UNIT- I

INTRODUCTION

Computer data often travels from one computer to another, leaving the safety of its protected physical surroundings. Once the data is out of hand, people with bad intention could modify or forge your data, either for amusement or for their own benefit. Cryptography can reformat and transform our data, making it safer on its trip between computers. The technology is based on the essentials of secret codes, augmented by modern mathematics that protects our data in powerful ways.

• **Computer Security** - generic name for the collection of tools designed to protect data and to thwart hackers

• **Network Security** - measures to protect data during their transmission

• **Internet Security** - measures to protect data during their transmission over a collection of interconnected networks

Basic Concepts

Cryptography The art or science encompassing the principles and methods of transforming an intelligible message into one that is unintelligible, and then retransforming that message back to its

original form

Plaintext The original intelligible message

Cipher text The transformed message

Cipher An algorithm for transforming an intelligible message into one that is unintelligible by transposition and/or substitution methods

Key Some critical information used by the cipher, known only to the sender& receiver

Encipher (encode) The process of converting plaintext to cipher text using a cipher and a key

Decipher (decode) the process of converting cipher text back into plaintext using a cipher and a key

Cryptanalysis The study of principles and methods of transforming an unintelligible message back into an intelligible message without knowledge of the key. Also called code breaking

Cryptology Both cryptography and cryptanalysis

Code An algorithm for transforming an intelligible message into an unintelligible one using a code-book

Cryptography

Cryptographic systems are generally classified along 3 independent dimensions:

Type of operations used for transforming plain text to cipher text

All the encryption algorithms are based on two general principles: **substitution, in which each** element in the plaintext is mapped into another element, and **transposition, in which**elements in the plaintext are rearranged.

The number of keys used

If the sender and receiver uses same key then it is said to be **symmetric key (or)single key (or) conventional encryption.**

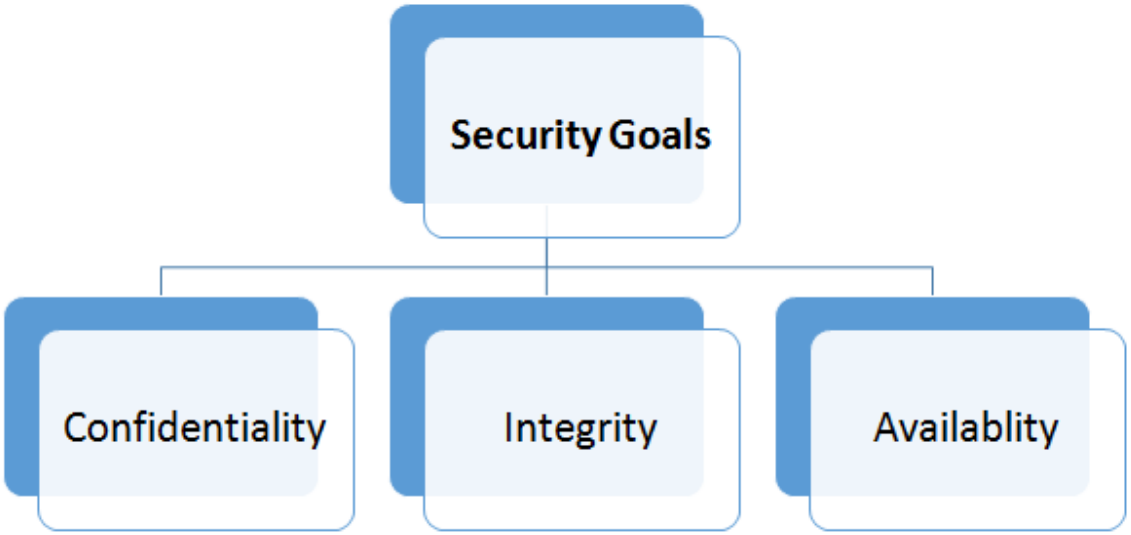
If the sender and receiver use different keys then it is said to be **public key encryption.**

The way in which the plain text is processed

A **block cipher** processes the input and block of elements at a time, producing output block for each input block.

A **stream cipher** processes the input elements continuously, producing output element one at a time, as it goes along.

Basic Principles Security Goals:



The primary goal of network security are Confidentiality, Integrity, and Availability. These three pillars of Network Security are often represented as **CIA triangle**.

- **Confidentiality** − The function of confidentiality is to protect precious business data from unauthorized persons. Confidentiality part of network security makes sure that the data is available only to the intended and authorized persons.
- **Integrity** − This goal means maintaining and assuring the accuracy and consistency of data. The function of integrity is to make sure that the data is reliable and is not changed by unauthorized persons.

- **Availability** − The function of availability in Network Security is to make sure that the data, network resources/services are continuously available to the legitimate users, whenever they require it.

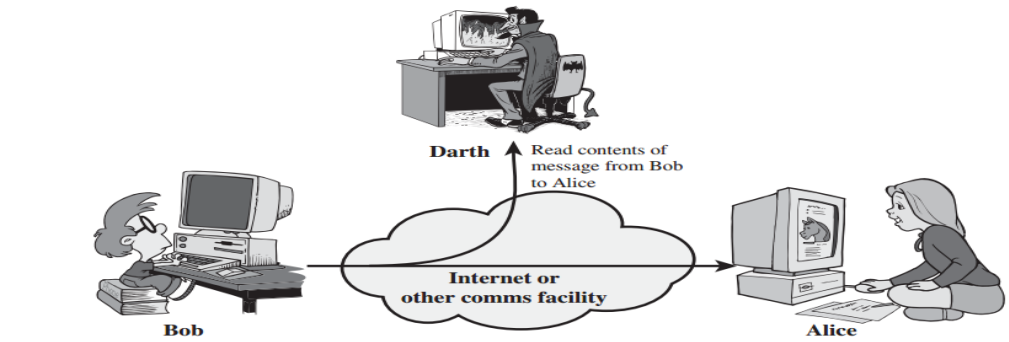
Cryptographic Attacks:

Passive Attacks

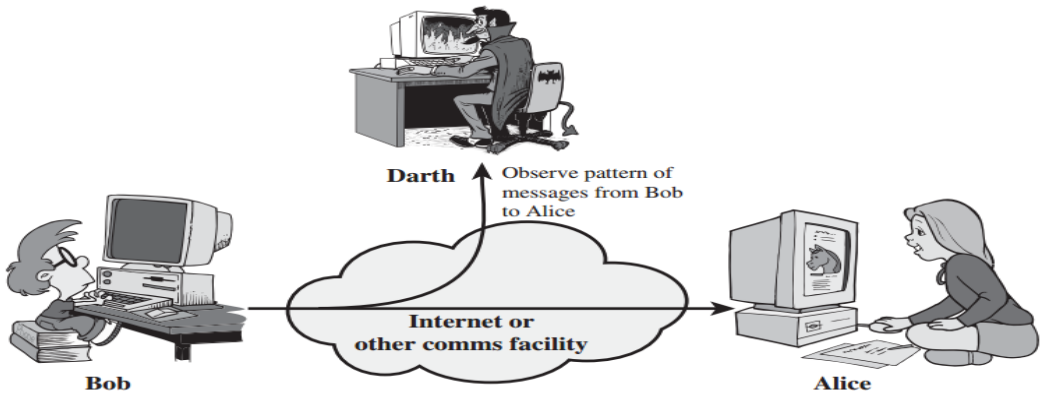
These attacks are not very dangerous as they do not cause any modification to the data. These attacks are generally done to secretly listen and monitor the communication of other parties. Passive attacks are very difficult to detect because these attacks do not change the information of the data.
There are two types of passive attacks in cryptography and network security: Traffic analysis and Release of Message content

- **Traffic Analysis:** In this attack, an attacker tries to predict the nature of communication by using information. The information such as analyzing traffic, identify communication hosts, and frequency of messages.

- **Release of Message content:** It is similar to hearing a telephone conversation between two users. In this attack, the attacker can monitor the content of the transmitted data such as email messages, etc.



(a) Release of message contents



Active attacks

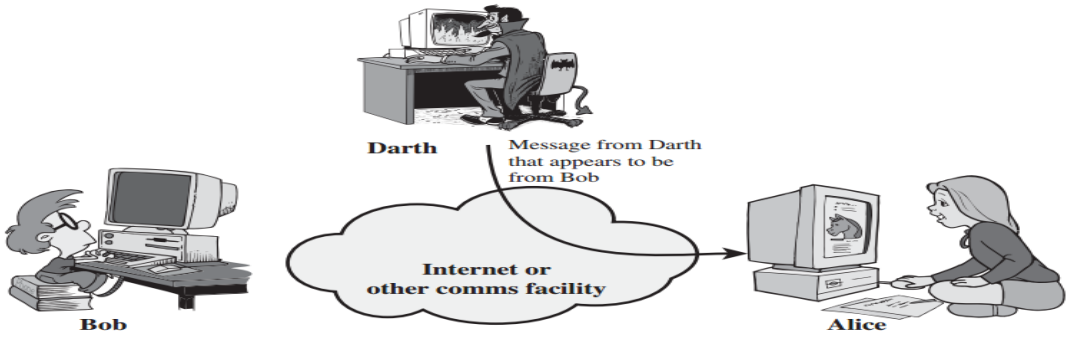
These attacks involve some modification of the data stream or the creation of a false stream. These attacks can be classified in to four categories:

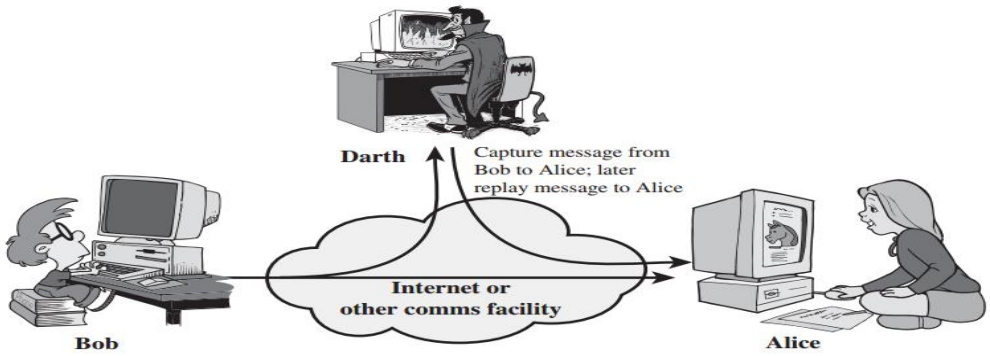
Masquerade – One entity pretends to be a different entity.

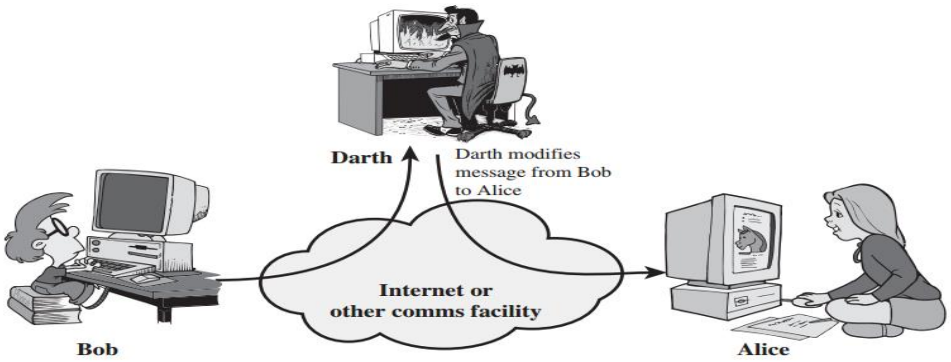
Replay – involves passive capture of a data unit and its subsequent transmission to produce an unauthorized effect.

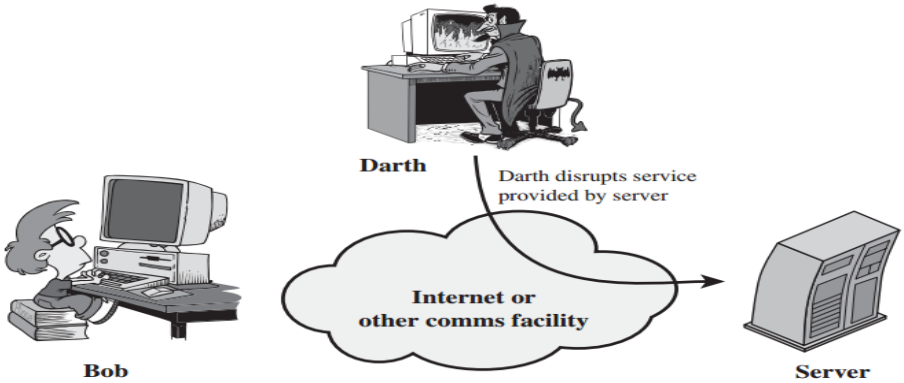
Modification of messages – Some portion of message is altered or the messages are delayed or recorded, to produce an unauthorized effect.

Denial of service – Prevents or inhibits the normal use or management of communication facilities. Another form of service denial is the disruption of an entire network, either by disabling the network or overloading it with messages so as to degrade performance. It is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communication facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them.


Darth
Message from Darth that appears to be from Bob
Internet or other comms facility
Bob
Alice


Darth
Capture message from Bob to Alice; later replay message to Alice
Internet or other comms facility
Bob
Alice


Darth
Darth modifies message from Bob to Alice
Internet or other comms facility
Bob
Alice


Darth
Darth disrupts service provided by server
Internet or other comms facility
Bob
Server

SECURITY ATTACKS

There are four general categories of attack which are listed below.

Interruption

An asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability e.g., destruction of piece of hardware, cutting of a communication line or Disabling of file management system.

Interception

An unauthorized party gains access to an asset. This is an attack on confidentiality. Unauthorized party could be a person, a program or a computer. e.g., wire tapping to capture data in the network, illicit copying of files.

Modification

An unauthorized party not only gains access to but tampers with an asset. This is an attack on integrity. e.g., changing values in data file, altering a program, modifying the contents of messages being transmitted in a network.

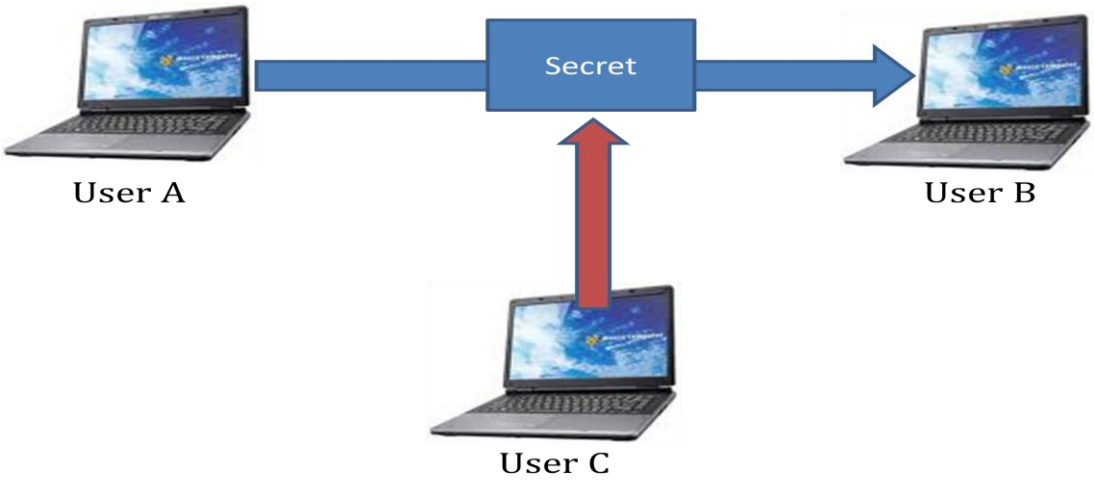
Fabrication

An unauthorized party inserts counterfeit objects into the system. This is an attack on authenticity. e.g., insertion of spurious message in a network or addition of records to a file.

Services and Mechanisms:

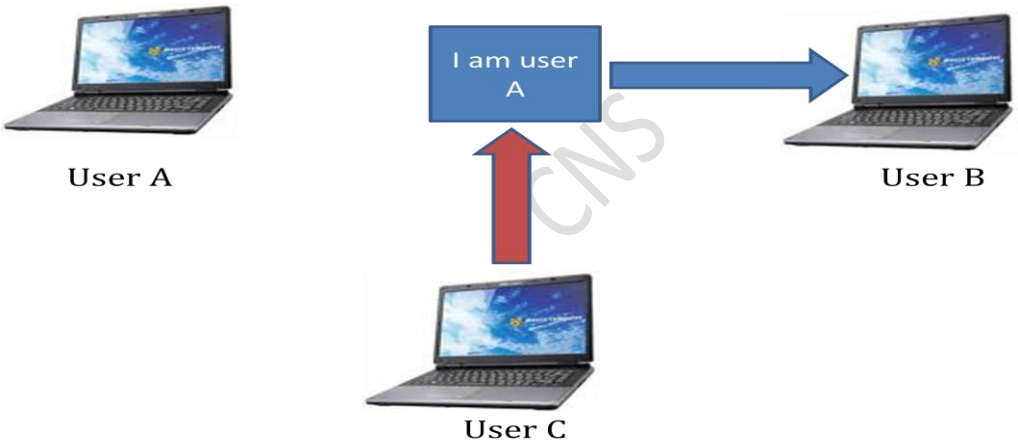
- Confidentiality
- Authentication
- Integrity
- Non-repudiation
- Access control
- availability

Confidentiality: refers to protecting information from being accessed by unauthorized parties. In other words, only the people who are authorized to do so can gain access to sensitive data



Note: Interception causes loss of message confidentiality

Authentication: Authentication is the process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials.



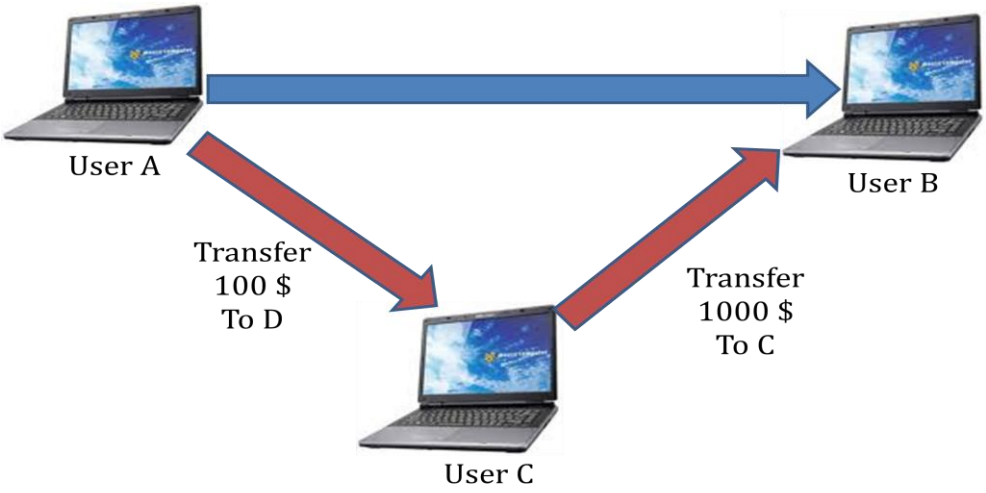
Note: Fabrication is possible in absence of authentication

Integrity:

1.Prevent unauthorized users from making modifications to data or programs.

2. Prevent authorized users from making improper or unauthorized modifications.

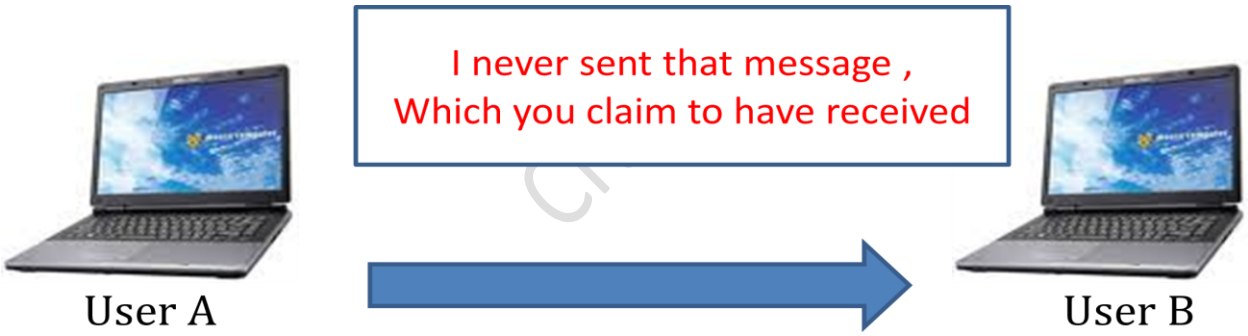
3.Maintain internal and external consistency of data and programs.



Note: Modification causes loss of message integrity

Non repudiation:

Non repudiation does not allow the sender of a message to refute the claim of not sending that message.



Access control

- Access control specifies and controls who can access what
- An access control mechanism controls which clients or applications have access to the server. Only properly authorized clients can connect to the server.
- Two different types of access control mechanisms are used: user based and host based.

Availability : means that information is accessible to authorized users. It is basically an assurance that your system and data are accessible by authorized users whenever it's needed.

User A

User B

User C

Note: Interruption puts the availability of resources in danger

Security mechanisms:

Table 1.3 Security Mechanisms (X.800)

SPECIFIC SECURITY MECHANISMS	PERVASIVE SECURITY MECHANISMS
May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.	Mechanisms that are not specific to any particular OSI security service or protocol layer.
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.	**Trusted Functionality** That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).
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).	**Security Label** The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.
Access Control A variety of mechanisms that enforce access rights to resources.	**Event Detection** Detection of security-relevant events.
Data Integrity A variety of mechanisms used to assure the integrity of a data unit or stream of data units.	**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.
Authentication Exchange A mechanism intended to ensure the identity of an entity by means of information exchange.	**Security Recovery** Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.
Traffic Padding The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.	
Routing Control Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.	
Notarization The use of a trusted third party to assure certain properties of a data exchange.	

Symmetric and public key algorithms

Encryption/Decryption methods fall into two categories.

Symmetric key

Public key

In symmetric key algorithms, the encryption and decryption keys are known both to sender and receiver. The encryption key is shared and the decryption key is easily calculated from it. In many cases, the encryption and decryption keys are the same.

In public key cryptography, encryption key is made public, but it is computationally infeasible to find the decryption key without the information known to the receiver.

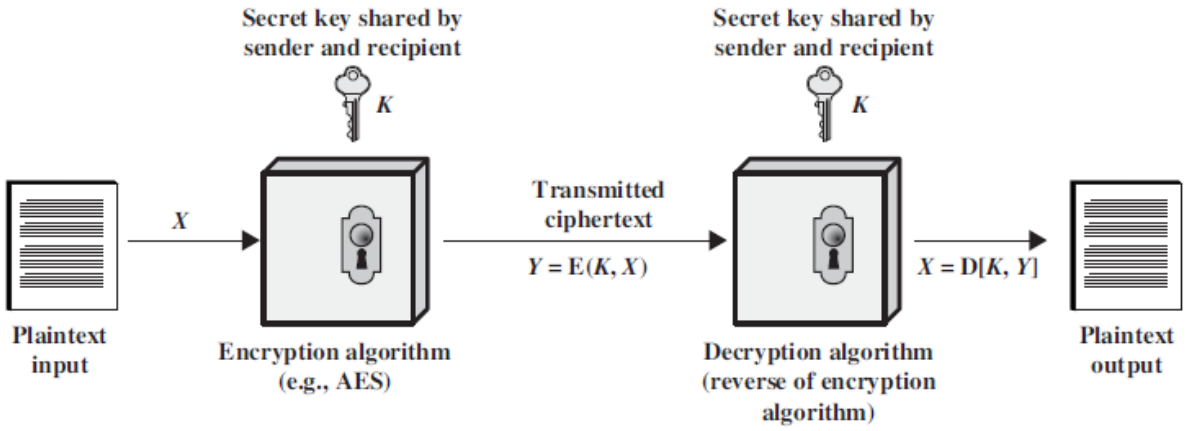
A MODEL FOR NETWORK SECURITY

A message is to be transferred from one party to another across some sort of internet. 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.

