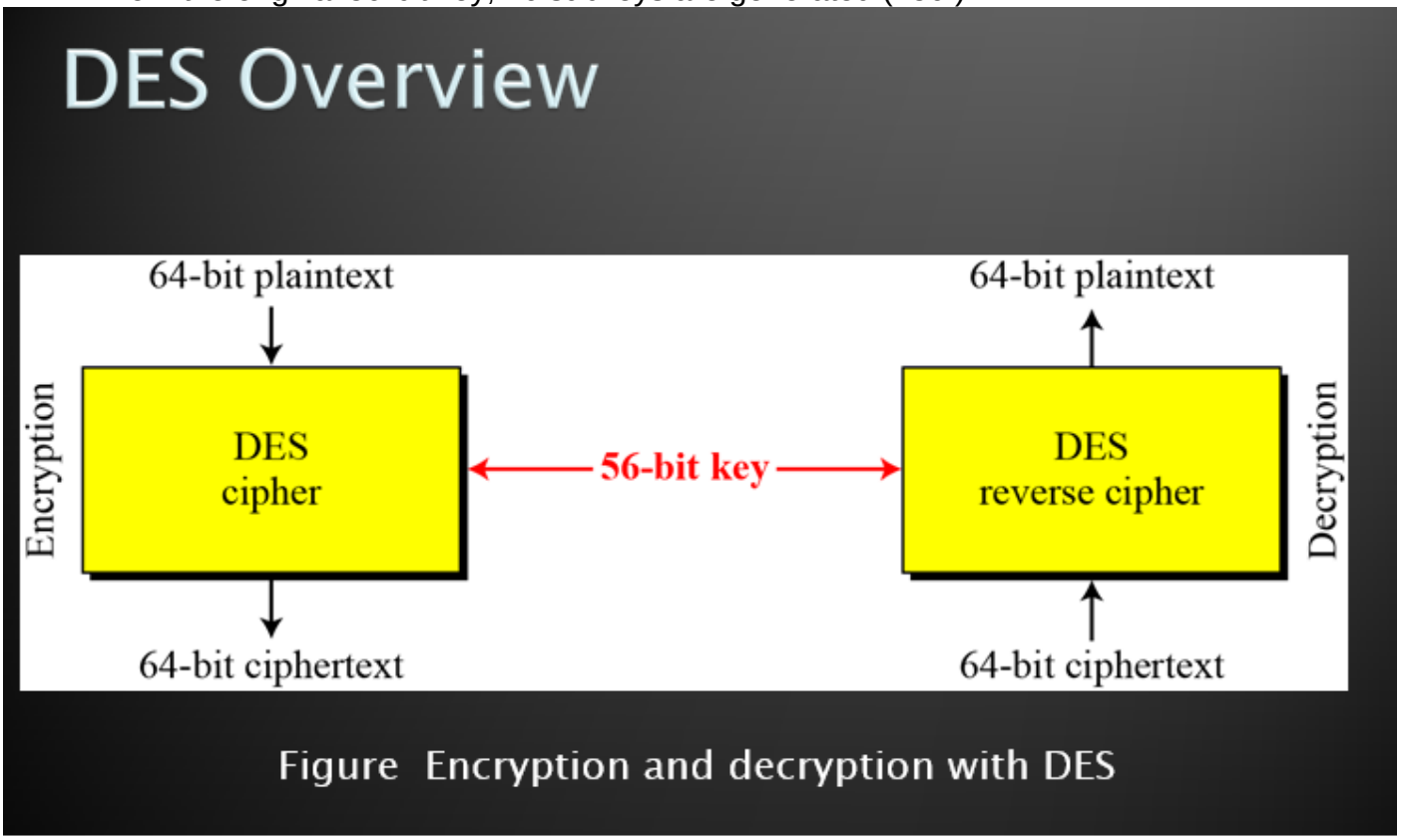


Figure Encryption and decryption with DES

DES Basics

- Fundamentally DES performs only two operations on its input, bit shifting (permutation), and bit substitution.
- The key controls exactly how this process works.
- By doing these operations repeatedly and in a non-linear manner you end up with a result which can not be used to retrieve the original without the key.
- By applying relatively simple operations repeatedly a system can achieve a state of near total randomness.

DES

 As with any encryption scheme, there are two inputs to the encryption function: the plaintext to be encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.

Looking at the left-hand side of the figure, we can see that the processing of the plaintext proceeds in three phases. First, the 64- bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input. This is followed by a phase consisting of 16 rounds of the same function, which involves both permutation and substitution functions. The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the preoutput.

Finally, the preoutput is passed through a permutation (IP-1) that is the inverse of the initial permutation function, to produce the 64-bit ciphertext. With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher.

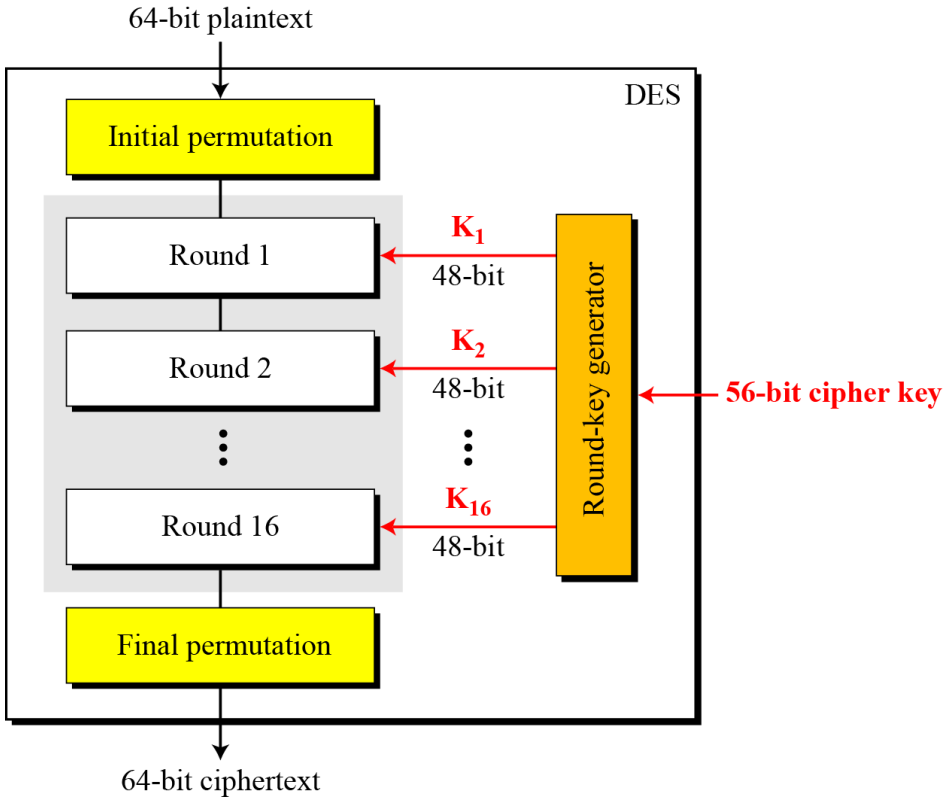
The right-hand portion of shows the way in which the 56-bit key is used. Initially, the key is passed through a permutation function.

Then, for each of the 16 rounds, a subkey (Ki) is produced by the combination of a left circular shift and a permutation. The

permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

DES Structure

The encryption process is made of two permutations (P- boxes), which we call initial and final permutations, and sixteen rounds of complex key dependent calculation.

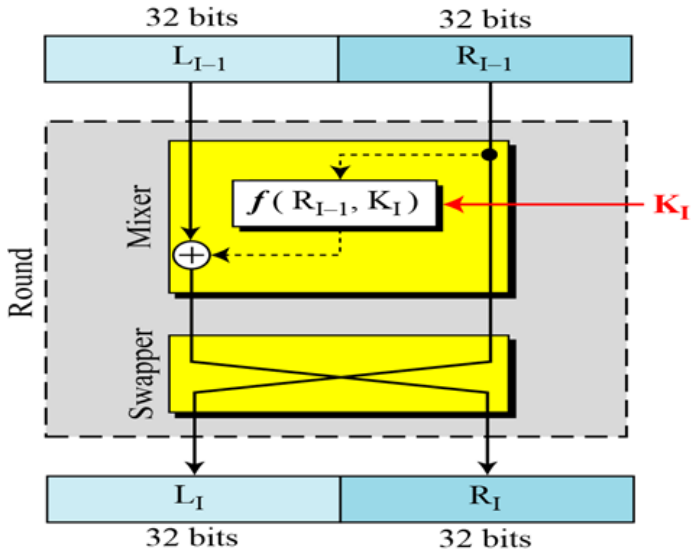


Since RI−1 is a 32-bit input and KI is a 48-bit key, we first need to expand RI−1 to 48 bits.