CONVENTIONAL ENCRYPTION

• Referred conventional / private-key / single-key

• Sender and recipient share a common key



Here the original message, referred to as plaintext, is converted into apparently random nonsense, referred to as cipher text. The encryption process consists of an algorithm and a key.

The key is a value independent of the plaintext. Changing the key changes the output of the algorithm. Once the cipher text is produced, it may be transmitted. Upon reception, the cipher text can be transformed back to the original plaintext by using a decryption algorithm and the same key that was used for encryption. The security depends on several factors. First, the

encryption algorithm must be powerful enough that it is impractical to decrypt a message on the basis of cipher text alone. Beyond that, the security depends on the secrecy of the key, not the secrecy of the algorithm.

• **Two requirements for secure use of symmetric encryption:**

– A strong encryption algorithm

– A secret key known only to sender / receiver

Y = EK(X)

X = DK(Y)

• **assume encryption algorithm is known**

• **implies a secure channel to distribute key**

CLASSICAL ENCRYPTION TECHNIQUES

There are two basic building blocks of all encryption techniques: substitution and transposition.

SUBSTITUTION TECHNIQUES

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

Caesar cipher (or) shift cipher

The earliest known use of a substitution cipher and the simplest was by Julius Caesar. The Caesar cipher involves replacing each letter of the alphabet with the letter standing 3 places further down the alphabet.

e.g., plain text : pay more money

Cipher text: SDB PRUH PRQHB

Note that the alphabet is wrapped around, so that letter following „z" is „a".

For each plaintext letter p, substitute the cipher text letter c such that

$C = E(p) = (p+3) \bmod 26$

A shift may be any amount, so that general Caesar algorithm is

$C = E(p) = (p+k) \bmod 26$

Where k takes on a value in the range 1 to 25. The decryption algorithm is simply

$P = D(C) = (C-k) \bmod 26$

Playfair cipher

The best known multiple letter encryption cipher is the playfair, which treats digrams in the plaintext as single units and translates these units into cipher text digrams. The playfair algorithm is based on the use of 5x5 matrix of letters constructed using a keyword. Let the keyword be „monarchy". The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of
the matrix with the remaining letters in alphabetical order.

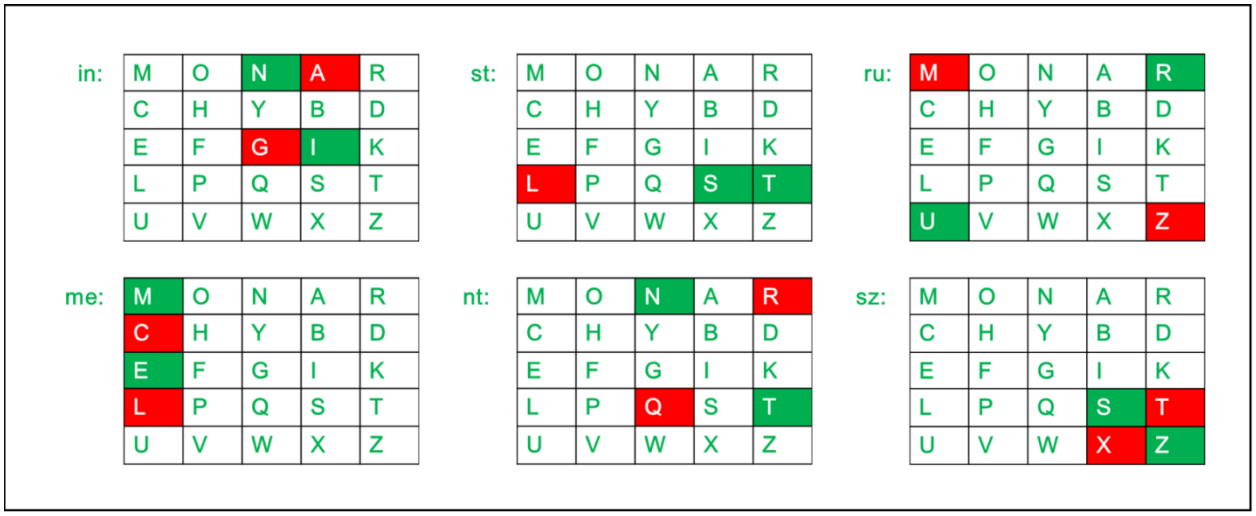
The letter „i" and „j" count as one letter. Plaintext is encrypted two letters at a time
According to the following rules:

Repeating plaintext letters that would fall in the same pair are separated with a
Filler letter such as „x".

Plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row following the last.

Plaintext letters that fall in the same column are replaced by the letter beneath, with the top element of the column following the last.

Otherwise, each plaintext letter is replaced by the letter that lies in its own row
And the column occupied by the other plaintext letter.



Plaintext = meet me at the school house
Splitting two letters as a unit => me et me at th es ch o x ol ho us ex
Corresponding cipher text => CL KL CL RS PD IL HY AV MP HF XL IU

Strength of playfair cipher

Playfair cipher is a great advance over simple mono alphabetic ciphers.
Since there are 26 letters, 26x26 = 676 diagrams are possible, so identification of individual diagram is more difficult.

1.15.1.3 Polyalphabetic ciphers

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

Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message. The general name
for this approach is polyalphabetic cipher. All the techniques have the following features in common.
A set of related monoalphabetic substitution rules are used
A key determines which particular rule is chosen for a given transformation.

Vigenere cipher

In this scheme, the set of related monoalphabetic substitution rules consisting of
26 caesar ciphers with shifts of 0 through 25. Each cipher is denoted by a key letter. e.g., Caesar
cipher with a shift of 3 is denoted by the key value 'd‟ (since a=0, b=1, c=2 and so on). To aid in understanding the scheme, a matrix known as vigenere tableau is
Constructed
Each of the 26 ciphers is laid out horizontally, with the key letter for each cipher to its left. A normal alphabet for the plaintext runs across the top. The process of

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

Encryption is simple: Given a key letter X and a plaintext letter y, the cipher text is at the intersection of the row labeled x and the column labeled y; in this case, the ciphertext is V.

To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword.

e.g., key = d e c e p t i v e d e c e p t i v e d e c e p t i v e PT = w e a r e d i s c o v e r e d s a v e y o u r s e l f CT = ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Decryption is equally simple. The key letter again identifies the row. The position of the cipher text letter in that row determines the column, and the plaintext letter is at the top of that column.Strength of Vigenere cipher o There are multiple cipher text letters for each plaintext letter.o Letter frequency information is obscured.

One Time Pad Cipher

It is an unbreakable cryptosystem. It represents the message as a sequence of 0s and 1s. this can be accomplished by writing all numbers in binary, for example, or by using ASCII. The

key is a random sequence of 0‟s and 1‟s of same length as the message. Once a key is used, it is

discarded and never used again. The system can be expressed as
Follows:
Ci = Pi Ki Ci - ith binary digit of cipher text Pi - ith binary digit of
plaintext Ki - ith binary digit of key
Exclusive OR operation
Thus the cipher text is generated by performing the bitwise XOR of the plaintext and the key. Decryption uses the same key. Because of the properties of XOR, decryption simply involves the
same bitwise operation:
Pi = Ci Ki
e.g., plaintext = 0 0 1 0 1 0 0 1
Key = 1 0 1 0 1 1 0 0
---------- ciphertext = 1 0 0 0 0 1 0 1
Advantage:
Encryption method is completely unbreakable for a ciphertext only attack.
Disadvantages
It requires a very long key which is expensive to produce and expensive to transmit.
Once a key is used, it is dangerous to reuse it for a second message; any knowledge
on the first message would give knowledge of the second.

TRANSPOSITION TECHNIQUES

All the techniques examined so far involve the substitution of a cipher text symbol
for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of
permutation on the plaintext letters. This technique is referred to as a transposition cipher. Rail fence
is simplest of such cipher, in which the plaintext is written down as a sequence of diagonals and
then read off as a sequence of rows.
Plaintext = meet at the school house
To encipher this message with a rail fence of depth 2, we write the message as follows:
m e a t e c o l o s
e t t h s h o h u e
The encrypted message is
MEATECOLOSETTHSHOHUE

Row Transposition Ciphers-

A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of columns then
becomes the key of the algorithm.
e.g., plaintext = meet at the school house
Key = 4 3 1 2 5 6 7
PT = m e e t a t t
h e s c h o o
l h o u s e
CT = ESOTCUEEHMHLAHSTOETO
A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext. The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is more complex permutation that is
not easily reconstructed.

Feistel cipher structure

The input to the encryption algorithm are a plaintext block of length 2w bits and a key K. the plaintext block is divided into two halves L0 and R0. The two halves of the data pass through „n" rounds of processing and then combine to produce the ciphertext block. Each round „i"
has inputs Li-1 and Ri-1, derived from the previous round, as well as the subkey Ki, derived fromthe overall key K. in general, the subkeys Ki are different from K and from each other. All rounds have the same structure. A substitution is performed on the left half of the data (as similar to S-DES). This is done by applying a round function F to the right half of the data and then taking the XOR of the output of that function and the left half of the data. The round function has the same general structure for each round but is parameterized by the round sub key ki.Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data. This structure is a particular form of the substitution-permutation network.

The exact realization of a Feistel network depends on the choice of the following parameters and design features:

Block size - Increasing size improves security, but slows cipher

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

Number of rounds - Increasing number improves security, but slows cipher

Subkey generation - Greater complexity can make analysis harder, but slows cipher

Round function - Greater complexity can make analysis harder, but slows cipher

Fast software en/decryption & ease of analysis - are more recent concerns for practical use and testing.

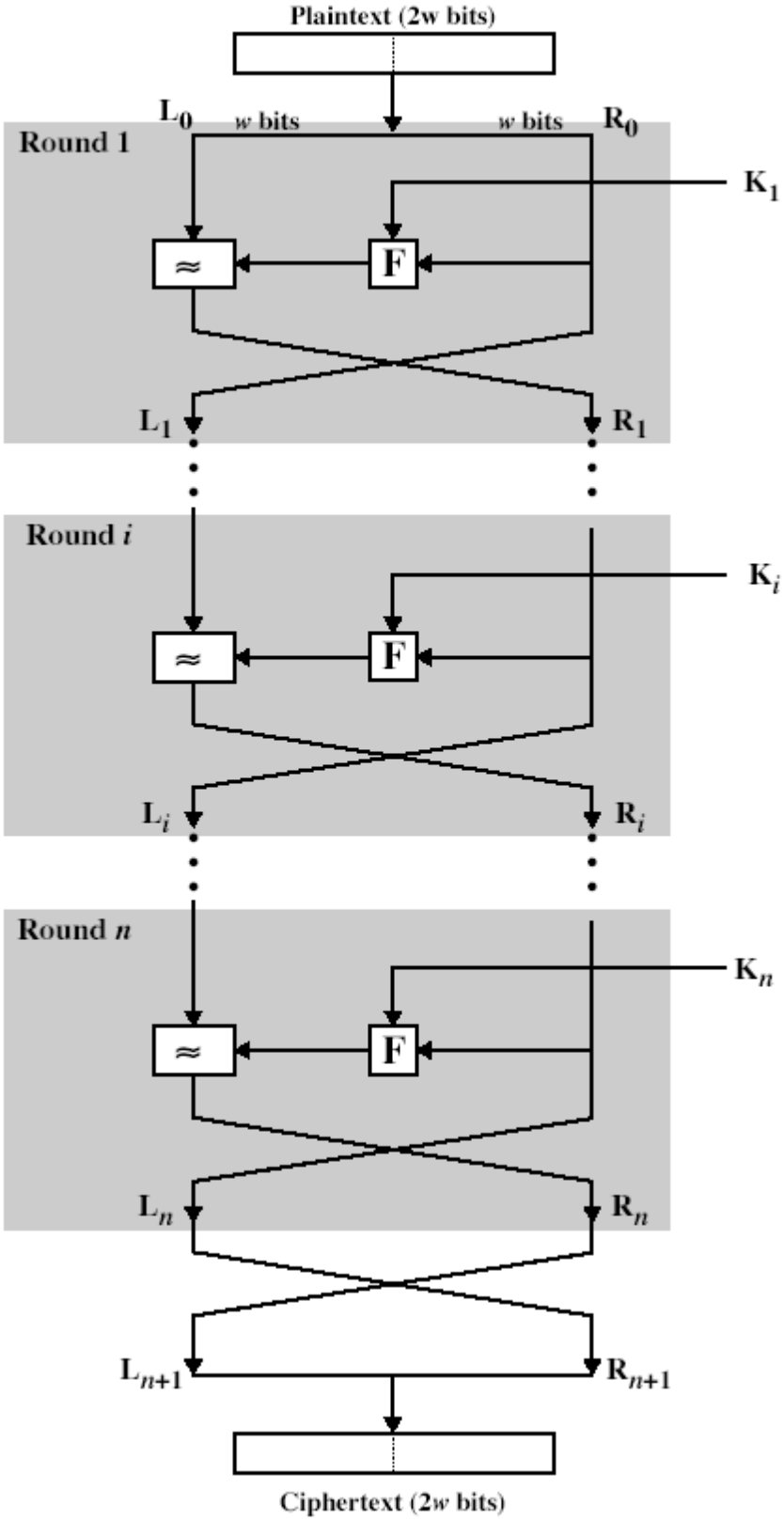


Fig: Classical Feistel Network

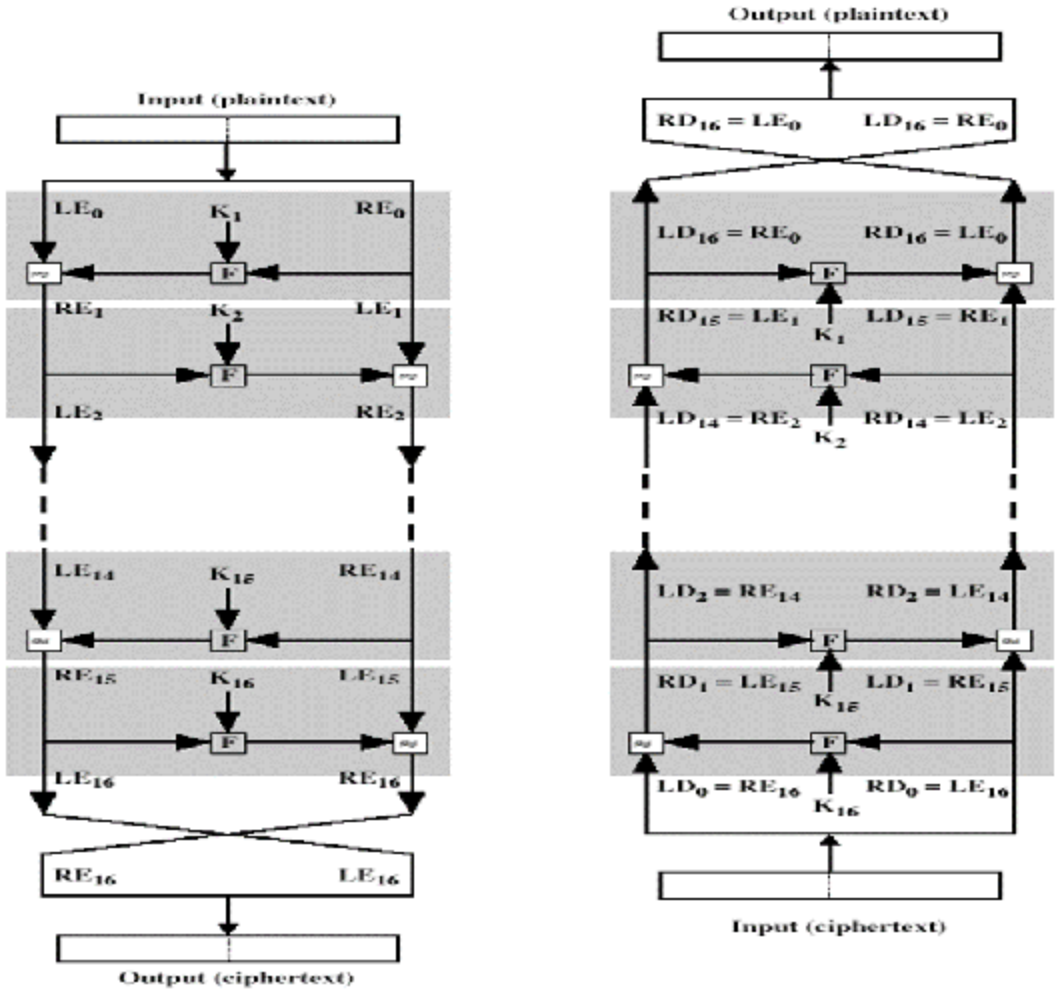


Fig: Feistel encryption and decryption

The process of decryption is essentially the same as the encryption process. The rule is as follows:

use the cipher text as input to the algorithm, but use the subkey ki in reverse order. i.e., kn in the first round, kn-1 in second round and so on. For clarity, we use the notation LEi and REi for

data traveling through the decryption algorithm. The diagram below indicates that, at each round, the intermediate value of the decryption process is same (equal) to the corresponding value

of the encryption process with two halves of the value swapped.

i.e., REi || LEi (or) equivalently RD16-i || LD16-i

After the last iteration of the encryption process, the two halves of the output are swapped, so that the cipher text is RE16 || LE16. The output of that round is the cipher text. Now

take the cipher text and use it as input to the same algorithm. The input to the first round is RE16

|| LE16, which is equal to the 32-bit swap of the output of the sixteenth round of the encryption process.

Now we will see how the output of the first round of the decryption process is equal to a 32-bit swap of the input to the sixteenth round of the encryption process. First consider the encryption process,

LE16 = RE15

RE16 = LE15 F (RE15, K16) On the decryption side,

LD1 =RD0 = LE16 =RE15
RD1 = LD0 F (RD0, K16)
= RE16 F (RE15, K16)
= [LE15 F (RE15, K16)] F (RE15, K16)
= LE15
Therefore, LD1 = RE15
RD1 = LE15 In general, for the ith iteration of the encryption algorithm, LEi = REi-1
REi = LEi-1 F (REi-1, Ki)
Finally, the output of the l

BLOCK CIPHER PRINCIPLES

Virtually, all symmetric block encryption algorithms in current use are based on a structure referred to as Fiestel block cipher. For that reason, it is important to examine the design principles
of the Fiestel cipher. We begin with a **comparison of stream cipher with block cipher.**

• **A stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. E.g, vigenere cipher. **A block cipher** is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length. Typically a block size of 64 or 128 bits is used.

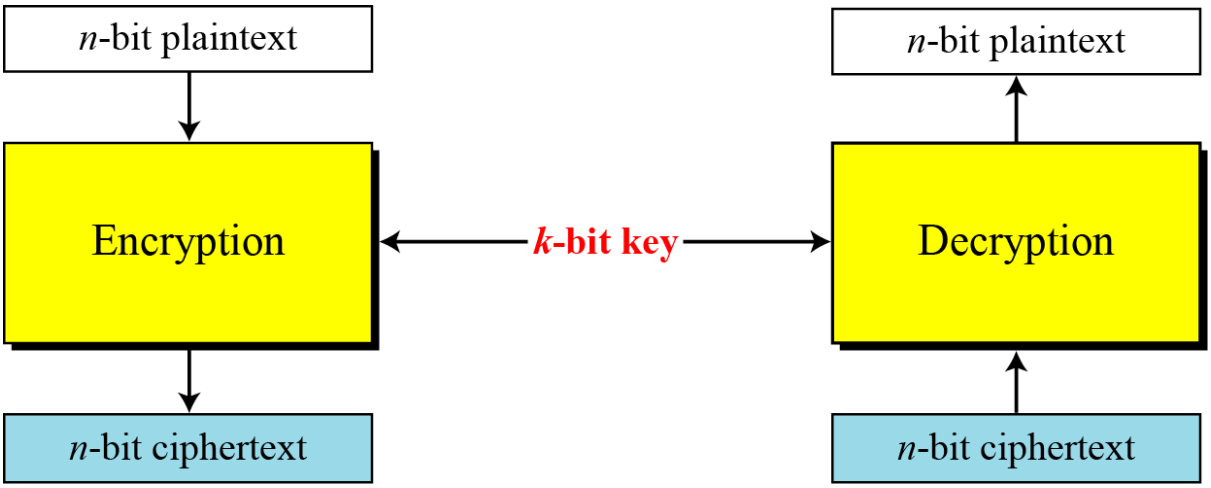
Block cipher principles

• 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
• using idea of a product cipher in 1949 Claude Shannon introduced idea of substitution-permutation (S-P) networks called modern substitution-transposition product cipher
these form the basis of modern block ciphers
• S-P networks are based on the two primitive cryptographic operations we have seen before:
• substitution (S-box)
• permutation (P-box)
• provide confusion and diffusion of message
• **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
• **confusion** – makes relationship between ciphertext and key as complex as possible

UNIT- II

Introduction to Modern Symmetric-key Ciphers

A symmetric-key modern block cipher encrypts an n-bit block of plaintext or decrypts an n-bit block of ciphertext. The encryption or decryption algorithm uses a k-bit key.



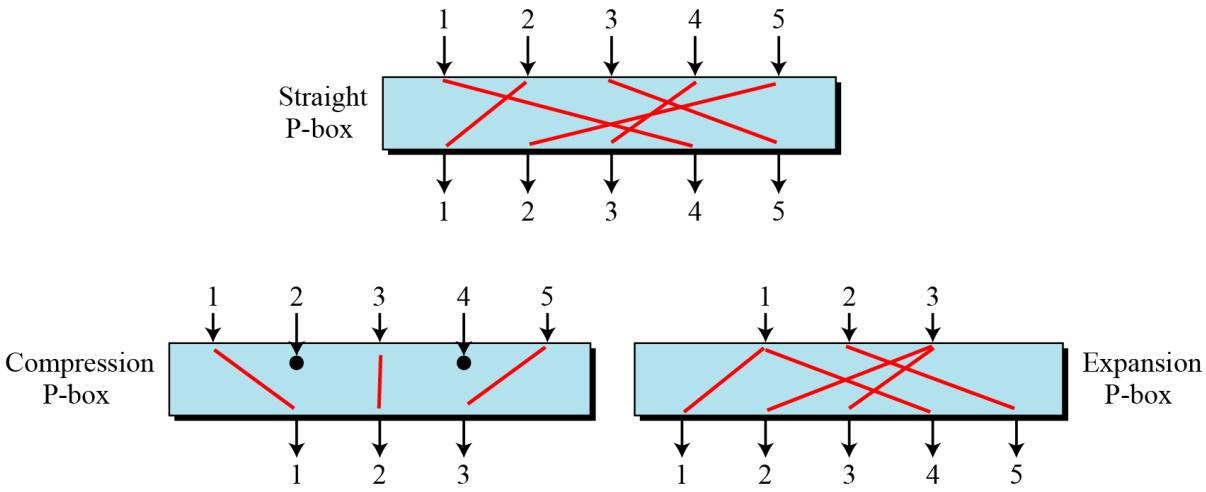
A modern block cipher

Components of a Modern Block Cipher

Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.

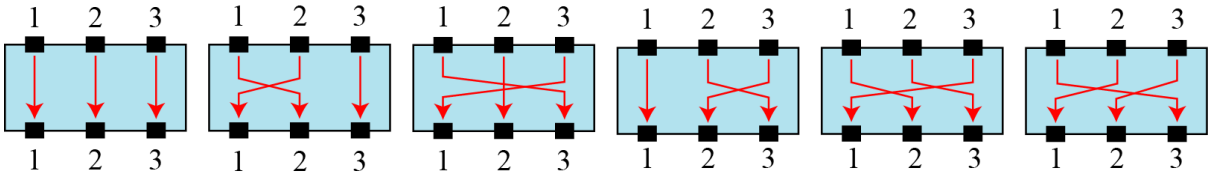
P-Boxes

A P-box (permutation box) parallels the traditional transposition cipher for characters. It transposes bits.