Continue

The S-boxes do the real mixing (confusion). DES uses 8 S- boxes, each with a 6-bit input and a 4-bit output. See Figure 6.7.

DES uses 16 rounds. Each round of DES is a Feistel cipher



CIPHERTEXT

PLAINTEXT
Encrypt Decrypt

- DES works on 64 bits of data at a time. Each 64 bits of data is iterated on from 1 to 16 times (16 is the DES standard).
- For each iteration a 48 bit subset of the 56 bit key is fed into the encryption block
- Decryption is the inverse of the encryption process.
- The key is usually stored as a 64-bit number, where every eighth bit is a parity bit.
- The parity bits are pitched during the algorithm, and the 56-bit key is used to create 16 different 48-bit subkeys - one for each round.
- Subkeys Generation
  - First, the key is loaded according to the PC-1 and then halved.
  - Then each half is rotated by 2 bits in every round except the first, second, 9th and last rounds.
  - The reason for this is that it makes it secure against related- key cryptanalysis.
  - Then 48 of the 56 bits are chosen according to a compression permutation.
- The subkeys used by the 16 rounds are formed by the key schedule which consists of:
- An initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
- 16 stages consisting of
- selecting 24-bits from each half and permuting them by PC2 for use in function f,
- rotating each half either 1 or 2 places depending on the key rotation schedule

Decrypt must unwind steps of data computation With Feistel design, do encryption steps again Using subkeys 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

- 56-bit keys have 256 = 7.2 x 1016 values

Brute force search looks hard

- Assuming on average half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand year to break the cipher
- Still must be able to recognize plaintext
- Now considering alternatives to DES
- 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
- Particularly problematic on smartcards

- Now have several analytic attacks on DES
- These utilize 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

 DES, as the first important block cipher, has gone through much scrutiny. Among the attempted attacks, three are of interest:

1. Brute-Force Attack
2. Differential Cryptanalysis
3. Linear Cryptanalysis

 Brute-Force Attack

◦ the most basic method of attack is brute force — trying every possible key in turn.

◦ Combining the weakness of short cipher key with the key complement weakness, it is clear that DES can be broken using 255 encryptions.

 Differential Cryptanalysis

◦ In the broadest sense, it is the study of how differences in an input can affect the resultant difference at the output.

◦ To break the full 16 rounds, differential cryptanalysis requires 247 chosen plaintexts.

◦ It has been revealed that the designers of DES already knew about this type of attack and designed S-boxes and chose 16 as the number of rounds to make DES specifically resistant to this type of attack.

 Linear cryptanalysis

◦ Linear cryptanalysis is newer than differential cryptanalysis.

◦ Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, "ciphertext" bits, and subkey bits.

◦ Linear cryptanalysis is a known plaintext attack and uses a linear approximation to describe the behavior of the block cipher. Given sufficient pairs of plaintext and corresponding ciphertext, bits of information about the key can be obtained and increased amounts of data will usually give a higher probability of success.

◦ DES is more vulnerable to linear cryptanalysis than to differential cryptanalysis. S-boxes are not very resistant to linear cryptanalysis.

- It has been shown that DES can be broken using 243 pairs of known plaintexts. However, from the practical point of view, finding so many pairs is very unlikely.
- Currently DES is no longer certified for US federal use.
- The availability of faster hardware, and access to large distributed systems meant that 56-bit DES keys could be recovered by brute force searches in an unreasonably short time (days or even hours).
- DES should almost certainly not be used in any new product, and should not be used in existing products to protect information with a lifetime of more than a few minutes.
- Triple-DES is a block cipher, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block.
- DES used a single 56-bit key.
- 3DES uses three 56-bit keys (often just referred to as a 3DES key), and performs three rounds of DES operations on the data.
- The result is that DES technology could be used until long term solution (the Advanced Encryption Standard) is found.