Three types of P-boxes

The possible mappings of a 3 × 3 P-box



58	50	42	34	26	18	10	02	60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06	64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01	59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05	63	55	47	39	31	23	15	07

Compression P-Boxes

A compression P-box is a P-box with n inputs and m outputs where m < n.

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

Attacks on Block Ciphers

Attacks on traditional ciphers can also be used on modern block ciphers, but today's block ciphers resist most of the attacks

Differential Cryptanalysis

Eli Biham and Adi Shamir introduced the idea of differential cryptanalysis. This is a chosen-plaintext attack.

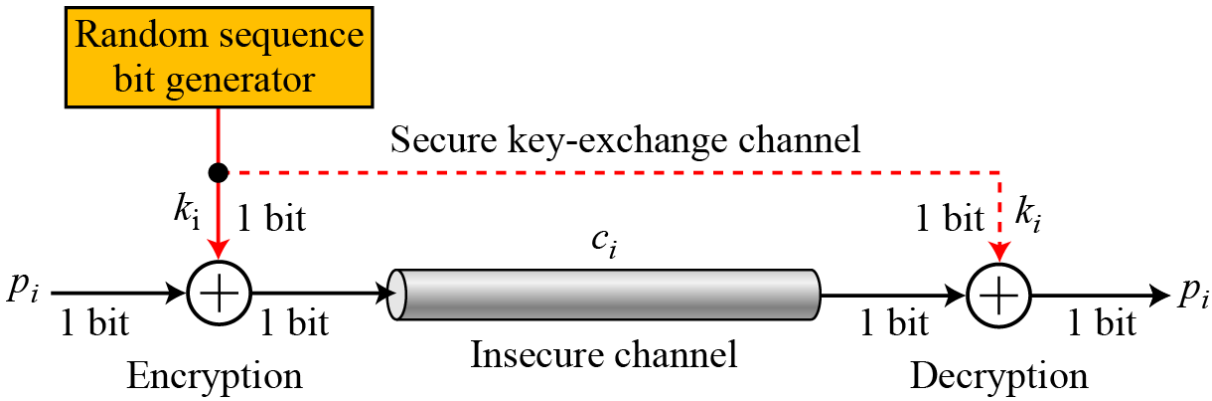
MODERN STREAM CIPHERS

In a modern stream cipher, encryption and decryption

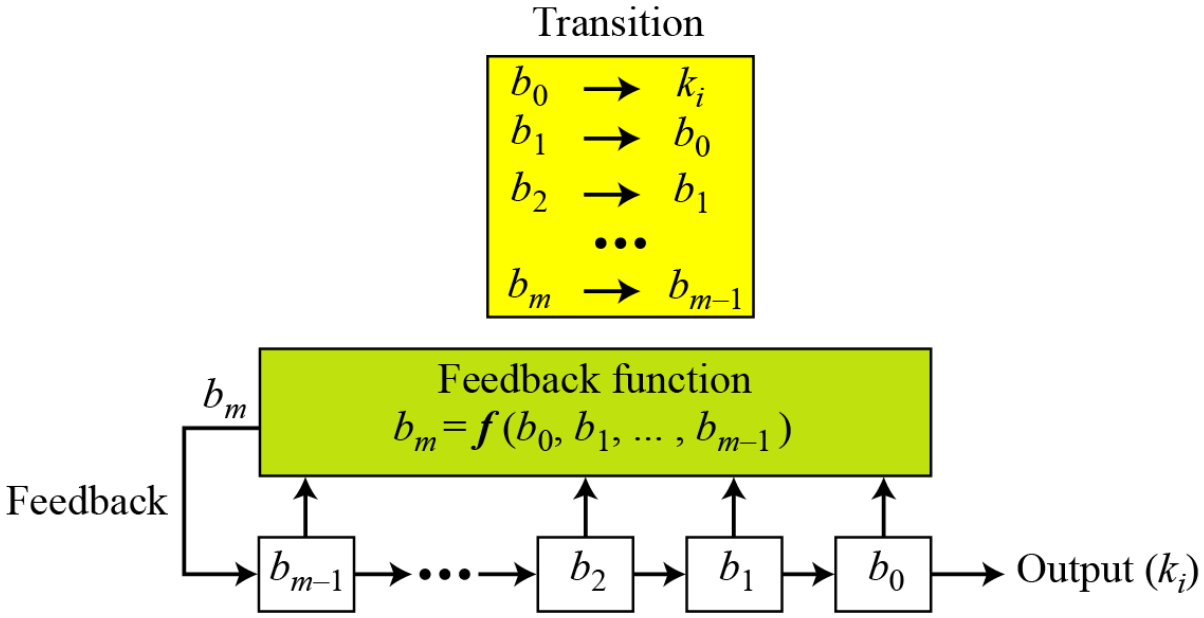
are done r bits at a time. We have a plaintext bit stream P = $p_n \ldots p_2\, p_1$, a ciphertext bit stream
C = $c_n \ldots c_2\, c_1$, and a key bit stream K = $k_n \ldots k_2\, k_1$, in which p_i , c_i , and k_i are r-bit words.

Synchronous Stream Ciphers

In a synchronous stream cipher the key is independent of the plaintext or ciphertext.



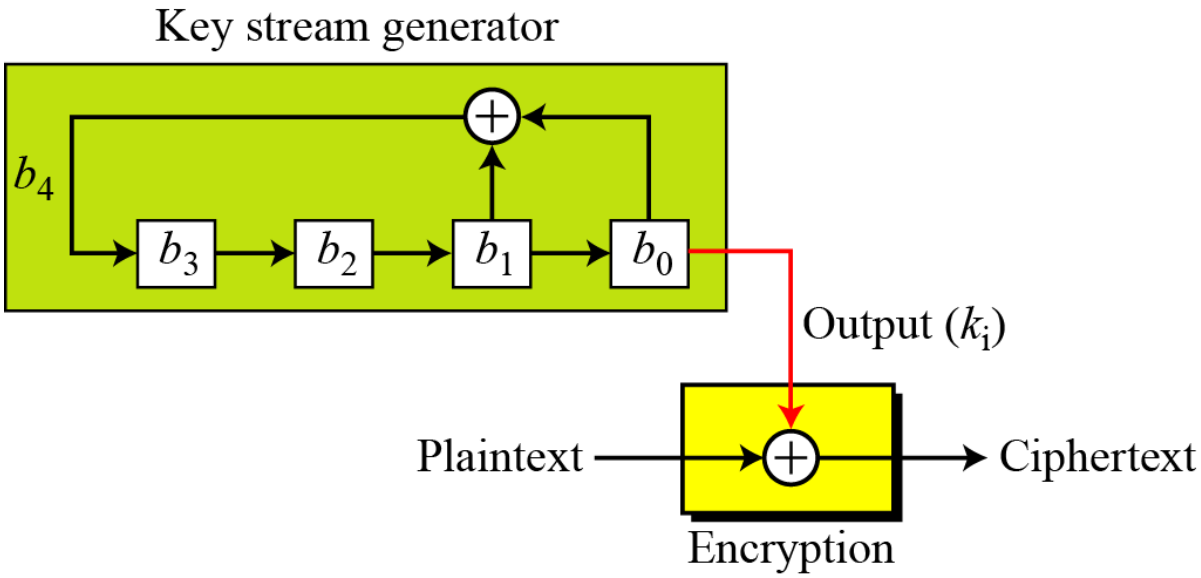
Feedback shift register (FSR)



Example 5.19

Create a linear feedback shift register with 4 cells in which

$b_4 = b_1$ Å b_0. Show the value of output for 20 transitions (shifts) if the seed is $(0001)_2$.



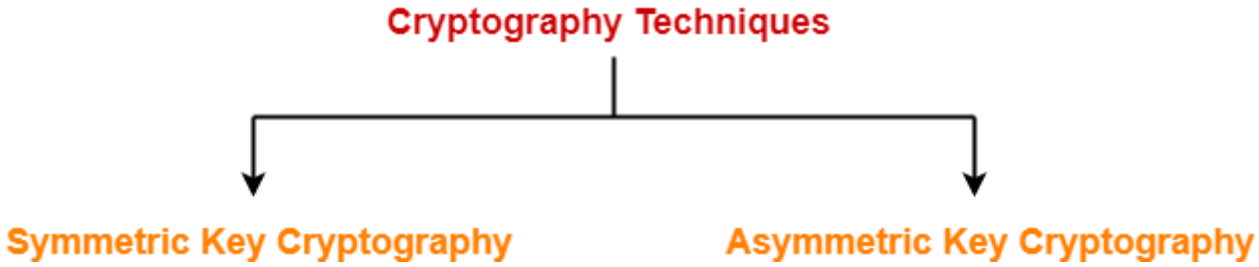
Nonsynchronous Stream Ciphers

In a nonsynchronous stream cipher, the key depends on either the plaintext or ciphertext.

Symmetric Encryption Mathematics of Symmetric Key Cryptography

Cryptography Techniques-

Cryptography techniques may be classified as-



1. Symmetric Key Cryptography
2. Asymmetric Key Cryptography

In this article, we will discuss about symmetric key cryptography.

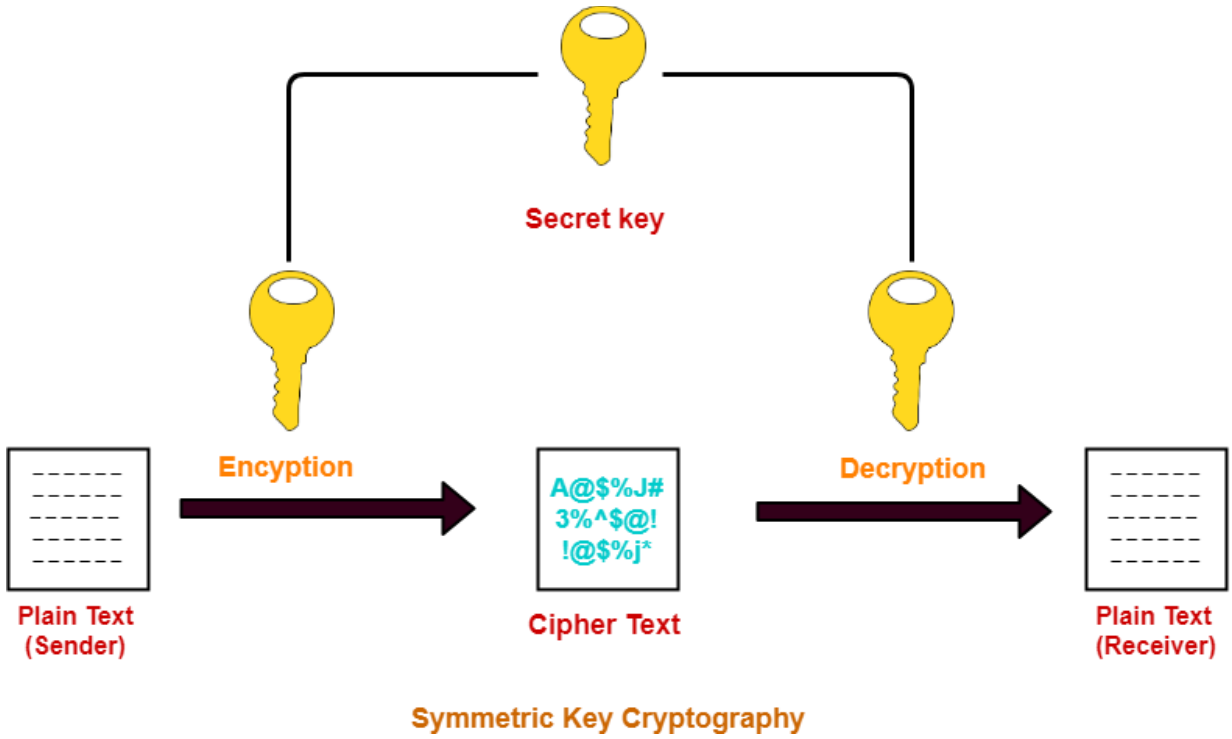
Symmetric Key Cryptography-

In this technique,

- Both sender and receiver uses a common key to encrypt and decrypt the message.
- This secret key is known only to the sender and to the receiver.
- It is also called as **secret key cryptography**.

Working-

The message exchange using symmetric key cryptography involves the following steps-



Symmetric Key Cryptography

- Before starting the communication, sender and receiver shares the secret key.
- This secret key is shared through some external means.
- At sender side, sender encrypts the message using his copy of the key.
- The cipher text is then sent to the receiver over the communication channel.
- At receiver side, receiver decrypts the cipher text using his copy of the key.
- After decryption, the message converts back into readable format.

Symmetric Encryption Algorithms-

Some of the encryption algorithms that use symmetric key are-

- **Advanced Encryption Standard (AES)**
- **Data Encryption Standard (DES)**

Advantages-

The advantages of symmetric key algorithms are-

- **They are efficient.**
- **They take less time to encrypt and decrypt the message.**

Disadvantages-

Point-01:

The number of keys required is very large.

In symmetric key cryptography,

- **Each pair of users require a unique secret key.**
- **If N people in the world wants to use this technique, then there needs to be N(N-1) / 2 secret keys.**
- **For 1 million people to communicate, a half billion secret keys would be needed.**

How N(N-1)/2 Keys Will Be Required?

- **Consider a complete graph with N nodes.**
- **Consider each node represents one person.**
- **Then, each person will require (N-1) keys to communicate with other (N-1) people.**
- **Thus, each edge must have a unique key for communication.**
- **Thus, Number of keys required = Number of edges = $^{n}C_2$ = n(n-1)/2.**

Point-02:

- **Sharing the secret key between the sender and receiver is an important issue.**
- **While sharing the key, attackers might intrude.**

To overcome this disadvantage,

Diffie Hellman Key Exchange Algorithm is used for exchanging the secret key.

Important Points-

Point-01:

In symmetric key cryptography,

- **Both sender and receiver uses the same key.**
- **Sender encrypts the message using his copy of the key.**
- **Receiver decrypts the message using his copy of the key.**
- **The key must not be known to anyone else other than sender and receiver.**
- **If the secret key is known to any intruder, he could decrypt the message.**

Point-02:

- **This cryptography technique is called as symmetric key cryptography.**
- **It is because both sender and receiver use the same key on their sides.**

Point-03:

- **This cryptography technique is called as secret key cryptography.**
- **It is because the key has to be kept secret between the sender and receiver.**

To gain better understanding about Symmetric Key Cryptography,

Introduction to Modern Symmetric Key Ciphers

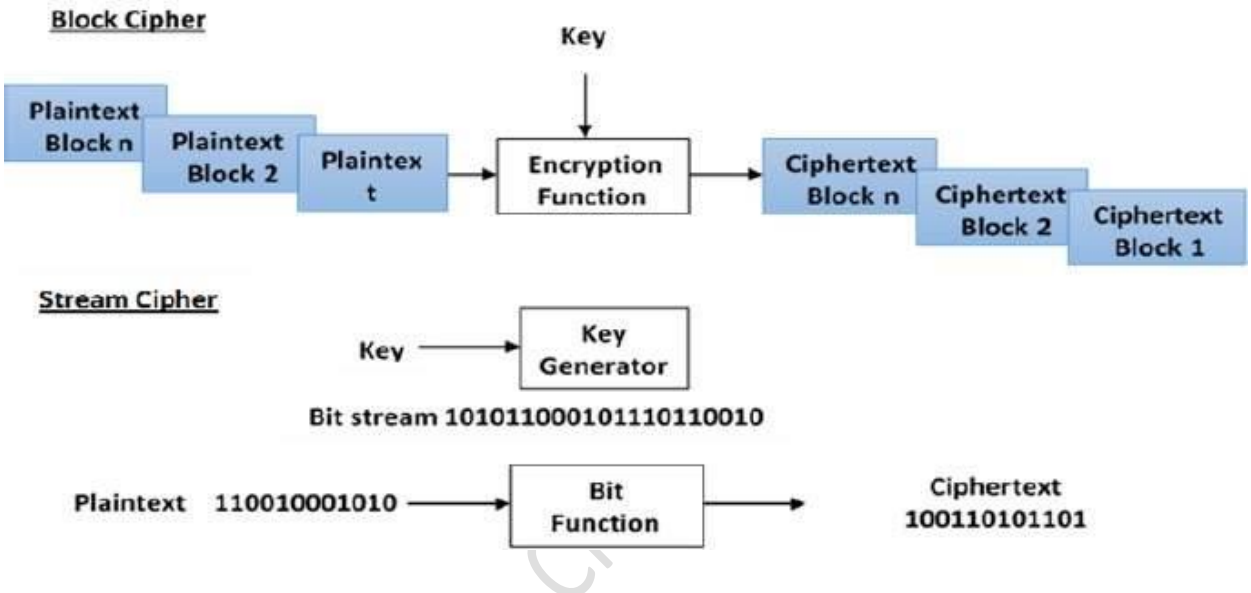
Digital data is represented in strings of binary digits (bits) unlike alphabets. Modern cryptosystems need to process this binary strings to convert in to another binary string. Based on how these binary strings are processed, a symmetric encryption schemes can be classified in to −

Block Ciphers

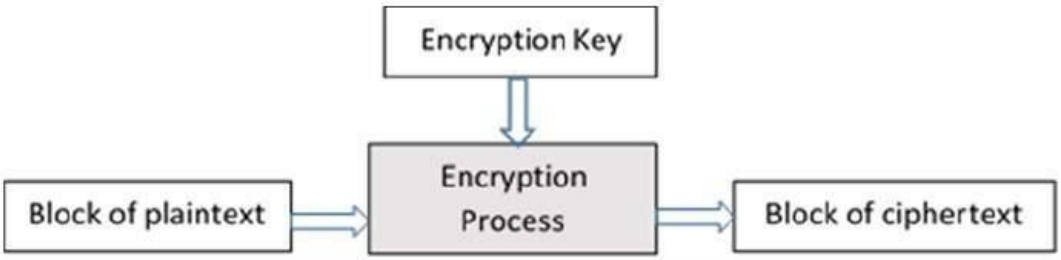
In this scheme, the plain binary text is processed in blocks (groups) of bits at a time; i.e. a block of plaintext bits is selected, a series of operations is performed on this block to generate a block of ciphertext bits. The number of bits in a block is fixed. For example, the schemes DES and AES have block sizes of 64 and 128, respectively.

Stream Ciphers

In this scheme, the plaintext is processed one bit at a time i.e. one bit of plaintext is taken, and a series of operations is performed on it to generate one bit of ciphertext. Technically, stream ciphers are block ciphers with a block size of one bit.



The basic scheme of a block cipher is depicted as follows −



A block cipher takes a block of plaintext bits and generates a block of ciphertext bits, generally of same size. The size of block is fixed in the given scheme. The choice of block size does not directly affect to the strength of encryption scheme. The strength of cipher depends up on the key length.

Block Size

Though any size of block is acceptable, following aspects are borne in mind while selecting a size of a block.

- **Avoid very small block size** − Say a block size is m bits. Then the possible plaintext bits combinations are then 2^m. If the attacker discovers the plain text blocks

corresponding to some previously sent ciphertext blocks, then the attacker can launch a type of 'dictionary attack' by building up a dictionary of plaintext/ciphertext pairs sent using that encryption key. A larger block size makes attack harder as the dictionary needs to be larger.

- **Do not have very large block size** − With very large block size, the cipher becomes inefficient to operate. Such plaintexts will need to be padded before being encrypted.
- **Multiples of 8 bit** − A preferred block size is a multiple of 8 as it is easy for implementation as most computer processor handle data in multiple of 8 bits.

Padding in Block Cipher

Block ciphers process blocks of fixed sizes (say 64 bits). The length of plaintexts is mostly not a multiple of the block size. For example, a 150-bit plaintext provides two blocks of 64 bits each with third block of balance 22 bits. The last block of bits needs to be padded up with redundant information so that the length of the final block equal to block size of the scheme. In our example, the remaining 22 bits need to have additional 42 redundant bits added to provide a complete block. The process of adding bits to the last block is referred to as **padding**.

Too much padding makes the system inefficient. Also, padding may render the system insecure at times, if the padding is done with same bits always.

Block Cipher Schemes

There is a vast number of block ciphers schemes that are in use. Many of them are publically known. Most popular and prominent block ciphers are listed below.

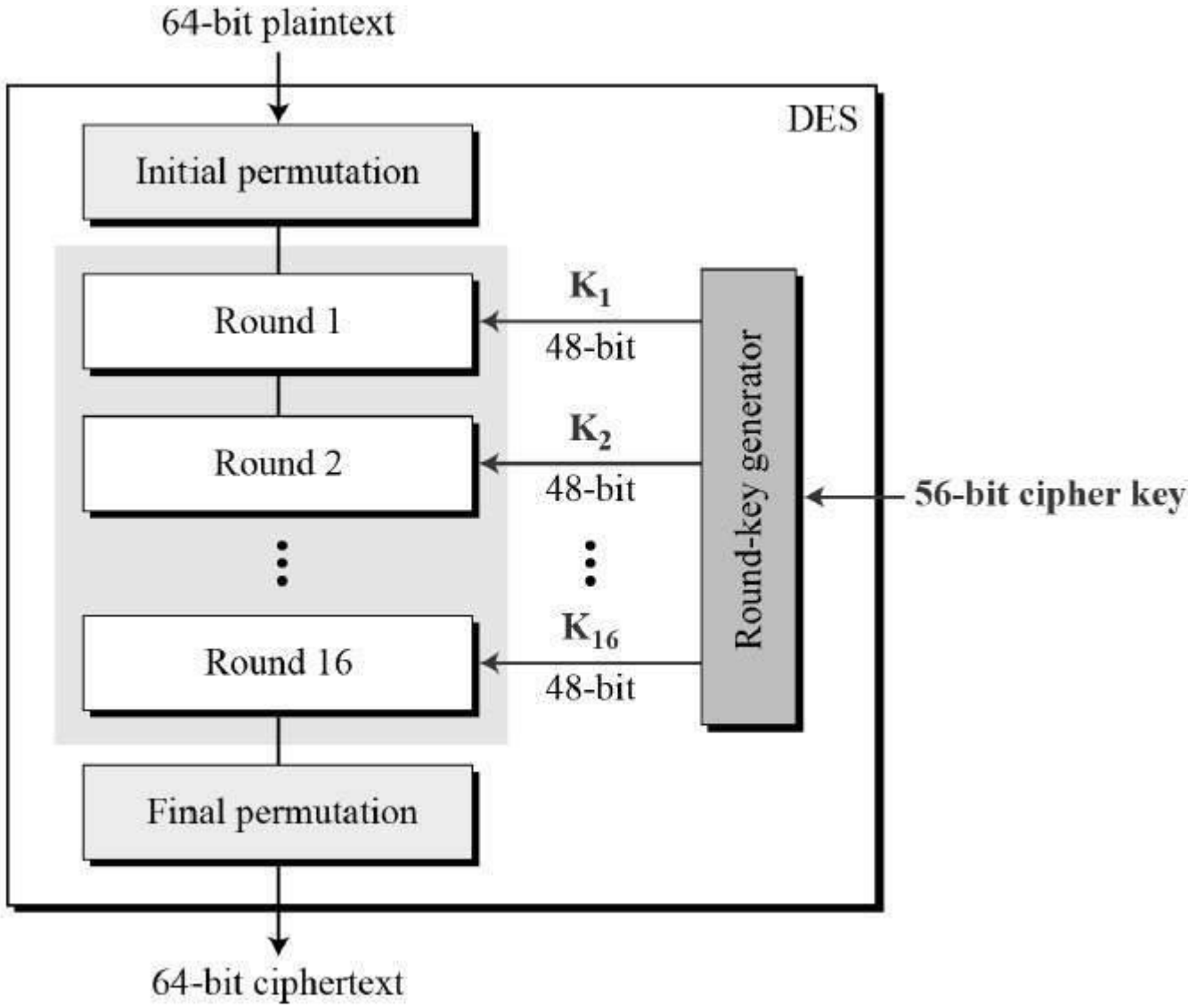
- **Digital Encryption Standard (DES)** − The popular block cipher of the 1990s. It is now considered as a 'broken' block cipher, due primarily to its small key size.
- **Triple DES** − It is a variant scheme based on repeated DES applications. It is still a respected block ciphers but inefficient compared to the new faster block ciphers available.
- **Advanced Encryption Standard (AES)** − It is a relatively new block cipher based on the encryption algorithm **Rijndael** that won the AES design competition.
- **IDEA** − It is a sufficiently strong block cipher with a block size of 64 and a key size of 128 bits. A number of applications use IDEA encryption, including early versions of Pretty Good Privacy (PGP) protocol. The use of IDEA scheme has a restricted adoption due to patent issues.
- **Twofish** − This scheme of block cipher uses block size of 128 bits and a key of variable length. It was one of the AES finalists. It is based on the earlier block cipher Blowfish with a block size of 64 bits.
- **Serpent** − A block cipher with a block size of 128 bits and key lengths of 128, 192, or 256 bits, which was also an AES competition finalist. It is a slower but has more secure design than other block cipher.

In the next sections, we will first discuss the model of block cipher followed by DES and AES, two of the most influential modern block ciphers.

Data Encryption Standard:

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

DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration −

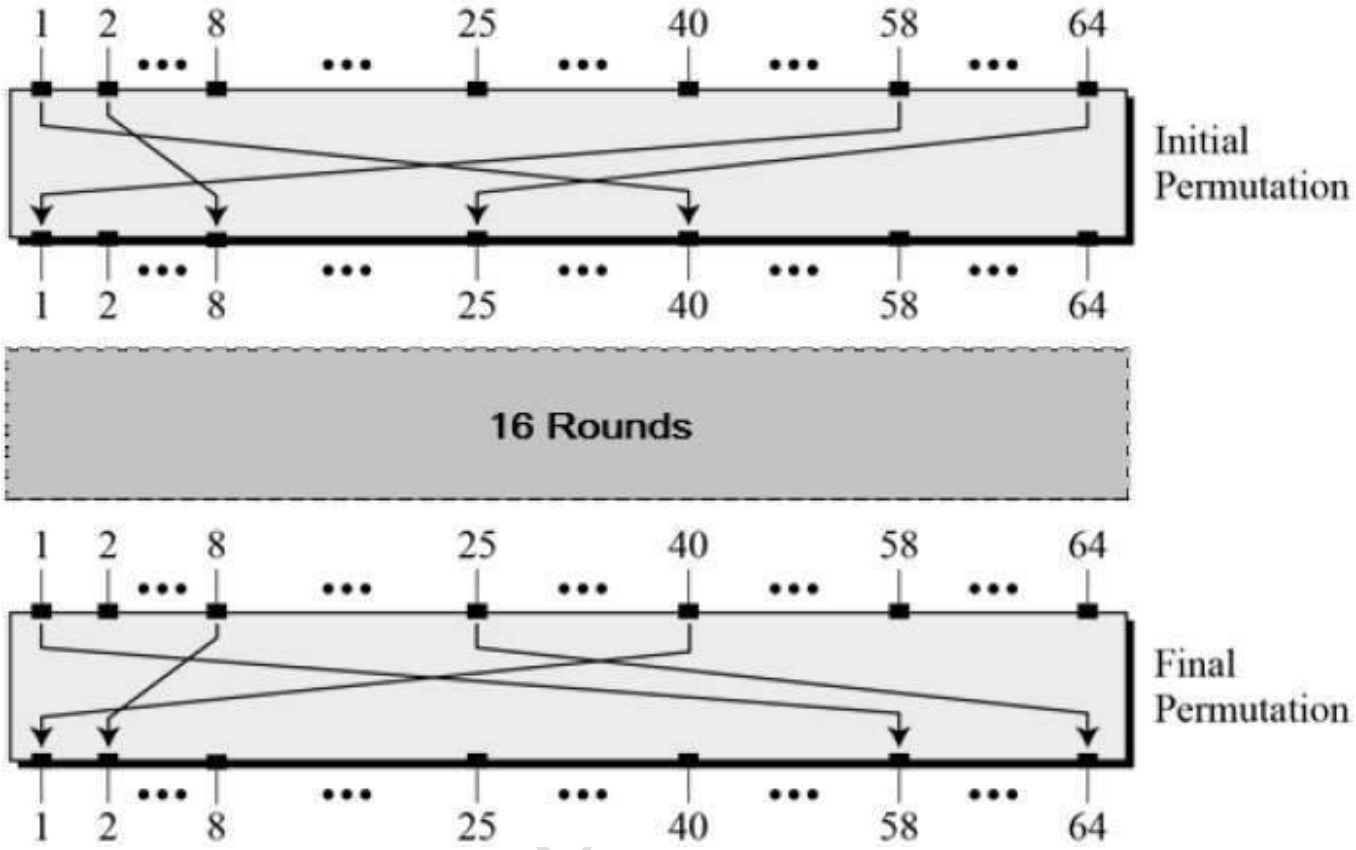


Since DES is based on the Feistel Cipher, all that is required to specify DES is −

- Round function
- Key schedule
- Any additional processing − Initial and final permutation

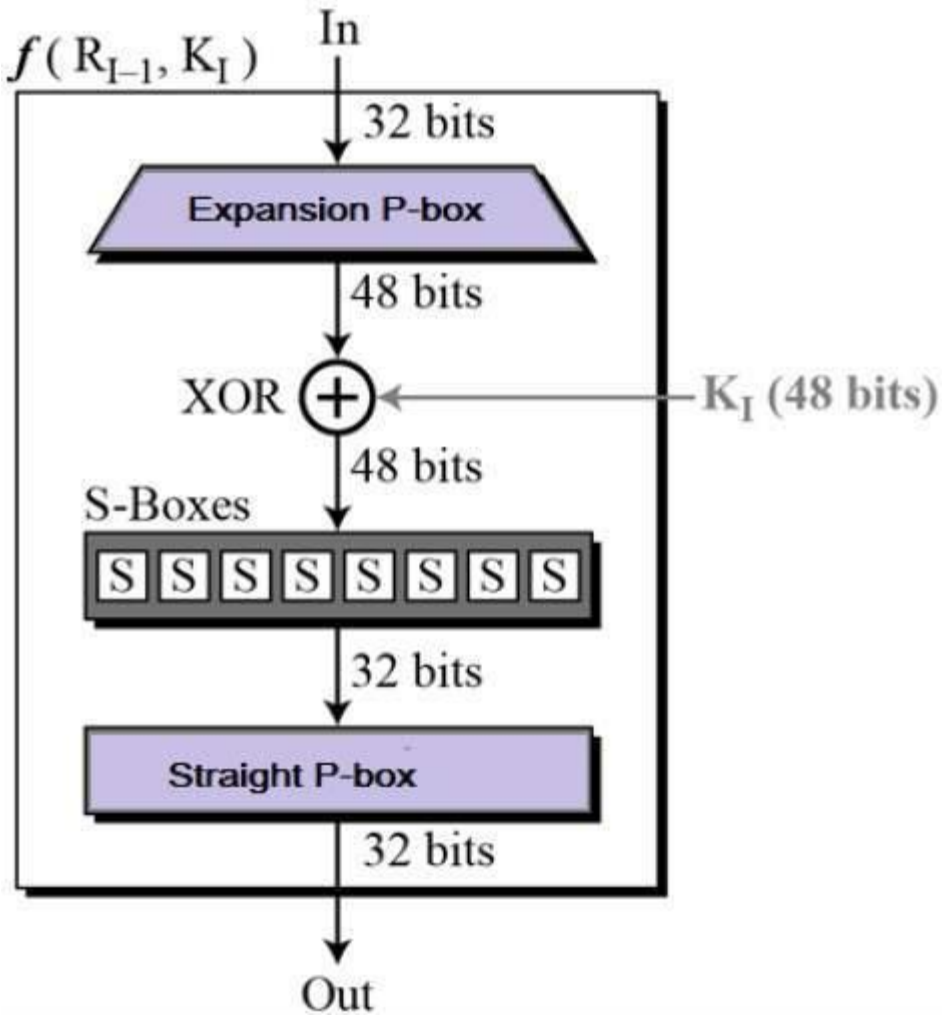
Initial and Final Permutation

The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in DES. The initial and final permutations are shown as follows −

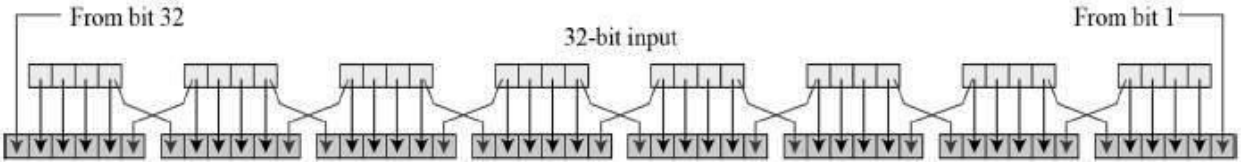


Round Function

The heart of this cipher is the DES function, f. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



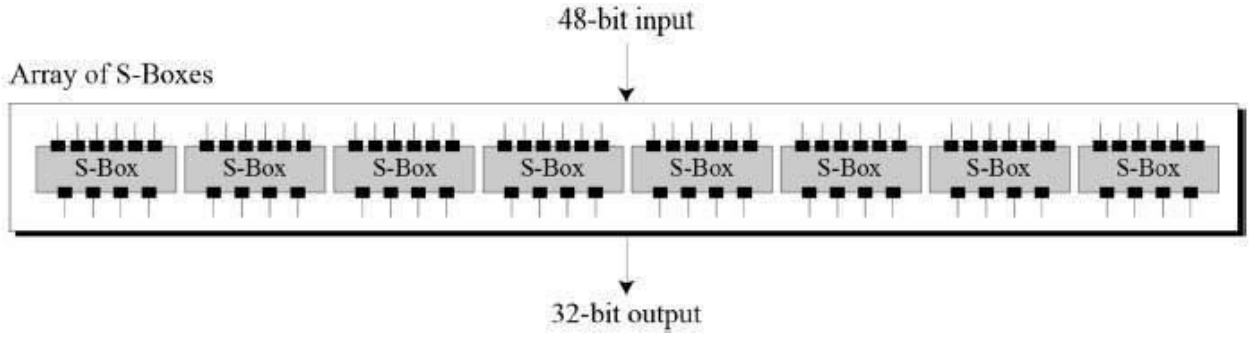
- **Expansion Permutation Box** − Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits. Permutation logic is graphically depicted in the following illustration −



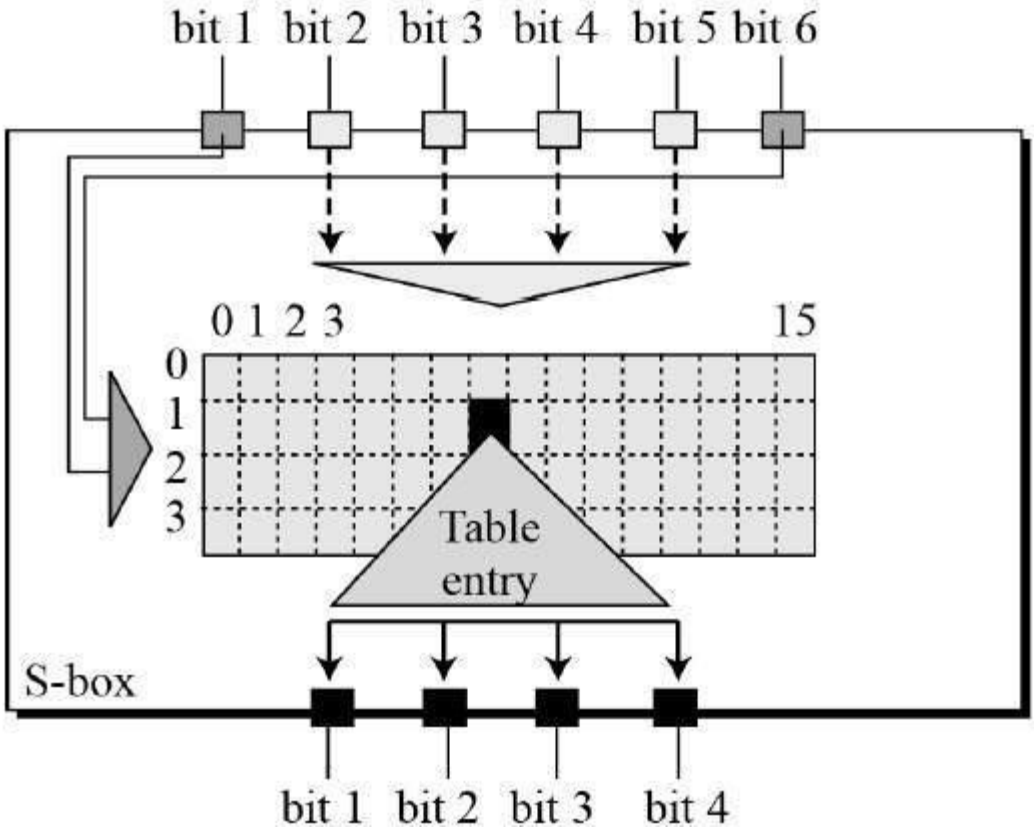
- The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown −

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

- **XOR (Whitener).** − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.
- **Substitution Boxes.** − The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration –



- The S-box rule is illustrated below −

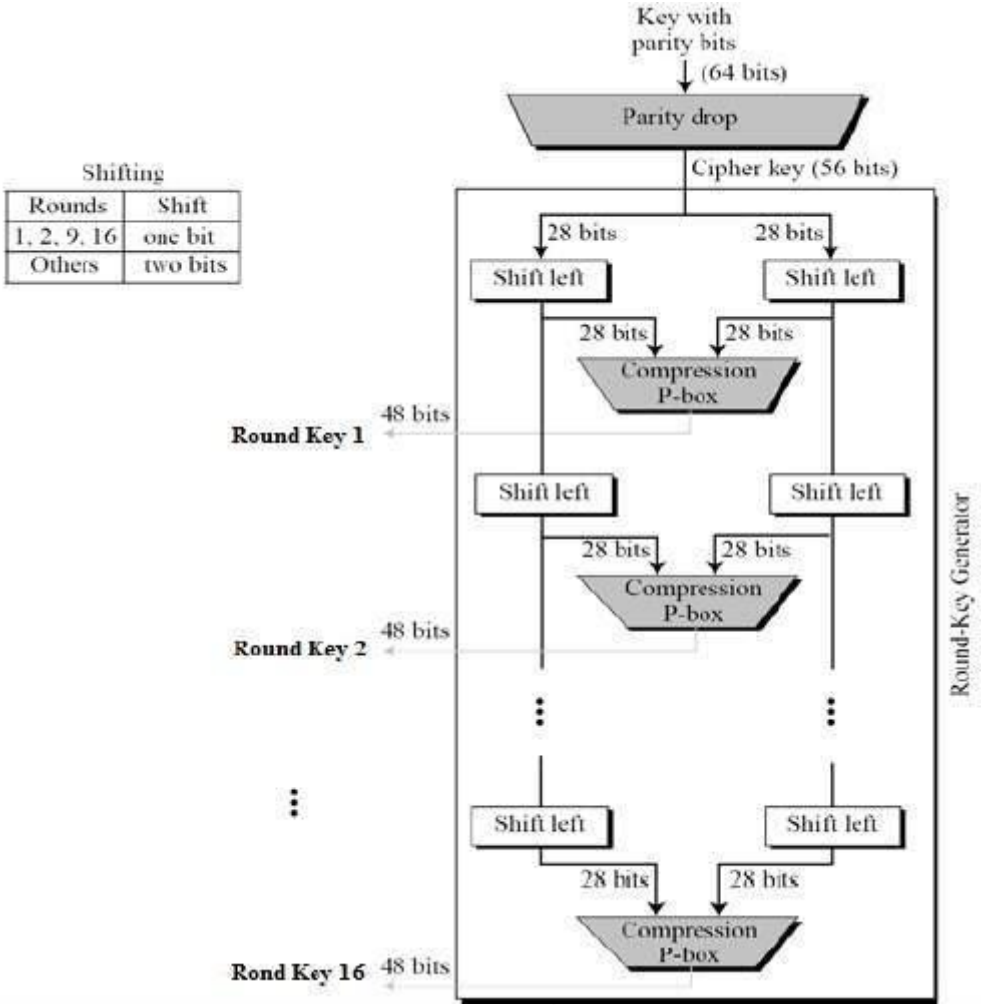


- There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.
- **Straight Permutation** − The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Key Generation

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is depicted in the following illustration −



The logic for Parity drop, shifting, and Compression P-box is given in the DES description.

DES Analysis

The DES satisfies both the desired properties of block cipher. These two properties make cipher very strong.

- **Avalanche effect** − A small change in plaintext results in the very great change in the ciphertext.
- **Completeness** − Each bit of ciphertext depends on many bits of plaintext.

During the last few years, cryptanalysis have found some weaknesses in DES when key selected are weak keys. These keys shall be avoided.

DES has proved to be a very well designed block cipher. There have been no significant cryptanalytic attacks on DES other than exhaustive key search.

Advanced Encryption Standard.

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

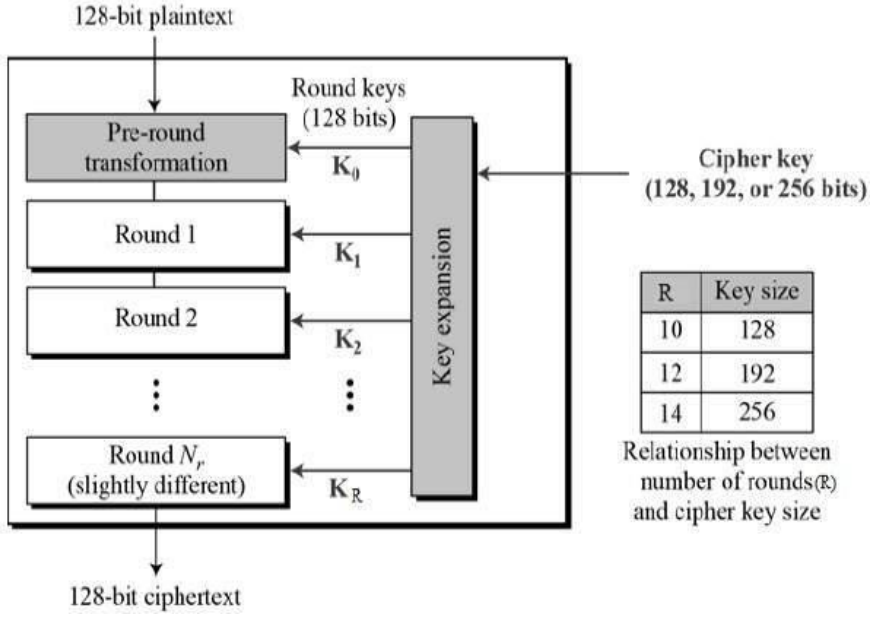
Operation of AES

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The schematic of AES structure is given in the following illustration −

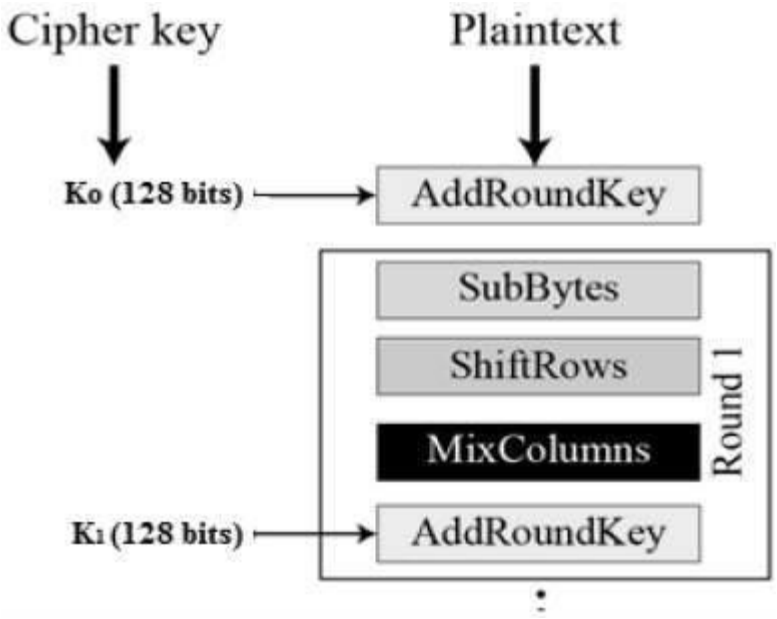


R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below −



Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows −

- First row is not shifted.
- Second row is shifted one (byte) position to the left.
- Third row is shifted two positions to the left.
- Fourth row is shifted three positions to the left.
- The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

MixColumns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

Decryption Process

The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order −

- Add round key
- Mix columns
- Shift rows
- Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms needs to be separately implemented, although they are very closely related.

AES Analysis

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches.

However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

UNIT- III

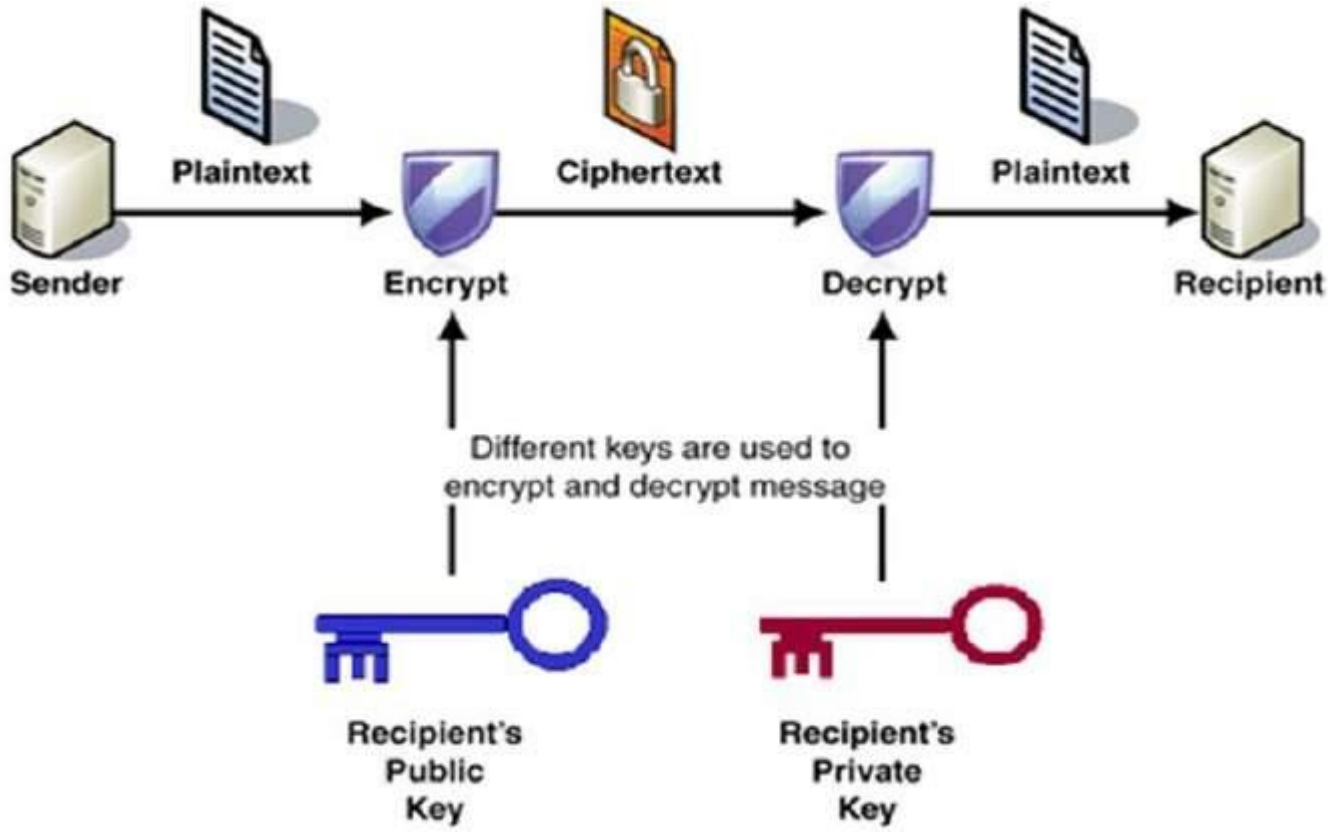
Asymmetric Encryption Mathematics of Asymmetric Key Cryptography:

Unlike symmetric key cryptography, we do not find historical use of public-key cryptography. It is a relatively new concept.