- **A typical application of 3DES is known as EDE (Encrypt-Decrypt-Encrypt).**
- In this case, the first and third keys are equal, so the effective key length is 112-bits.
- In the first operation, the plaintext is encrypted with the first DES key, K1.
- In the second step, the results of the first step, C1, is decrypted using the second key, K2
- Since K2 ≠ K1, this does not result in the original plaintext message.

- In the final step, the results of the second step, C2, is encrypted using the third key, K3
- The output ciphertext C3 is the final encrypted message.
- Recall that K3 = K1 in this case, so even though there are three 56-bit keys, the effective key length is only 112-bits.
- Decryption in this case follows the reverse of the encryption process, as shown below.
- Although the length of the key has doubled, there are 256 (= 72,057,594,037,927,936) times as many keys.
- Therefore a brute force search for a 3DES-EDE key would take 256 times longer on the same hardware than a brute force search for a DES key.
- There are some approaches that can recover 3DES keys more quickly than brute force searches, but for many kinds of data 3DES is still an acceptable encryption method.
- (a) Triple encryption using DES. (b) Decryption.
- **Rules for AES proposals**

1. The algorithm must be a symmetric block cipher.
2. The full design must be public.
3. Key lengths of 128, 192, and 256 bits supported.
4. Both software and hardware implementations required
5. The algorithm must be public or licensed on

- An outline of Rijndael.
- Creating of the state and rk arrays.
- DES is near end of useful life
- NIST has begun process to look for successor to DES
- The Advanced Encryption Standard (AES) was the result of an open international search organized by NIST for a replacement for DES.
- AES Process:
- Proposals submitted 3/98

 AES Workshop - 8/98
 15 proposals selected
 Key sizes of 128, 192, and 256 bits

 Rules:
- Unclassified
- Royalty-free
- Worldwide
- Public domain
- Significantly More Efficient than 3DES
- Symmetric Block Cipher

 AES Timeline:
- Public comment through April 1999
- Candidate Conference, March 22-23, 1999, Rome
- Finalists selected summer 1999
- AES3 conference, April 13 -14, 2000, New York

chapter-3rd 58

 Algorithms were submitted, and five finalists were selected.
- Finalists for the AES standard are:

 MARS (IBM - USA)
 RC6 (RSA Labs - USA)
 Rijndael (Daemen and Rijmen - Belgium)
 SERPENT (Anderson, Biham, and Knudsen - UK, Israel, Norway)
 TWOFISH (Schneier, Kelsey, et al. - USA)

 The finalists were subjected to open review by the cryptographic community.
 The entire process took over 3 years to complete.
 The Rijndael algorithm was declared by NIST to be the eventual winner, and is now generally referred to as AES.
- http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

 Rijndael was developed and submitted by two European cryptographers, Joan Daemen and Vincent Rijmen.

 Security
 Computational Efficiency
 Memory Requirements
 Hardware and Software Suitability
 Flexibility

 The strengths of modern symmetric key encryption algorithms include:
- Fairly fast encryption/decryption process (in comparison to public key techniques, for example)
- Several well known, well tested algorithms are available, including 3DES and AES.
- Library implementations of symmetric key algorithms are commonly available for many programming languages.

KeyExpansion—round keys are derived from the cipher key using Rijndael's key schedule
 Initial Round

- AddRoundKey—each byte of the state is combined with the round key using bitwise xor

Rounds

- SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
- ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
- MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
- **AddRoundKey**

Final Round (no MixColumns)

- SubBytes
- ShiftRows
- AddRoundKey

Input is a 128 bit block (16 bytes) that is placed in the state array

The key is entered in a block and divided into key schedule words of 4 bytes/word.

The key schedule is an expansion of the key—eg, a 128 bit key is expanded into 44 key schedule words.

A square matrix of bytes is used by the standard to describe the state.

The encryption process executes a round function, Nr times, with the number of rounds (Nr) being dependent on key size.

The round function consists of four transformation stages.

- SubBytes()
- ShiftRows()
- MixColumns()
- AddRoundKey()

The cipher begins with an AddRoundKey().

All rounds then execute each of the transformations except the last round.

The MixColumns( ) transformation is not executed in the final round.