Symmetric cryptography was well suited for organizations such as governments, military, and big financial corporations were involved in the classified communication.

With the spread of more unsecure computer networks in last few decades, a genuine need was felt to use cryptography at larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems.

The process of encryption and decryption is depicted in the following illustration −



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.

There are three types of Public Key Encryption schemes. We discuss them in following sections −

RSA Cryptosystem

This cryptosystem is one the initial system. It remains most employed cryptosystem even today. The system was invented by three scholars **Ron Rivest, Adi Shamir,** and **Len Adleman** and hence, it is termed as RSA cryptosystem.

We will see two aspects of the RSA cryptosystem, firstly generation of key pair and secondly encryption-decryption algorithms.

Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below −

- **Generate the RSA modulus (n)**
  - Select two large primes, p and q.
  - Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.
- **Find Derived Number (e)**
  - Number **e** must be greater than 1 and less than (p − 1)(q − 1).
  - There must be no common factor for e and (p − 1)(q − 1) except for 1. In other words two numbers e and (p – 1)(q – 1) are coprime.
- **Form the public key**
  - The pair of numbers (n, e) form the RSA public key and is made public.
  - Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.
- **Generate the private key**
  - Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
  - Number d is the inverse of e modulo (p - 1)(q – 1). This means that d is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1).

- This relationship is written mathematically as follows −

$ed = 1 \bmod (p - 1)(q - 1)$

The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output.

Example

An example of generating RSA Key pair is given below. (For ease of understanding, the primes p & q taken here are small values. Practically, these values are very high).

- Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.
- Select e = 5, which is a valid choice since there is no number that is common factor of 5 and $(p - 1)(q - 1) = 6 \times 12 = 72$, except for 1.
- The pair of numbers (n, e) = (91, 5) forms the public key and can be made available to anyone whom we wish to be able to send us encrypted messages.
- Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output will be d = 29.
- Check that the d calculated is correct by computing −

$de = 29 \times 5 = 145 = 1 \bmod 72$

- Hence, public key is (91, 5) and private keys is (91, 29).

Encryption and Decryption

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy.

Interestingly, RSA does not directly operate on strings of bits as in case of symmetric key encryption. It operates on numbers modulo n. Hence, it is necessary to represent the plaintext as a series of numbers less than n.

RSA Encryption

- Suppose the sender wish to send some text message to someone whose public key is (n, e).
- The sender then represents the plaintext as a series of numbers less than n.
- To encrypt the first plaintext P, which is a number modulo n. The encryption process is simple mathematical step as −

$C = P^e \bmod n$

- In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then reduced modulo n. This means that C is also a number less than n.
- Returning to our Key Generation example with plaintext P = 10, we get ciphertext C −

$C = 10^5 \bmod 91$

RSA Decryption

- The decryption process for RSA is also very straightforward. Suppose that the receiver of public-key pair (n, e) has received a ciphertext C.

- Receiver raises C to the power of his private key d. The result modulo n will be the plaintext P.

Plaintext = C^d mod n

- Returning again to our numerical example, the ciphertext C = 82 would get decrypted to number 10 using private key 29 −

Plaintext = 82^{29} mod 91 = 10

RSA Analysis

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

- **Encryption Function** − It is considered as a one-way function of converting plaintext into ciphertext and it can be reversed only with the knowledge of private key d.
- **Key Generation** − The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n. An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless he can factor n. It is also a one way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken. In fact, if a technique for factoring efficiently is developed then RSA will no longer be safe.

The strength of RSA encryption drastically goes down against attacks if the number p and q are not large primes and/ or chosen public key e is a small number.

ElGamal Cryptosystem

Along with RSA, there are other public-key cryptosystems proposed. Many of them are based on different versions of the Discrete Logarithm Problem.

ElGamal cryptosystem, called Elliptic Curve Variant, is based on the Discrete Logarithm Problem. It derives the strength from the assumption that the discrete logarithms cannot be found in practical time frame for a given number, while the inverse operation of the power can be computed efficiently.

Let us go through a simple version of ElGamal that works with numbers modulo p. In the case of elliptic curve variants, it is based on quite different number systems.

Generation of ElGamal Key Pair

Each user of ElGamal cryptosystem generates the key pair through as follows −

- **Choosing a large prime p.** Generally a prime number of 1024 to 2048 bits length is chosen.
- **Choosing a generator element g.**
    - This number must be between 1 and p − 1, but cannot be any number.
    - It is a generator of the multiplicative group of integers modulo p. This means for every integer m co-prime to p, there is an integer k such that g^k=a mod n.

For example, 3 is generator of group 5 (Z_5 = {1, 2, 3, 4}).

N	3^n	3^n mod 5
1	3	3
2	9	4
3	27	2
4	81	1

- **Choosing the private key.** The private key x is any number bigger than 1 and smaller than p−1.
- **Computing part of the public key.** The value y is computed from the parameters p, g and the private key x as follows −

$y = g^x \bmod p$

- **Obtaining Public key.** The ElGamal public key consists of the three parameters (p, g, y).

  For example, suppose that p = 17 and that g = 6 (It can be confirmed that 6 is a generator of group Z_{17}). The private key x can be any number bigger than 1 and smaller than 71, so we choose x = 5. The value y is then computed as follows −

$y = 6^5 \bmod 17 = 7$

- Thus the private key is 62 and the public key is (17, 6, 7).

Encryption and Decryption

The generation of an ElGamal key pair is comparatively simpler than the equivalent process for RSA. But the encryption and decryption are slightly more complex than RSA.

ElGamal Encryption

Suppose sender wishes to send a plaintext to someone whose ElGamal public key is (p, g, y), then −

- Sender represents the plaintext as a series of numbers modulo p.
- To encrypt the first plaintext P, which is represented as a number modulo p. The encryption process to obtain the ciphertext C is as follows −
  - Randomly generate a number k;
  - Compute two values C1 and C2, where −

$C1 = g^k \bmod p$

$C2 = (P^{*}y^{k}) \bmod p$

- Send the ciphertext C, consisting of the two separate values (C1, C2), sent together.
- Referring to our ElGamal key generation example given above, the plaintext P = 13 is encrypted as follows −
  - Randomly generate a number, say k = 10
  - Compute the two values C1 and C2, where −

$C1 = 6^{10} \bmod 17$
$C2 = (13^{*}7^{10}) \bmod 17 = 9$

- Send the ciphertext C = (C1, C2) = (15, 9).

ElGamal Decryption

- To decrypt the ciphertext (C1, C2) using private key x, the following two steps are taken −
  - Compute the modular inverse of $(C1)^{x}$ modulo p, which is $(C1)^{-x}$, generally referred to as decryption factor.
  - Obtain the plaintext by using the following formula −

$C2 \times (C1)^{-x} \bmod p = \text{Plaintext}$

- In our example, to decrypt the ciphertext C = (C1, C2) = (15, 9) using private key x = 5, the decryption factor is

$15^{-5} \bmod 17 = 9$

- Extract plaintext $P = (9 \times 9) \bmod 17 = 13$.

ElGamal Analysis

In ElGamal system, each user has a private key x. and has **three components** of public key − **prime modulus p, generator g, and public $Y = g^{x} \bmod p$**. The strength of the ElGamal is based on the difficulty of discrete logarithm problem.

The secure key size is generally > 1024 bits. Today even 2048 bits long key are used. On the processing speed front, Elgamal is quite slow, it is used mainly for key authentication protocols. Due to higher processing efficiency, Elliptic Curve variants of ElGamal are becoming increasingly popular.

Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is a term used to describe a suite of cryptographic tools and protocols whose security is based on special versions of the discrete logarithm problem. It does not use numbers modulo p.

ECC is based on sets of numbers that are associated with mathematical objects called elliptic curves. There are rules for adding and computing multiples of these numbers, just as there are for numbers modulo p.

ECC includes a variants of many cryptographic schemes that were initially designed for modular numbers such as ElGamal encryption and Digital Signature Algorithm.

It is believed that the discrete logarithm problem is much harder when applied to points on an elliptic curve. This prompts switching from numbers modulo p to points on an elliptic curve. Also an equivalent security level can be obtained with shorter keys if we use elliptic curve-based variants.

The shorter keys result in two benefits −

- Ease of key management
- Efficient computation

These benefits make elliptic-curve-based variants of encryption scheme highly attractive for application where computing resources are constrained.

RSA and ElGamal Schemes – A Comparison

Let us briefly compare the RSA and ElGamal schemes on the various aspects.

RSA	ElGamal
It is more efficient for encryption.	It is more efficient for decryption.
It is less efficient for decryption.	It is more efficient for decryption.
For a particular security level, lengthy keys are required in RSA.	For the same level of security, very short keys are required.
It is widely accepted and used.	It is new and not very popular in market.

UNIT- IV

Data Integrity:

Threats to Data Integrity

When sensitive information is exchanged, the receiver must have the assurance that the message has come intact from the intended sender and is not modified inadvertently or otherwise. There are two different types of data integrity threats, namely **passive** and **active**.

Passive Threats

This type of threats exists due to accidental changes in data.

- These data errors are likely to occur due to noise in a communication channel. Also, the data may get corrupted while the file is stored on a disk.
- Error-correcting codes and simple checksums like Cyclic Redundancy Checks (CRCs) are used to detect the loss of data integrity. In these techniques, a digest of data is computed mathematically and appended to the data.

Active Threats

In this type of threats, an attacker can manipulate the data with malicious intent.

- At simplest level, if data is without digest, it can be modified without detection. The system can use techniques of appending CRC to data for detecting any active modification.
- At higher level of threat, attacker may modify data and try to derive new digest for modified data from exiting digest. This is possible if the digest is computed using simple mechanisms such as CRC.
- Security mechanism such as Hash functions are used to tackle the active modification threats.

Digital Signature Schemes

Digital signatures are the public-key primitives of message authentication. In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind signatory to the message.

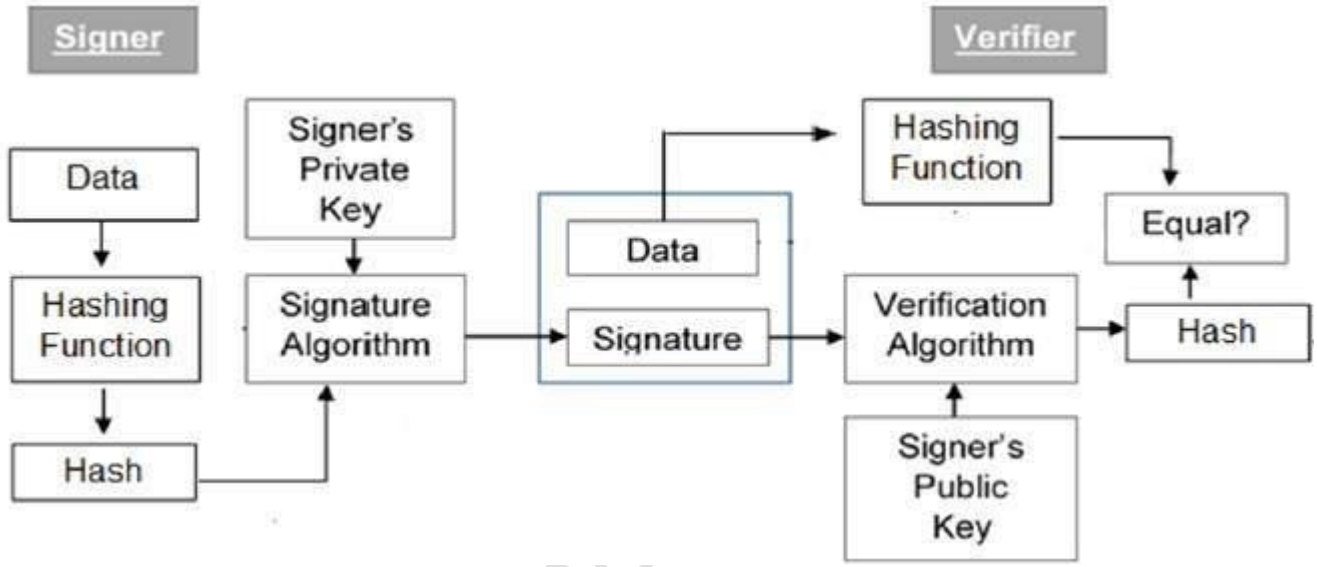
Similarly, a digital signature is a technique that binds a person/entity to the digital data. This binding can be independently verified by receiver as well as any third party.

Digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.

In real world, the receiver of message needs assurance that the message belongs to the sender and he should not be able to repudiate the origination of that message. This requirement is very crucial in business applications, since likelihood of a dispute over exchanged data is very high.

Model of Digital Signature

As mentioned earlier, the digital signature scheme is based on public key cryptography. The model of digital signature scheme is depicted in the following illustration −



The following points explain the entire process in detail −

- Each person adopting this scheme has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.
- Signer feeds data to the hash function and generates hash of data.
- Hash value and signature key are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the data and then both are sent to the verifier.
- Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.
- Verifier also runs same hash function on received data to generate hash value.
- For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.
- Since digital signature is created by 'private' key of signer and no one else can have this key; the signer cannot repudiate signing the data in future.

It should be noticed that instead of signing data directly by signing algorithm, usually a hash of data is created. Since the hash of data is a unique representation of data, it is sufficient to

sign the hash in place of data. The most important reason of using hash instead of data directly for signing is efficiency of the scheme.

Let us assume RSA is used as the signing algorithm. As discussed in public key encryption chapter, the encryption/signing process using RSA involves modular exponentiation.

Signing large data through modular exponentiation is computationally expensive and time consuming. The hash of the data is a relatively small digest of the data, hence **signing a hash is more efficient than signing the entire data**.

Importance of Digital Signature

Out of all cryptographic primitives, the digital signature using public key cryptography is considered as very important and useful tool to achieve information security.

Apart from ability to provide non-repudiation of message, the digital signature also provides message authentication and data integrity. Let us briefly see how this is achieved by the digital signature −

- **Message authentication** − When the verifier validates the digital signature using public key of a sender, he is assured that signature has been created only by sender who possess the corresponding secret private key and no one else.
- **Data Integrity** − In case an attacker has access to the data and modifies it, the digital signature verification at receiver end fails. The hash of modified data and the output provided by the verification algorithm will not match. Hence, receiver can safely deny the message assuming that data integrity has been breached.
- **Non-repudiation** − Since it is assumed that only the signer has the knowledge of the signature key, he can only create unique signature on a given data. Thus the receiver can present data and the digital signature to a third party as evidence if any dispute arises in the future.

By adding public-key encryption to digital signature scheme, we can create a cryptosystem that can provide the four essential elements of security namely − Privacy, Authentication, Integrity, and Non-repudiation.

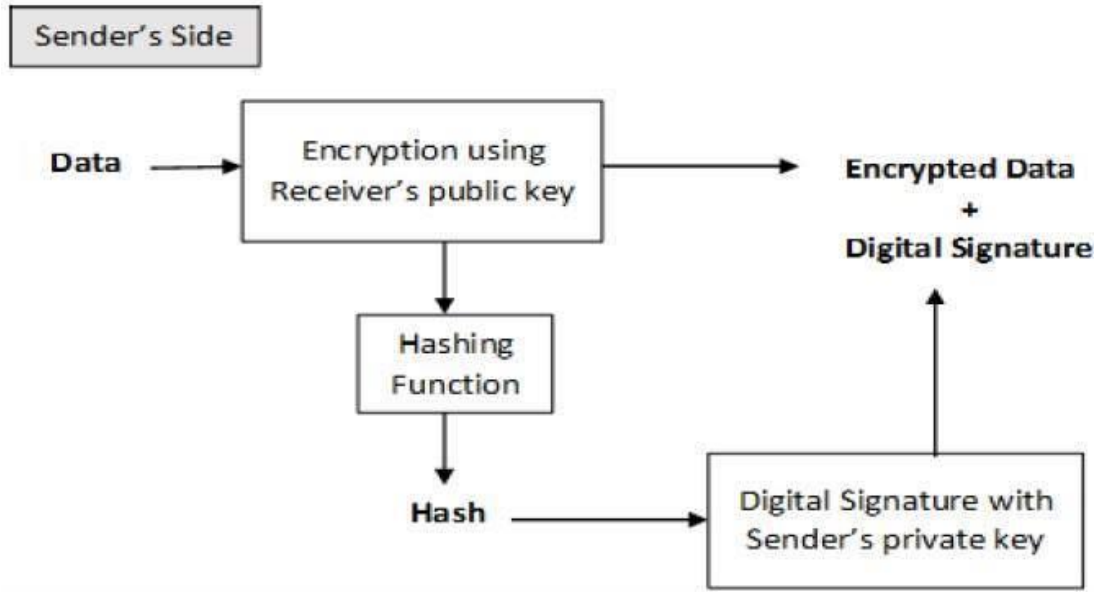
Encryption with Digital Signature

In many digital communications, it is desirable to exchange an encrypted messages than plaintext to achieve confidentiality. In public key encryption scheme, a public (encryption) key of sender is available in open domain, and hence anyone can spoof his identity and send any encrypted message to the receiver.

This makes it essential for users employing PKC for encryption to seek digital signatures along with encrypted data to be assured of message authentication and non-repudiation.

This can archived by combining digital signatures with encryption scheme. Let us briefly discuss how to achieve this requirement. There are **two possibilities, sign-then-encrypt** and **encrypt-then-sign**.

However, the crypto system based on sign-then-encrypt can be exploited by receiver to spoof identity of sender and sent that data to third party. Hence, this method is not preferred. The process of encrypt-then-sign is more reliable and widely adopted. This is depicted in the following illustration −



The receiver after receiving the encrypted data and signature on it, first verifies the signature using sender's public key. After ensuring the validity of the signature, he then retrieves the data through decryption using his private key.

Key Management

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.

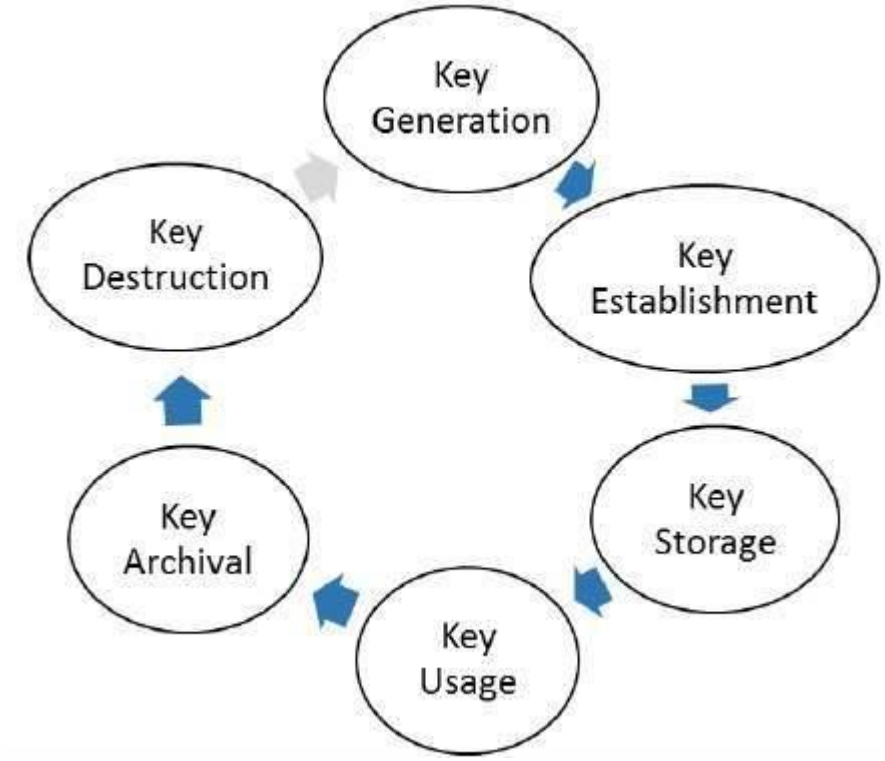
Key Management

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.

The most crucial requirement of 'assurance of public key' can be achieved through the public-key infrastructure (PKI), a key management systems for supporting public-key cryptography.

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.

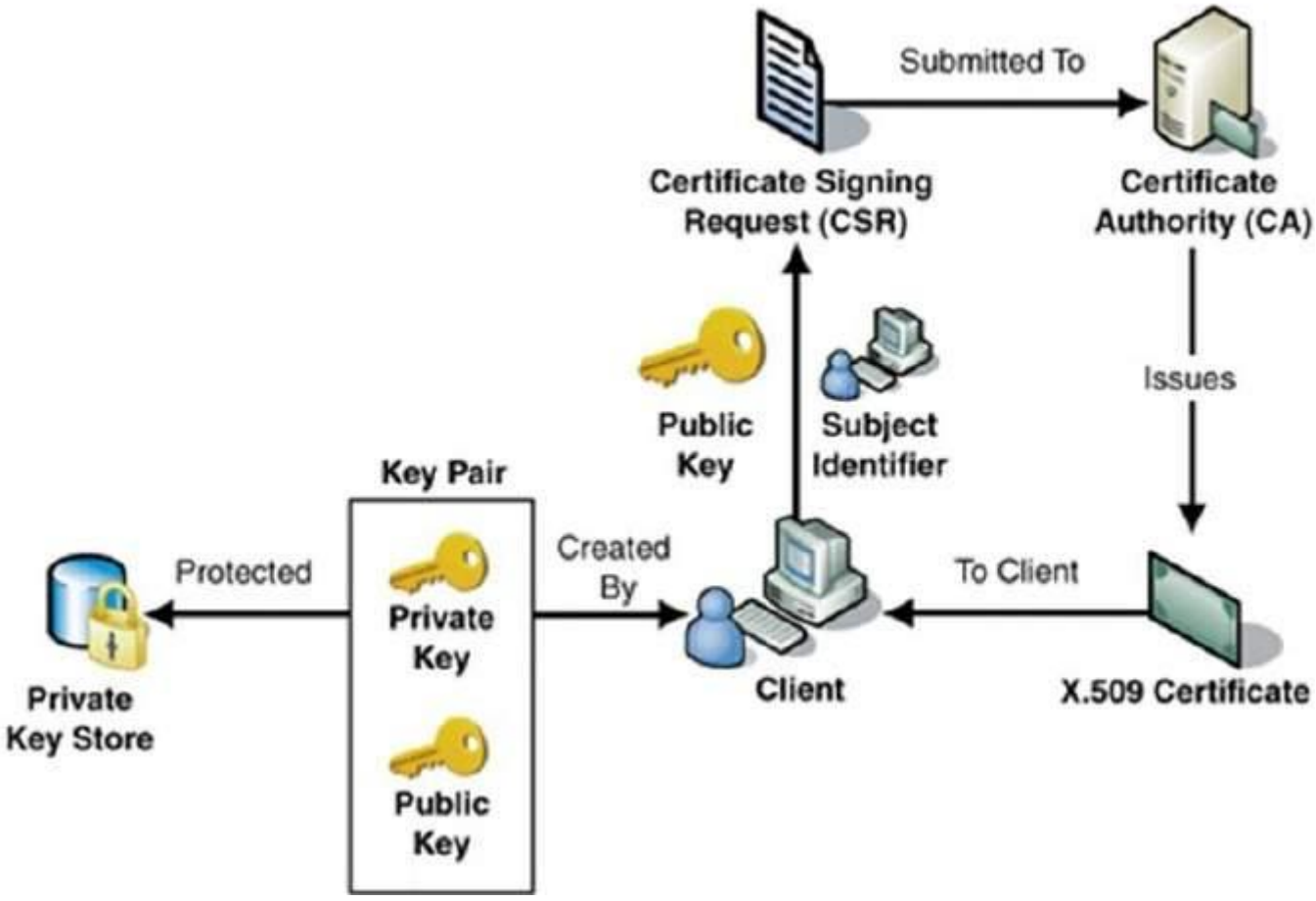
Digital Certificates are not only issued to people but they can be issued to computers, software packages or anything else that need to prove the identity in the electronic world.

- 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.

The process of obtaining Digital Certificate by a person/entity is depicted in the following illustration.



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

The key functions of a CA are as follows −

- **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.

Hierarchy of CA

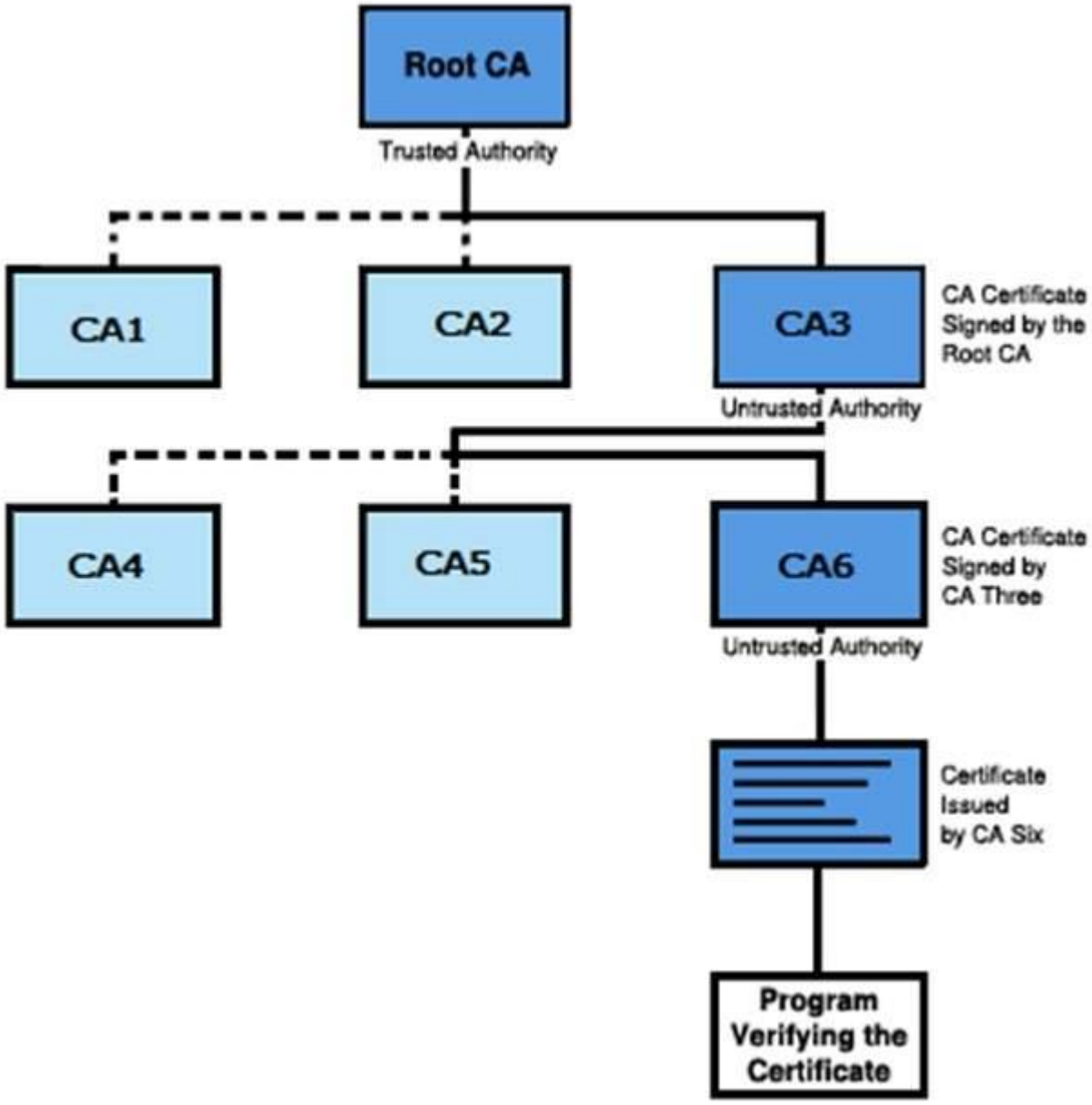
With vast networks and requirements of global communications, it is practically not feasible to have only one trusted CA from whom all users obtain their certificates. Secondly, availability of only one CA may lead to difficulties if CA is compromised.

In such case, the hierarchical certification model is of interest since it allows public key certificates to be used in environments where two communicating parties do not have trust relationships with the same CA.

- The root CA is at the top of the CA hierarchy and the root CA's certificate is a self-signed certificate.
- The CAs, which are directly subordinate to the root CA (For example, CA1 and CA2) have CA certificates that are signed by the root CA.
- The CAs under the subordinate CAs in the hierarchy (For example, CA5 and CA6) have their CA certificates signed by the higher-level subordinate CAs.

Certificate authority (CA) hierarchies are reflected in certificate chains. A certificate chain traces a path of certificates from a branch in the hierarchy to the root of the hierarchy.

The following illustration shows a CA hierarchy with a certificate chain leading from an entity certificate through two subordinate CA certificates (CA6 and CA3) to the CA certificate for the root CA.



Verifying a certificate chain is the process of ensuring that a specific certificate chain is valid, correctly signed, and trustworthy. The following procedure verifies a certificate chain, beginning with the certificate that is presented for authentication −

- A client whose authenticity is being verified supplies his certificate, generally along with the chain of certificates up to Root CA.
- Verifier takes the certificate and validates by using public key of issuer. The issuer's public key is found in the issuer's certificate which is in the chain next to client's certificate.
- Now if the higher CA who has signed the issuer's certificate, is trusted by the verifier, verification is successful and stops here.
- Else, the issuer's certificate is verified in a similar manner as done for client in above steps. This process continues till either trusted CA is found in between or else it continues till Root CA.

Message Integrity and Message Authentication:

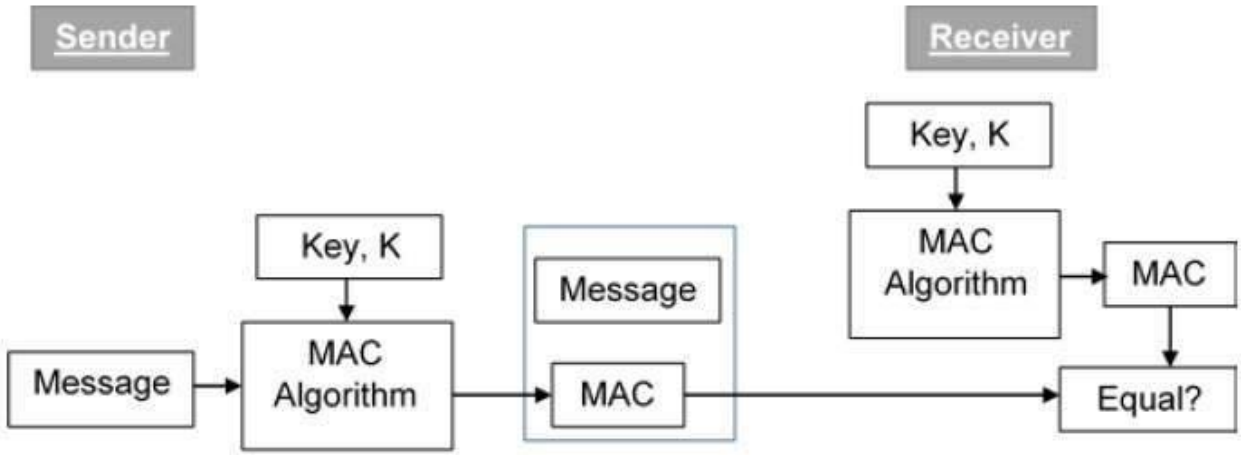
Another type of threat that exist for data is the lack of **message authentication**. In this threat, the user is not sure about the originator of the message. Message authentication can be provided using the cryptographic techniques that use secret keys as done in case of encryption.

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 process of using MAC for authentication is depicted in the following illustration −



Let us now try to understand the entire process in detail −

- 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

There are two major limitations of MAC, both due to its symmetric nature of operation −

- **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.

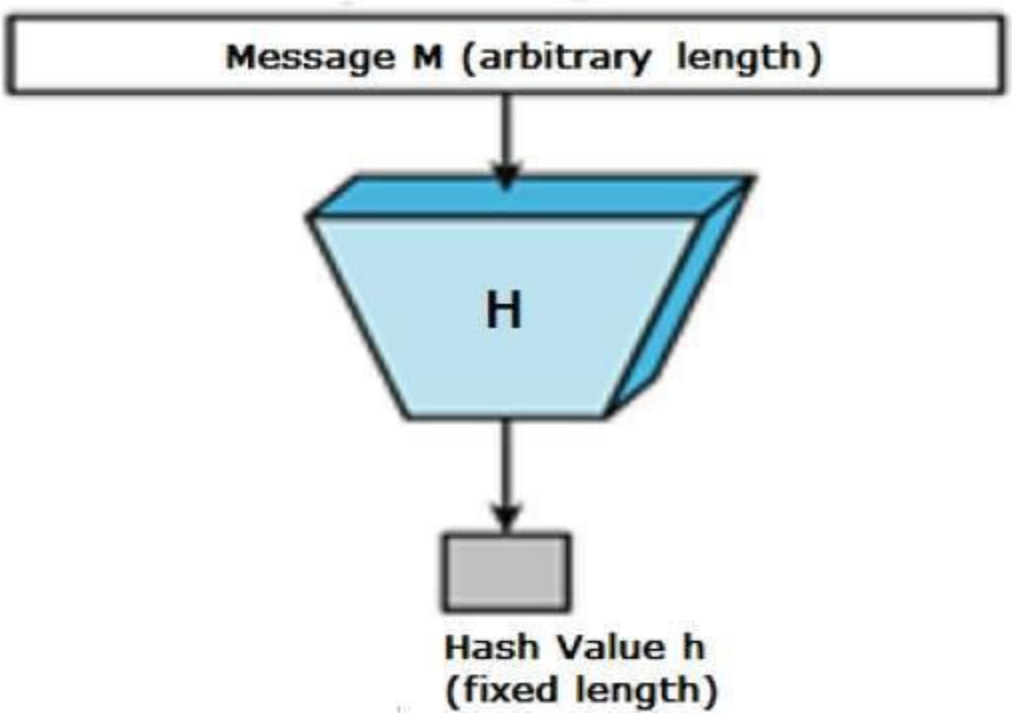
Both these limitations can be overcome by using the public key based digital signatures discussed in following section.

Cryptographic Hash Functions:

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)**
  - 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

In order to be an effective cryptographic tool, the hash function is desired to possess following properties −

- **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.

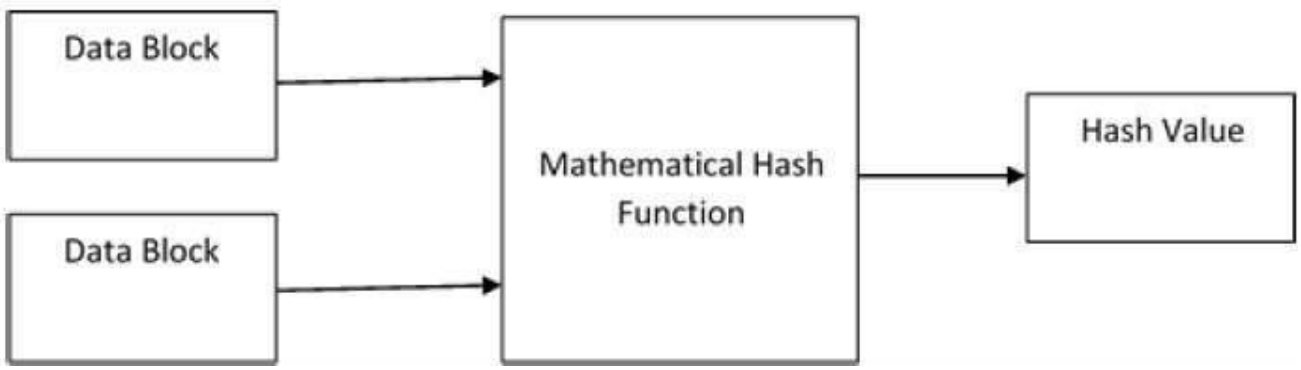
- **Second Pre-Image Resistance**
  - This property means given an input and its hash, it should be hard to find a different input with the same hash.
  - In other words, if a hash function h for an input x produces hash value h(x), then it should be difficult to find any other input value y such that 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

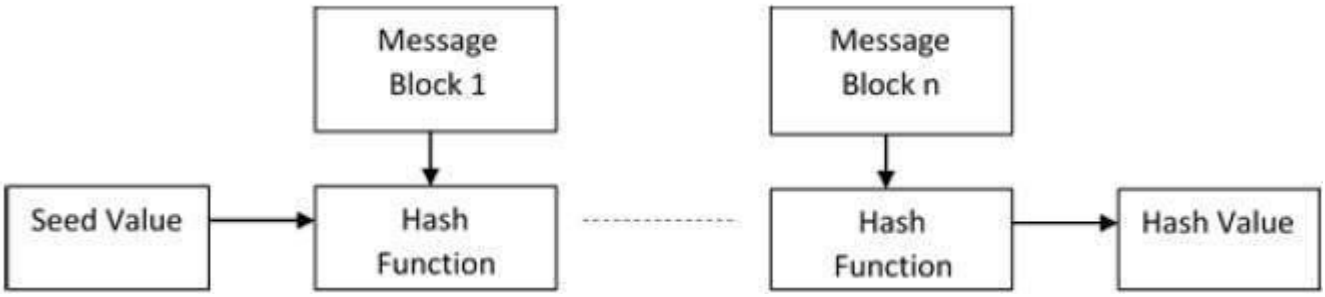
At the heart of a hashing is a mathematical function that operates on two fixed-size blocks of data to create a hash code. This hash function forms the part of the hashing algorithm.

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.

Understand the difference between hash function and algorithm correctly. The hash function generates a hash code by operating on two blocks of fixed-length binary data.

Hashing algorithm is a process for using the hash function, specifying how the message will be broken up and how the results from previous message blocks are chained together.

Popular Hash Functions

Let us briefly see some popular hash functions −

Message Digest (MD)

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.

RIPEMD

The RIPEMD is an acronym for RACE Integrity Primitives Evaluation Message Digest. This set of hash functions was designed by open research community and generally known as a family of European hash functions.

- The set includes RIPEMD, RIPEMD-128, and RIPEMD-160. There also exist 256, and 320-bit versions of this algorithm.
- Original RIPEMD (128 bit) is based upon the design principles used in MD4 and found to provide questionable security. RIPEMD 128-bit version came as a quick fix replacement to overcome vulnerabilities on the original RIPEMD.
- RIPEMD-160 is an improved version and the most widely used version in the family. The 256 and 320-bit versions reduce the chance of accidental collision, but do not have higher levels of security as compared to RIPEMD-128 and RIPEMD-160 respectively.

Whirlpool

This is a 512-bit hash function.

- It is derived from the modified version of Advanced Encryption Standard (AES). One of the designer was Vincent Rijmen, a co-creator of the AES.
- Three versions of Whirlpool have been released; namely WHIRLPOOL-0, WHIRLPOOL-T, and WHIRLPOOL.

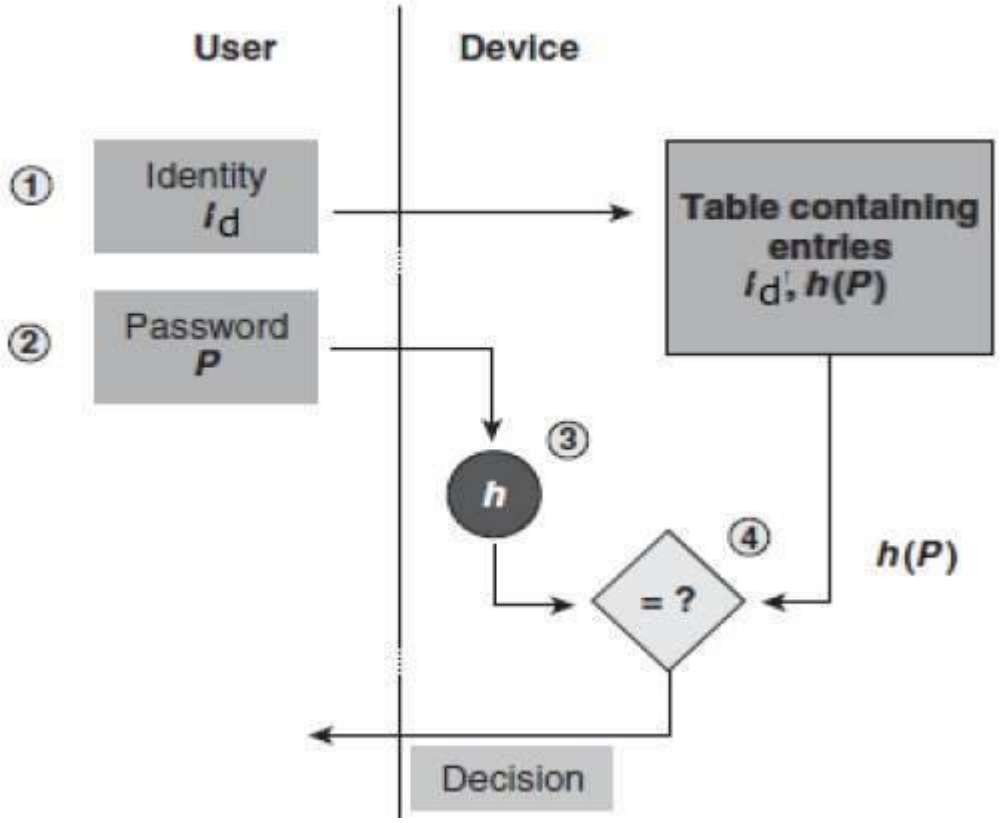
Applications of Hash Functions

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

Password Storage

Hash functions provide protection to 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 −

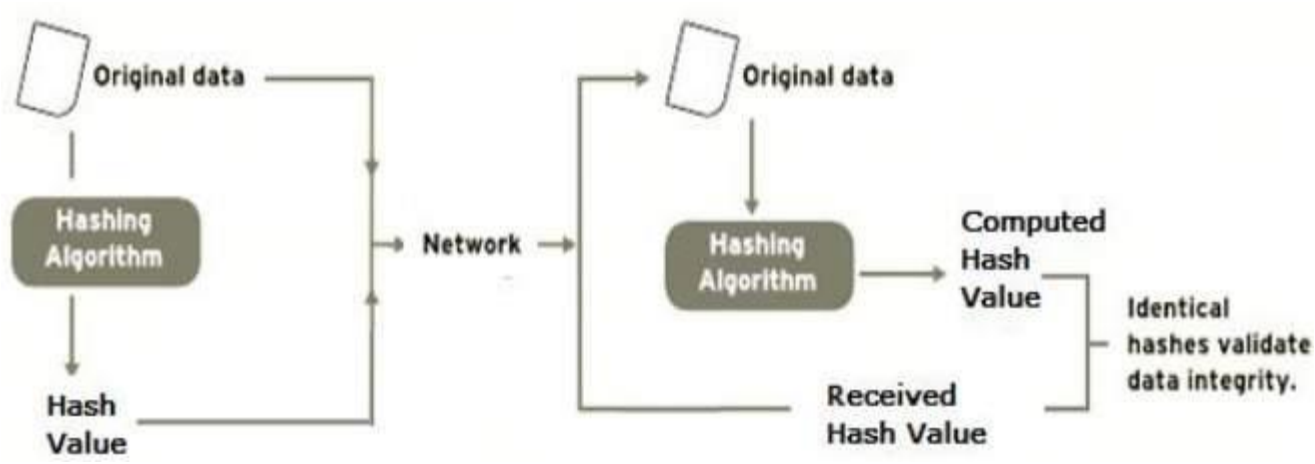


- An intruder can only see the hashes of passwords, even if he accessed the password. He can neither logon using hash nor can he derive the password from hash value since hash function possesses the property of pre-image resistance.

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.

UNIT –V

Network Security-I Security at application layer: PGP and S/MIME,

PGP (Pretty Good Privacy) is an encryption protocol which provides cryptographic privacy and authentication. PGP can sign, encrypt and decrypt data. PGP's Features: ● Confidentiality (Encryption) ● Digital signatures (Authentication)

PGP

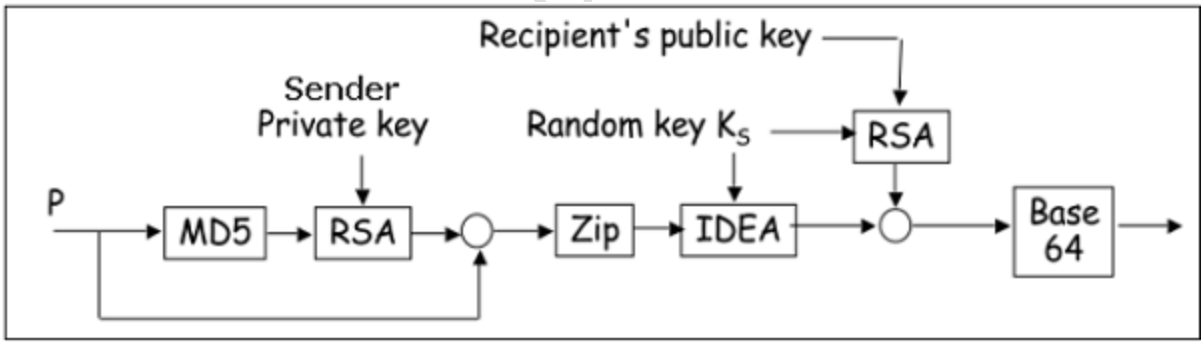
Pretty Good Privacy (PGP) is an e-mail encryption scheme. It has become the de-facto standard for providing security services for e-mail communication.