For a 128 bit key, there are 10 rounds.

12 and 14 rounds are used with keys of 192 and 256.

Public key In fracture

- RFC 2822 (Internet Security Glossary) defines public-key infrastructure (PKI) as the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography.
- The principal objective for developing a PKI is to enable secure, convenient, and efficient acquisition of public keys. The Internet Engineering Task Force (IETF) Public Key Infrastructure X.509 (PKIX) working group has been the driving force behind setting up a formal (and generic) model based on X.509 that is suitable for deploying a certificate-based architecture on the Internet. This section describes the PKIX model.
- Figure 14.7 shows the interrelationship among the key elements of the PKIX model. These elements are

Digital certificates ?

- In order to bind public keys with their associated user (owner of the private key), PKIs use digital certificates. Digital certificates are the credentials that facilitate the verification of identities between users in a transaction. Much as a passport certifies one's identity as a citizen of a country, the digital certificate establishes the identity of users within the ecosystem. Because digital certificates are used to identify the users to whom encrypted data is sent, or to verify the identity of the signer of information, protecting the authenticity and integrity of the certificate is imperative to maintain the trustworthiness of the system.

PKIX Architectural Model

- End entity: A generic term used to denote end users, devices (e.g., servers, routers), or any other entity that can be identified in the subject field of a public key certificate. End entities typically consume and/or support PKI-related services.
- Certification authority (CA): The issuer of certificates and (usually) certificate revocation lists (CRLs). It may also support a variety of administrative functions, although these are often delegated to one or more Registration Authorities.
- Registration authority (RA): CA may use a third-party Registration Authority (RA) to perform the necessary checks on the person or company requesting the certificate to confirm their identity. The RA may appear to the client as a CA, but they do not actually sign the certificate that is issued.
- CRL issuer: a certificate revocation list (or CRL) is "a list of digital certificates that have been revoked by the issuing certificate authority (CA) before their scheduled expiration date and should no longer be trusted".

- Repository: A generic term used to denote any method for storing certificates and CRLs so that they can be retrieved by End Entities

PKIX Management Functions

- PKIX identifies a number of management functions that potentially need to be supported by management protocols. These are indicated in and include the following:
- Registration: This is the process whereby a user first makes itself known to a CA (directly, or through an RA), prior to that CA issuing a certificate or certificates for that user. Registration begins the process of enrolling in a PKI. Registration usually involves some offline or online procedure for mutual authentication. Typically, the end entity is issued one or more shared secret keys used for subsequent authentication.
- Initialization: Before a client system can operate securely, it is necessary to install key materials that have the appropriate relationship with keys stored elsewhere in the infrastructure. For

example, the client needs to be securely initialized with the public key and other assured information of the trusted CA(s), to be used in validating certificate paths.

- Certification: This is the process in which a CA issues a certificate for a user's public key, and returns that certificate to the user's client system and/or posts that certificate in a repository.

- Key pair recovery: Key pairs can be used to support digital signature creation and verification, encryption and decryption, or both. When a key pair is used for encryption/decryption, it is important to provide a mechanism to recover the necessary decryption keys when normal access to the keying material is no longer possible, otherwise it will not be possible to recover the encrypted data. Loss of access to the decryption key can result from forgotten passwords/PINs, corrupted disk drives, damage to hardware tokens, and so on. Key pair recovery allows end entities to restore their encryption/decryption key pair from an authorized key backup facility (typically, the CA that issued the End Entity's certificate).
- Key pair update: All key pairs need to be updated regularly (i.e., replaced with a new key pair) and new certificates issued. Update is required when the certificate lifetime expires and as a result of certificate revocation.
- Revocation request: An authorized person advises a CA of an abnormal situation requiring certificate revocation. Reasons for revocation include private key compromise, change in affiliation, and name change.
- Cross certification: Two CAs exchange information used in establishing a cross- certificate. A cross-certificate is a certificate issued by one CA to another CA that contains a CA signature key used for issuing certificates.