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

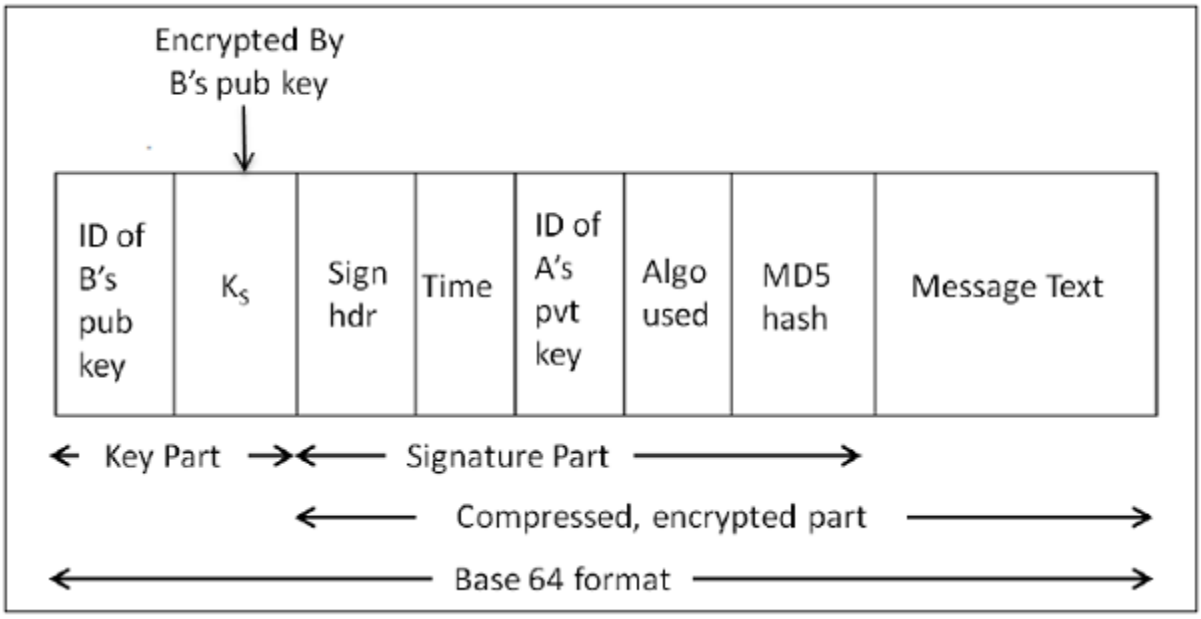
Along with these security services, it also provides data compression and key management support. PGP uses existing cryptographic algorithms such as RSA, IDEA, MD5, etc., rather than inventing the new ones.

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, K_S is generated and used to encrypt the compressed message with IDEA.
- K_S 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.

Working of S/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.

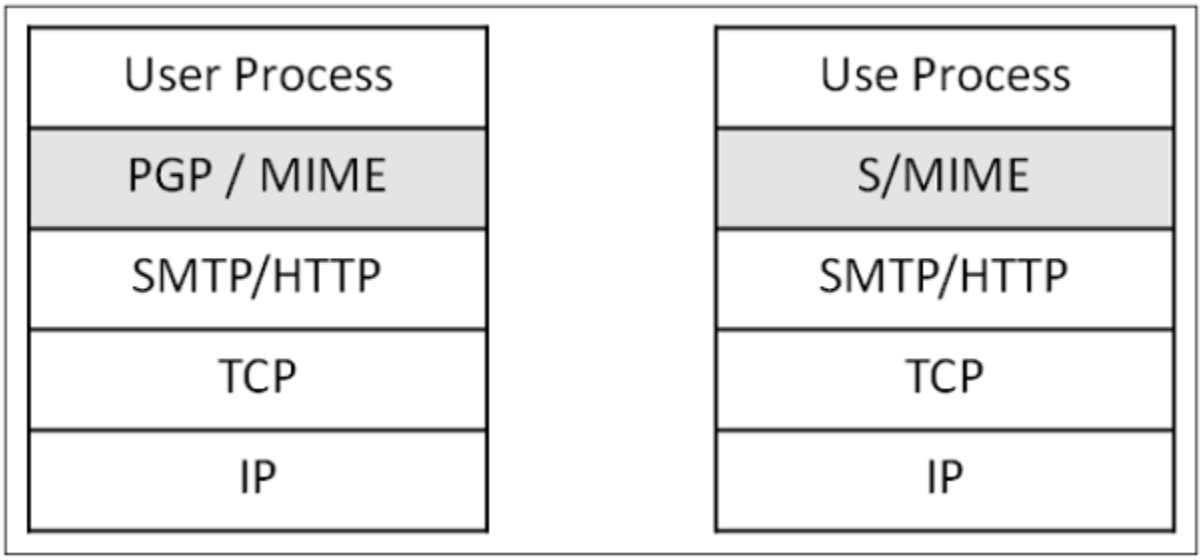
S/MIME relies on X.509 certificates for public key distribution. It needs top-down hierarchical PKI for certification support.

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.

Security at the Transport Layer: SSL and TLS

Secure Socket Layer (SSL)

In this section, 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

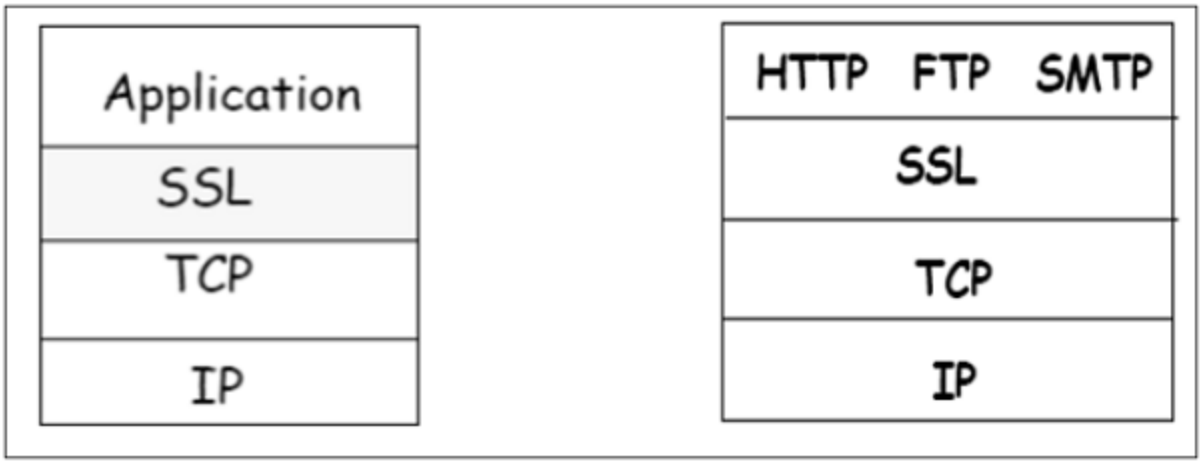
The salient features of SSL protocol are as follows −

- 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

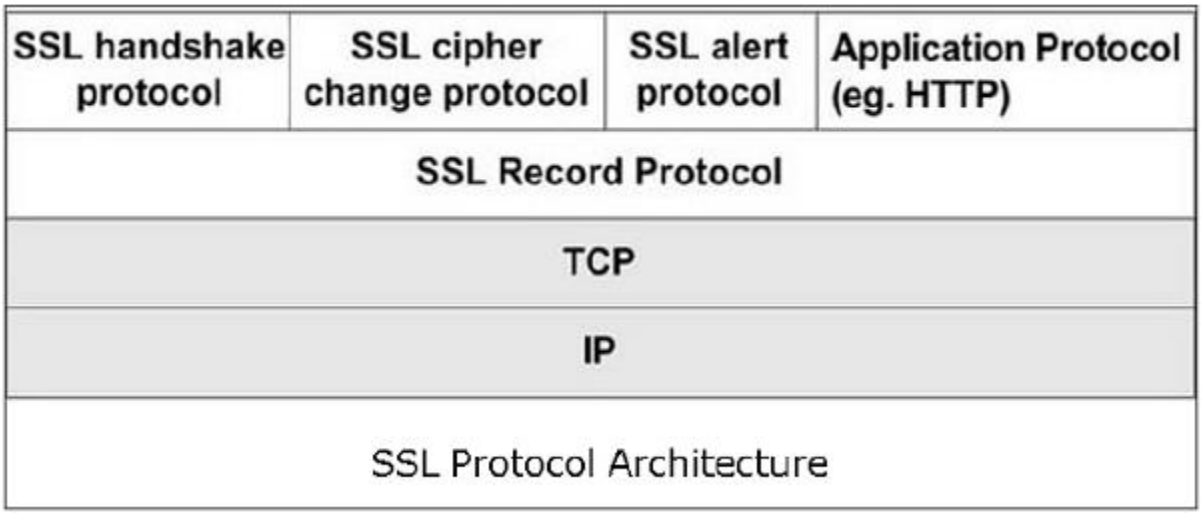
SSL is specific to TCP and it does not work with UDP. SSL provides Application Programming Interface (API) to applications. C and Java SSL libraries/classes are readily available.

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.
- These three protocols manage all of SSL message exchanges and are discussed later in this section.

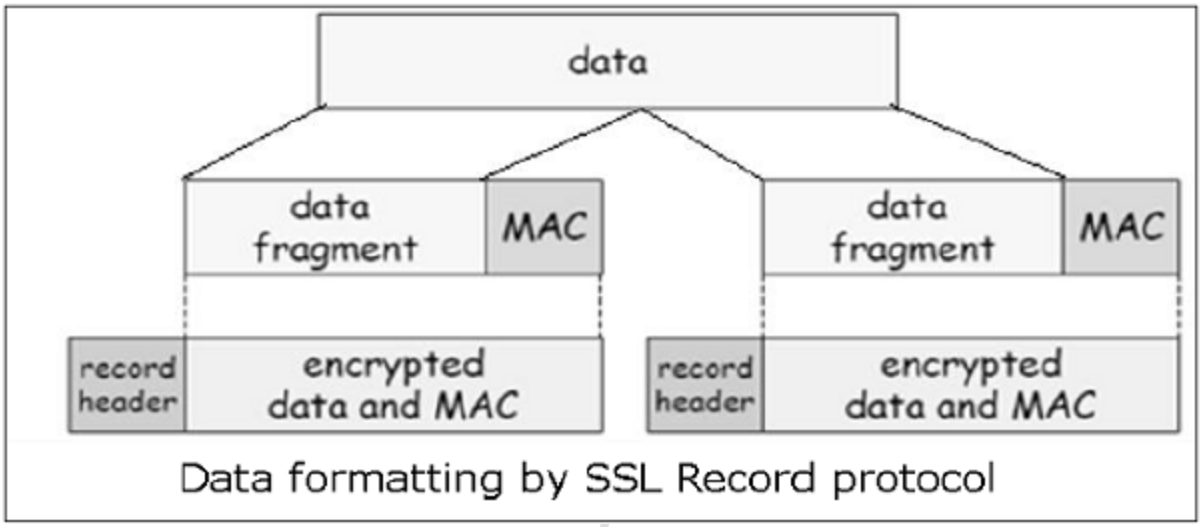


Functions of SSL Protocol Components

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.



Data formatting by SSL Record protocol

- 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. These are discussed in the next section.
- ChangeCipherSpec 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 ChangeCipherSpec 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

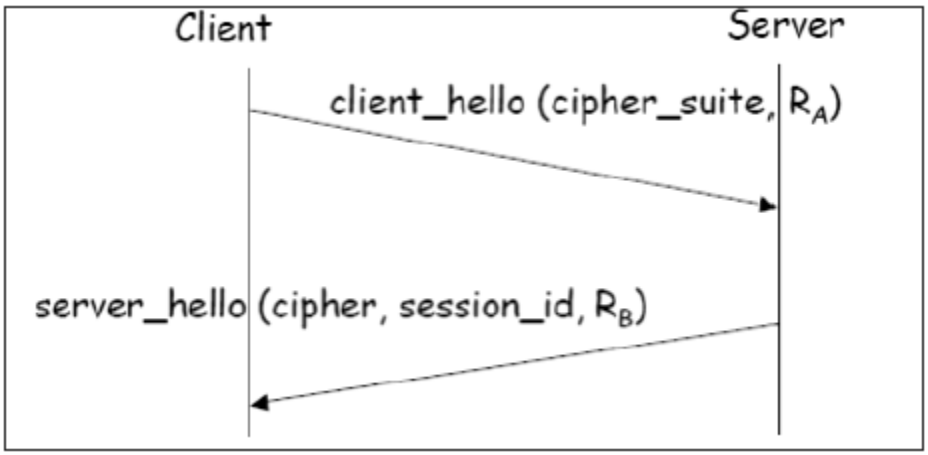
- This protocol is used to report errors – such as unexpected message, bad record MAC, security parameters negotiation failed, etc.
- It is also used for other purposes – such as notify closure of the TCP connection, notify receipt of bad or unknown certificate, etc.

Establishment of SSL Session

As discussed above, there are four phases of SSL session establishment. These are mainly handled by SSL Handshake protocol.

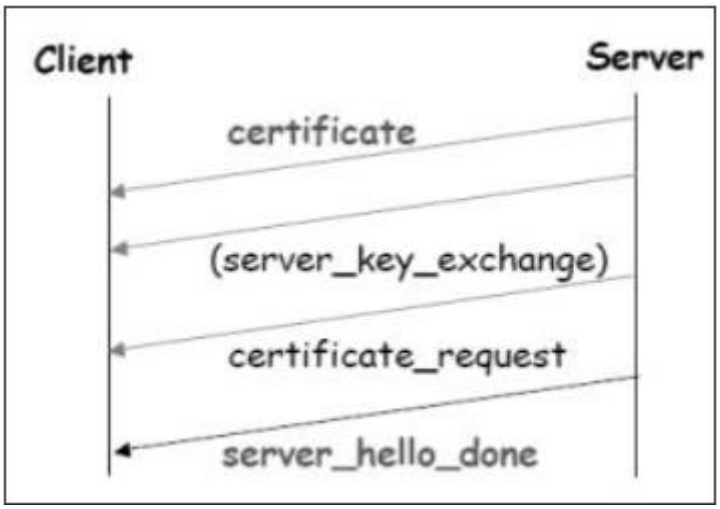
Phase 1 − Establishing security capabilities.

- This phase comprises of exchange of two messages – Client_hello and Server_hello.



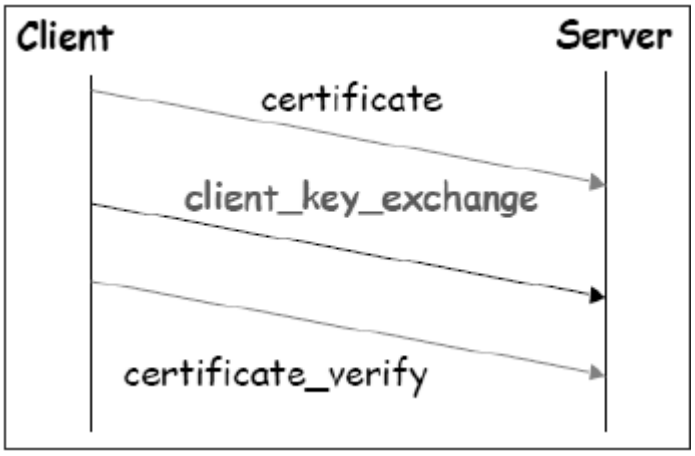
- Client_hello contains of list of cryptographic algorithms supported by the client, in decreasing order of preference.
- Server_hello contains the selected Cipher Specification (CipherSpec) and a new session_id.
- The CipherSpec contains fields like −
  - Cipher Algorithm (DES, 3DES, RC2, and RC4)
  - MAC Algorithm (based on MD5, SHA-1)
  - Public-key algorithm (RSA)
  - Both messages have “nonce” to prevent replay attack.

Phase 2 − Server authentication and key exchange.



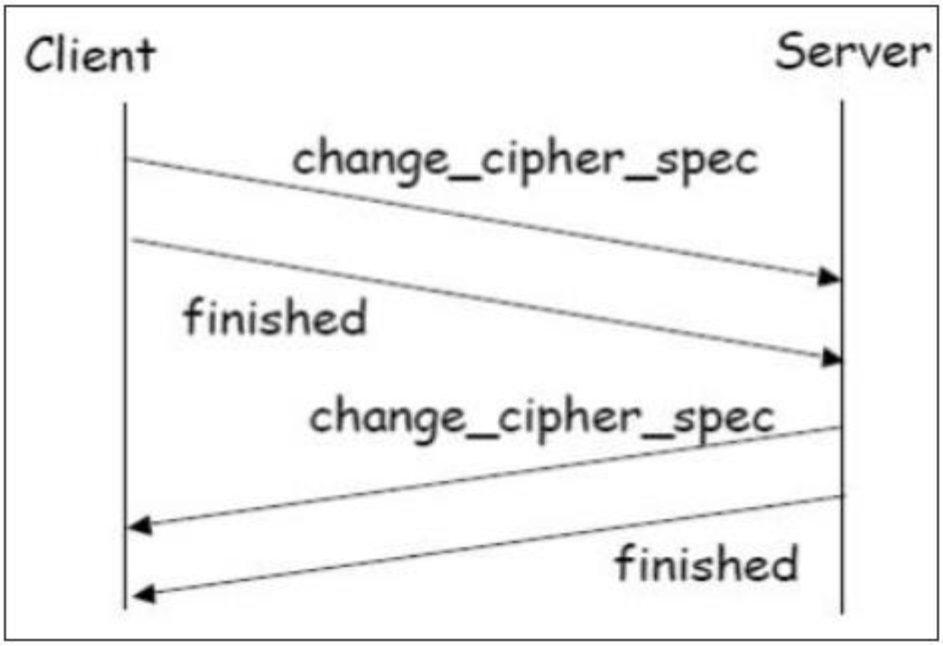
- Server sends certificate. Client software comes configured with public keys of various "trusted" organizations (CAs) to check certificate.
- Server sends chosen cipher suite.
- Server may request client certificate. Usually it is not done.
- Server indicates end of Server_hello.

Phase 3 − Client authentication and key exchange.



- Client sends certificate, only if requested by the server.
- It also sends the Pre-master Secret (PMS) encrypted with the server's public key.
- Client also sends Certificate_verify message if certificate is sent by him to prove he has the private key associated with this certificate. Basically, the client signs a hash of the previous messages.

Phase 4 − Finish.



- Client and server send Change_cipher_spec messages to each other to cause the pending cipher state to be copied into the current state.
- From now on, all data is encrypted and integrity protected.
- Message "Finished" from each end verifies that the key exchange and authentication processes were successful.

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

Resuming a Disconnected Session

- It is possible to resume a disconnected session (through Alert message), if the client sends a hello_request to the server with the encrypted session_id information.
- The server then determines if the session_id is valid. If validated, it exchanges ChangeCipherSpec and finished messages with the client and secure communications resume.
- This avoids recalculating of session cipher parameters and saves computing at the server and the client end.

SSL Session Keys

We have seen that during Phase 3 of SSL session establishment, a pre-master secret is sent by the client to the server encrypted using server's public key. The master secret and various session keys are generated as follows −

- The master secret is generated (via pseudo random number generator) using −
  - The pre-master secret.
  - Two nonces (RA and RB) exchanged in the client_hello and server_hello messages.
- Six secret values are then derived from this master secret as −

- Secret key used with MAC (for data sent by server)
- Secret key used with MAC (for data sent by client)
- Secret key and IV used for encryption (by server)
- Secret key and IV used for encryption (by client)

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

- 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.

Comparison of TLS and SSL Protocols

There are main eight differences between TLS and SSLv3 protocols. These are as follows −

- **Protocol Version** − The header of TLS protocol segment carries the version number 3.1 to differentiate between number 3 carried by SSL protocol segment header.
- **Message Authentication** − TLS employs a keyed-hash message authentication code (H-MAC). Benefit is that H-MAC operates with any hash function, not just MD5 or SHA, as explicitly stated by the SSL protocol.
- **Session Key Generation** − There are two differences between TLS and SSL protocol for generation of key material.
    - Method of computing pre-master and master secrets is similar. But in TLS protocol, computation of master secret uses the HMAC standard and pseudorandom function (PRF) output instead of ad-hoc MAC.
    - The algorithm for computing session keys and initiation values (IV) is different in TLS than SSL protocol.
- Alert Protocol Message −
    - TLS protocol supports all the messages used by the Alert protocol of SSL, except No certificate alert message being made redundant. The client sends empty certificate in case client authentication is not required.
    - Many additional Alert messages are included in TLS protocol for other error conditions such as record_overflow, decode_error etc.

- **Supported Cipher Suites** − SSL supports RSA, Diffie-Hellman and Fortezza cipher suites. TLS protocol supports all suits except Fortezza.
- **Client Certificate Types** − TLS defines certificate types to be requested in a certificate_request message. SSLv3 support all of these. Additionally, SSL support certain other types of certificate such as Fortezza.
- CertificateVerify and Finished Messages −
    - In SSL, complex message procedure is used for the certificate_verify message. With TLS, the verified information is contained in the handshake messages itself thus avoiding this complex procedure.
    - Finished message is computed in different manners in TLS and SSLv3.
- **Padding of Data** − In SSL protocol, the padding added to user data before encryption is the minimum amount required to make the total data-size equal to a multiple of the cipher's block length. In TLS, the padding can be any amount that results in data-size that is a multiple of the cipher's block length, up to a maximum of 255 bytes.

The above differences between TLS and SSLv3 protocols are summarized in the following table.

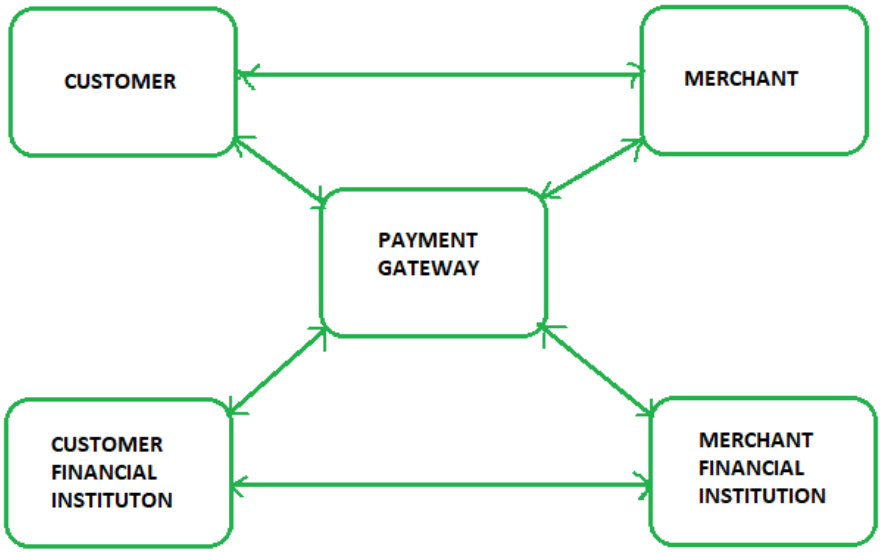
	SSL v3.0	TLS v1.0
Protocol version in messages	3.0	3.1
Alert protocol message types	12	23
Message authentication	ad hoc	standard
Key material generation	ad hoc	PRF
CertificateVerify	complex	simple
Finished	ad hoc	PRF
Baseline cipher suites	includes Fortezza	no Fortezza

Secure electronic transaction (SET):

Secure Electronic Transaction or SET is a system which ensures security and integrity of electronic transactions done using credit cards in a scenario. SET is not some system that enables payment but it is a security protocol applied on those payments. It uses different encryption and hashing techniques to secure payments over internet done through credit cards. SET protocol was supported in development by major organizations like Visa, Mastercard, Microsoft which provided its Secure Transaction Technology (STT) and NetScape which provided technology of Secure Socket Layer (SSL).
SET protocol restricts revealing of credit card details to merchants thus keeping hackers and thieves at bay. SET protocol includes Certification Authorities for making use of standard Digital Certificates like X.509 Certificate.

Before discussing SET further, let's see a general scenario of electronic transaction, which includes client, payment gateway, client financial institution, merchant and merchant

financial institution.



Requirements in SET :
SET protocol has some requirements to meet, some of the important requirements are :

- It has to provide mutual authentication i.e., customer (or cardholder) authentication by confirming if the customer is intended user or not and merchant authentication.
- It has to keep the PI (Payment Information) and OI (Order Information) confidential by appropriate encryptions.
- It has to be resistive against message modifications i.e., no changes should be allowed in the content being transmitted.
- SET also needs to provide interoperability and make use of best security mechanisms.