PKI Management Protocols

- The PKIX working group has defines two alternative management protocols between PKI entities that support the management functions listed in the preceding subsection. RFC 2510 defines the certificate management protocols (CMP). Within CMP, each of the management functions is explicitly identified by specific protocol exchanges. CMP is designed to be a flexible protocol able to accommodate a variety of technical, operational, and business models.
- RFC 2797 defines certificate management messages over CMS (CMC), where CMS refers to RFC 2630, cryptographic message syntax. CMC is built on earlier work and is intended to leverage existing implementations. Although all of the PKIX functions are supported, the functions do not all map into specific protocol exchanges.

Transport Layers

SSL(Secure Socket Layer):

is designed to make use of TCP to provide a reliable end-to-end secure service. SSL is not a single protocol but rather two layers of protocols, as illustrated in figure.

The SSL Record Protocol provides basic security services to various higher layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are defined as part of SSL: the Handshake Protocol, The Change Cipher Spec Protocol, and the Alert Protocol. These SSL specific protocols are used in the management of SSL exchanges and are examined later in this section.

Two important SSL concepts are the SSL session and the SSL connection, which are defined in the specification as follows.

- Connection: A connection is a transport (in the OSI layering model definition)

that provides a suitable type of service. For SSL, such connections are

peer-to-peer relationships. The connections are transient. Every connection is associated with one session.

- Session: An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic

Architecture of SSL

The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment.

Received data are decrypted, verified, decompressed, and reassembled before being delivered to higher-level users. The SSL Record protocol is responsible for ensuring data security through encryption, and data integrity.

The most complex part of SSL is the Handshake Protocol. This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The Handshake Protocol is used before any application data is transmitted.

The Alert Protocol is used to convey SSL-related alerts to the peer entity. As with other applications that use SSL, alert messages are compressed and encrypted, as specified by the current state.

The Change Cipher Spec Protocol is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest. This protocol consists of a single message , which consists of a single byte with the value 1.The sole purpose of this
message is to cause the pending state to be copied into the current state, which
updates the cipher suite to be used on this connection.

The Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL.

HTTPS

HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server. The HTTPS capability is built into all modern Web browsers. Its use depends on the Web server supporting HTTPS communication. The principal difference seen by a user of a Web browser is that URL (uniform resource locator) addresses begin with https:// rather than http://. A normal HTTP connection uses port 80. If HTTPS is specified, port 443 is used, which invokes SSL.

When HTTPS is used, the following elements of the communication are encrypted:

- URL of the requested document
- Contents of the document
- Contents of browser forms (filled in by browser user)
- Cookies sent from browser to server and from server to browser
- Contents of HTTP header

HTTPS is documented in RFC 2818, HTTP Over TLS. There is no fundamental change in using HTTP over either SSL or TLS, and both implementations are referred to as HTTPS.

Processes in HTTPS

Connection Initiation

For HTTPS, the agent acting as the HTTP client also acts as the TLS client. The client initiates a connection to the server on the appropriate port and then sends the TLS ClientHello to begin the TLS handshake. When the TLS handshake has finished, the client may then initiate the first HTTP

request. All HTTP data is to be sent as TLS application data. Normal HTTP behavior, including retained connections, should be followed.

Connection Closure

An HTTP client or server can indicate the closing of a connection by including the following line in an HTTP record: Connection: close. This indicates that the connection will be closed after this record is delivered. The closure of an HTTPS connection requires that TLS close the connection with the peer TLS entity on the remote side, which will involve closing the underlying TCP connection. At the TLS level, the proper way to close a connection is foreach side to use the TLS alert protocol to send a close_notify alert.

Secure Shell (SSH)

Secure Shell (SSH) is a protocol for secure network communications designed to be relatively simple and inexpensive to implement.

The initial version, SSH1 was focused

on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security. SSH also provides a more general

client/server capability and can be used for such network functions as file transfer and e-mail.

A new version, SSH2, fixes a number of security flaws in the original scheme. SSH2 is documented as a proposed standard in IETF RFCs 4250 through 4256.

SSH client and server applications are widely available for most operating systems. It has become the method of choice for remote login and X tunneling and is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems.

- **Transport Layer Protocol:** Provides server authentication, data confidentiality, and data integrity with forward secrecy (i.e., if a key is compromised during one session, the knowledge does not affect the security of earlier sessions).The transport layer may optionally provide compression.

User Authentication Protocol: Authenticates the user to the server.

- • Connection Protocol: Multiplexes multiple logical communications channels over a single, underlying SSH connection.