Participants in SET :

1. **Cardholder –** customer
2. **Issuer –** customer financial institution
3. **Merchant**
4. **Acquirer –** Merchant financial
5. **Certificate authority –** Authority which follows certain standards and issues certificates(like X.509V3) to all other participants.

SET functionalities :

- **Provide Authentication**
  - **Merchant Authentication** – To prevent theft, SET allows customers to check previous relationships between merchant and financial institution. Standard X.509V3 certificates are used for this verification.
  - **Customer / Cardholder Authentication** – SET checks if use of credit card is done by an authorized user or not using X.509V3 certificates.

- **Provide Message Confidentiality** : Confidentiality refers to preventing unintended people from reading the message being transferred. SET implements confidentiality by using encryption techniques. Traditionally DES is used for encryption purpose.
- **Provide Message Integrity** : SET doesn't allow message modification with the help of signatures. Messages are protected against unauthorized modification using RSA digital signatures with SHA-1 and some using HMAC with SHA-1,

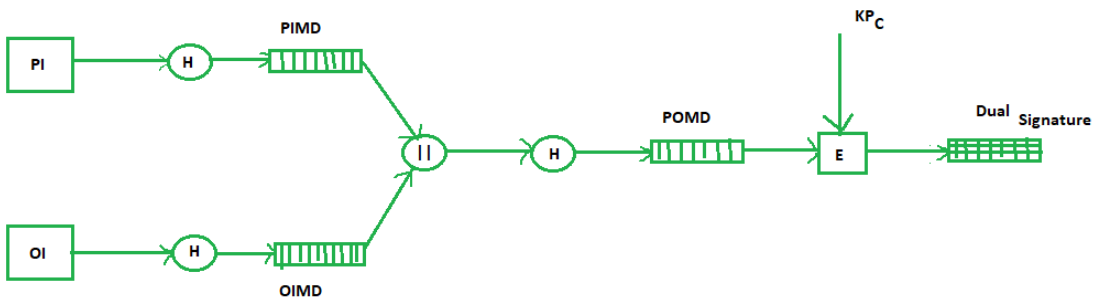
Dual Signature :

The dual signature is a concept introduced with SET, which aims at connecting two information pieces meant for two different receivers :

Order Information (OI) for merchant

Payment Information (PI) for bank

You might think sending them separately is an easy and more secure way, but sending them in a connected form resolves any future dispute possible. Here is the generation of dual signature:



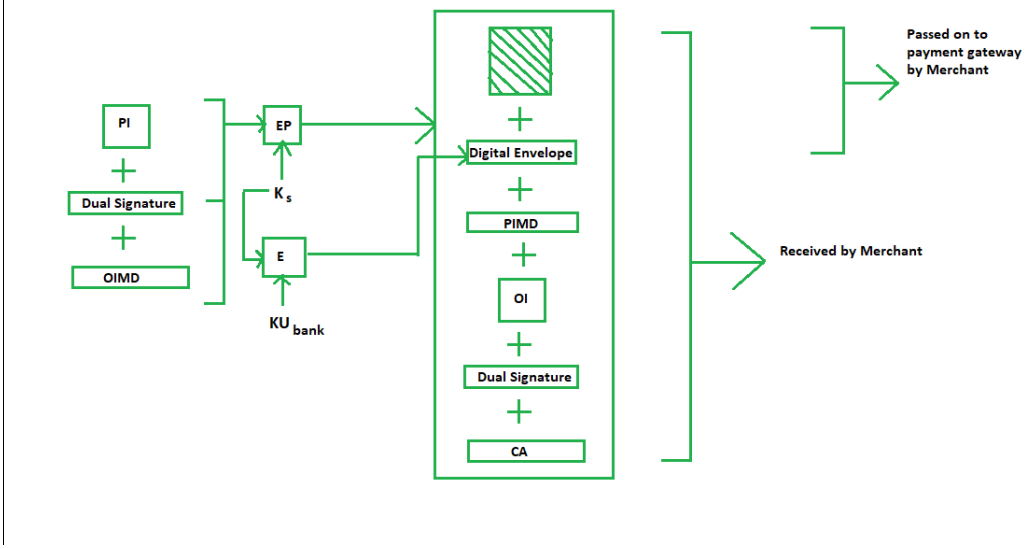

```
Where,

   PI stands for payment information
   OI stands for order information
   PIMD stands for Payment Information Message Digest
   OIMD stands for Order Information Message Digest
   POMD stands for Payment Order Message Digest
   H stands for Hashing
   E stands for public key encryption
   KPc is customer's private key
   || stands for append operation
   Dual signature, DS= E(KPc, [H(H(PI)||H(OI))])
```

Purchase Request Generation :

The process of purchase request generation requires three inputs:

- Payment Information (PI)
- Dual Signature
- Order Information Message Digest (OIMD)

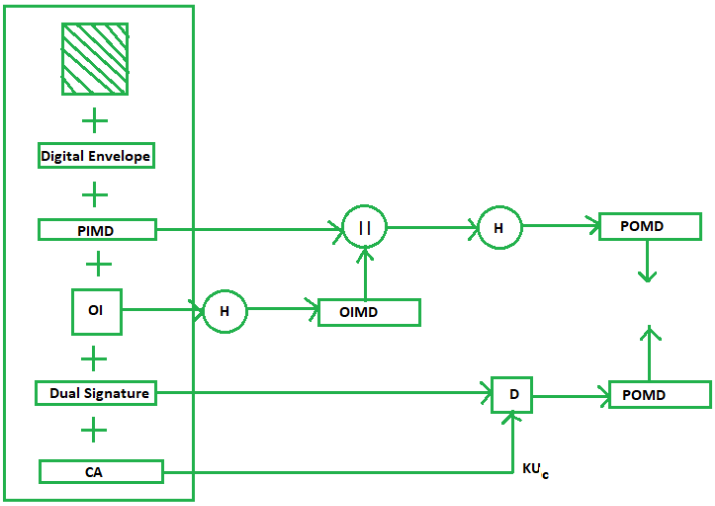
The purchase request is generated as follows:




```
Here,

PI, OIMD, OI all have the same meanings as before.

The new things are :

EP which is symmetric key encryption

Ks is a temporary symmetric key

KUbank is public key of bank

CA is Cardholder or customer Certificate

Digital Envelope = E(KUbank, Ks)
```

Purchase Request Validation on Merchant Side :

The Merchant verifies by comparing POMD generated through PIMD hashing with POMD generated through decryption of Dual Signature as follows:



Since we used Customer private key in encryption here we use KUc which is public key of customer or cardholder for decryption 'D'.

Payment Authorization and Payment Capture :

Payment authorization as the name suggests is the authorization of payment information by merchant which ensures payment will be received by merchant. Payment capture is the process by which merchant receives payment which includes again generating some request blocks to gateway and payment gateway in turn issues payment to merchant.

UNIT -VI

Network Security-II Security at the Network Layer: IPSec,

The popular framework developed for ensuring security at network layer is Internet Protocol Security (IPsec).

Features of IPsec

- IPsec is not designed to work only with TCP as a transport protocol. It works with UDP as well as any other protocol above IP such as ICMP, OSPF etc.
- IPsec protects the entire packet presented to IP layer including higher layer headers.
- Since higher layer headers are hidden which carry port number, traffic analysis is more difficult.
- IPsec works from one network entity to another network entity, not from application process to application process. Hence, security can be adopted without requiring changes to individual user computers/applications.
- Tough widely used to provide secure communication between network entities, IPsec can provide host-to-host security as well.
- The most common use of IPsec is to provide a Virtual Private Network (VPN), either between two locations (gateway-to-gateway) or between a remote user and an enterprise network (host-to-gateway).

Security Functions

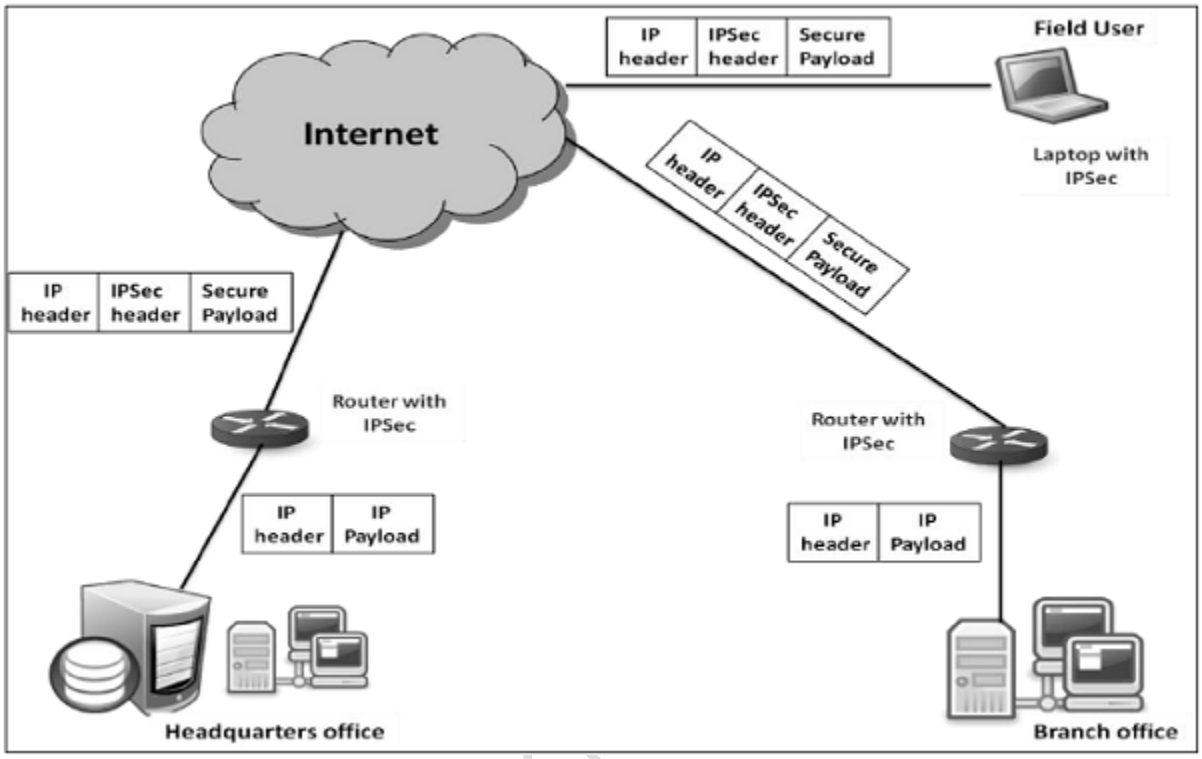
The important security functions provided by the IPsec are as follows −

- Confidentiality
  - Enables communicating nodes to encrypt messages.
  - Prevents eavesdropping by third parties.
- Origin authentication and data integrity.
  - Provides assurance that a received packet was actually transmitted by the party identified as the source in the packet header.
  - Confirms that the packet has not been altered or otherwise.
- Key management.
  - Allows secure exchange of keys.
  - Protection against certain types of security attacks, such as replay attacks.

Virtual Private Network

Ideally, any institution would want its own private network for communication to ensure security. However, it may be very costly to establish and maintain such private network over geographically dispersed area. It would require to manage complex infrastructure of communication links, routers, DNS, etc.

IPsec provides an easy mechanism for implementing Virtual Private Network (VPN) for such institutions. VPN technology allows institution's inter-office traffic to be sent over public Internet by encrypting traffic before entering the public Internet and logically separating it from other traffic. The simplified working of VPN is shown in the following diagram −



Overview of IPsec

IPsec is a framework/suite of protocols for providing security at the IP layer.

Origin

In early 1990s, Internet was used by few institutions, mostly for academic purposes. But in later decades, the growth of Internet became exponential due to expansion of network and several organizations using it for communication and other purposes.

With the massive growth of Internet, combined with the inherent security weaknesses of the TCP/IP protocol, the need was felt for a technology that can provide network security on the Internet. A report entitled "Security in the Internet Architecture" was issued by the Internet Architecture Board (IAB) in 1994. It identified the key areas for security mechanisms.

The IAB included authentication and encryption as essential security features in the IPv6, the next-generation IP. Fortunately, these security capabilities were defined such that they can be implemented with both the current IPv4 and futuristic IPv6.

Security framework, IPsec has been defined in several 'Requests for comments' (RFCs). Some RFCs specify some portions of the protocol, while others address the solution as a whole.

Operations Within IPsec

The IPsec suite can be considered to have two separate operations, when performed in unison, providing a complete set of security services. These two operations are IPsec Communication and Internet Key Exchange.

- IPsec Communication
    - It is typically associated with standard IPsec functionality. It involves encapsulation, encryption, and hashing the IP datagrams and handling all packet processes.
    - It is responsible for managing the communication according to the available Security Associations (SAs) established between communicating parties.
    - It uses security protocols such as Authentication Header (AH) and Encapsulated SP (ESP).
    - IPsec communication is not involved in the creation of keys or their management.
    - IPsec communication operation itself is commonly referred to as IPsec.
- Internet Key Exchange (IKE)
    - IKE is the automatic key management protocol used for IPsec.
    - Technically, key management is not essential for IPsec communication and the keys can be manually managed. However, manual key management is not desirable for large networks.
    - IKE is responsible for creation of keys for IPsec and providing authentication during key establishment process. Though, IPsec can be used for any other key management protocols, IKE is used by default.
    - IKE defines two protocol (Oakley and SKEME) to be used with already defined key management framework Internet Security Association Key Management Protocol (ISAKMP).
    - ISAKMP is not IPsec specific, but provides the framework for creating SAs for any protocol.

This chapter mainly discusses the IPsec communication and associated protocol employed to achieve security.

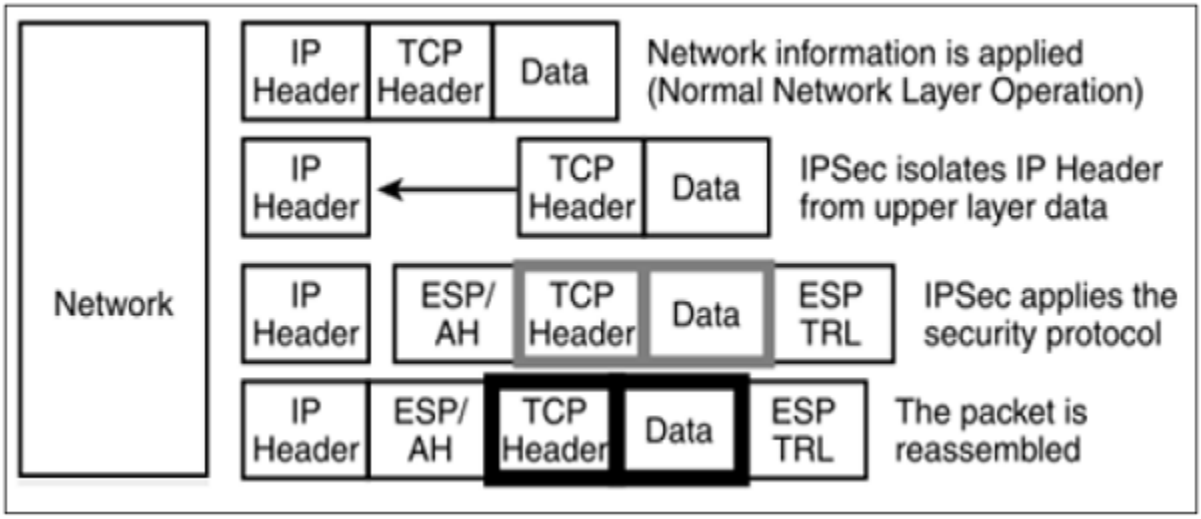
IPsec Communication Modes

IPsec Communication has two modes of functioning; transport and tunnel modes. These modes can be used in combination or used individually depending upon the type of communication desired.

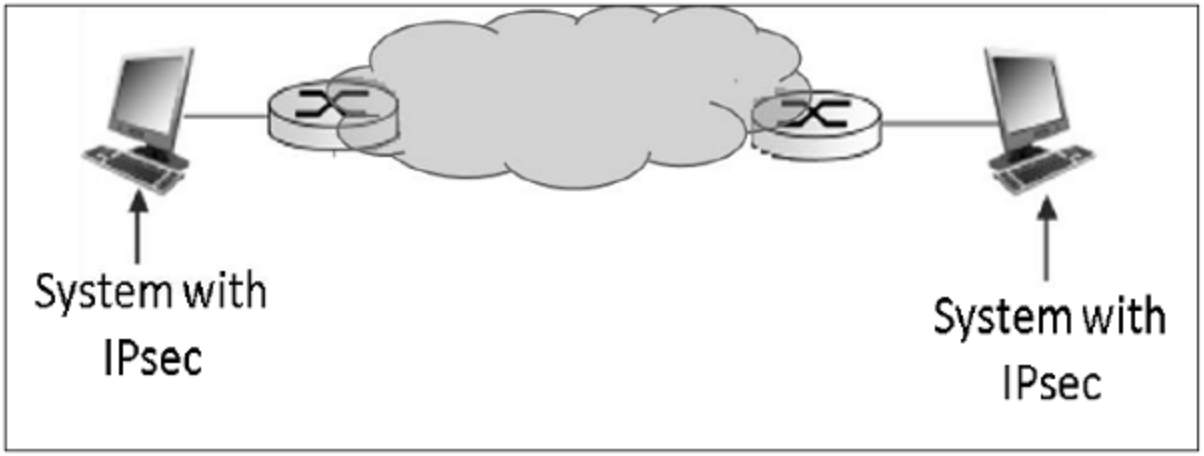
Transport Mode

- IPsec does not encapsulate a packet received from upper layer.
- The original IP header is maintained and the data is forwarded based on the original attributes set by the upper layer protocol.

- The following diagram shows the data flow in the protocol stack.

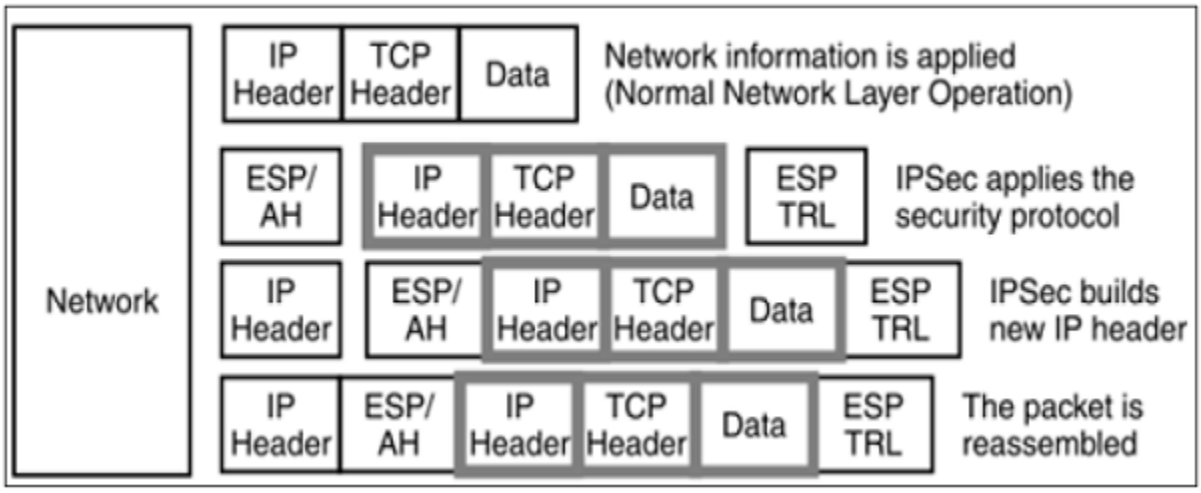


- The limitation of transport mode is that no gateway services can be provided. It is reserved for point-to-point communications as depicted in the following image.

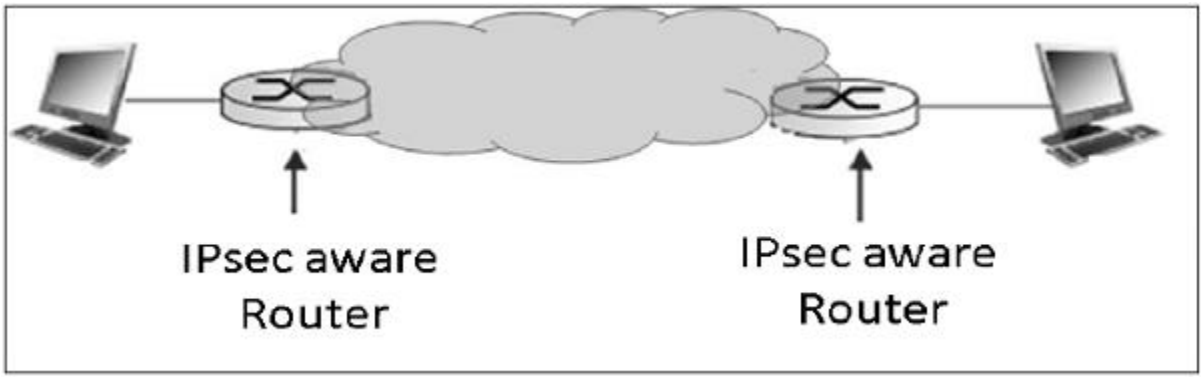


Tunnel Mode

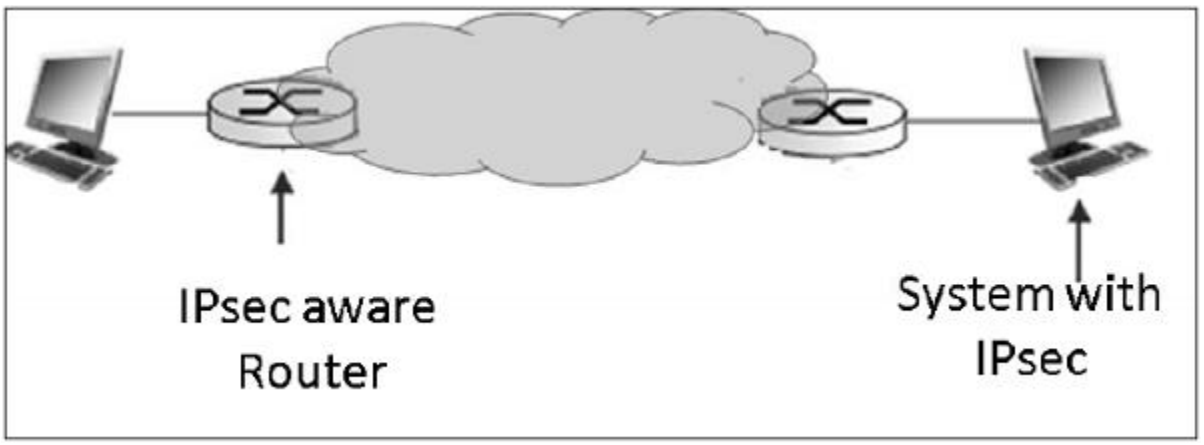
- This mode of IPsec provides encapsulation services along with other security services.
- In tunnel mode operations, the entire packet from upper layer is encapsulated before applying security protocol. New IP header is added.
- The following diagram shows the data flow in the protocol stack.



- Tunnel mode is typically associated with gateway activities. The encapsulation provides the ability to send several sessions through a single gateway.
- The typical tunnel mode communication is as depicted in the following diagram.



- As far as the endpoints are concerned, they have a direct transport layer connection. The datagram from one system forwarded to the gateway is encapsulated and then forwarded to the remote gateway. The remote associated gateway de-encapsulates the data and forwards it to the destination endpoint on the internal network.
- Using IPsec, the tunneling mode can be established between the gateway and individual end system as well.



IPsec Protocols

IPsec uses the security protocols to provide desired security services. These protocols are the heart of IPsec operations and everything else is designed to support these protocol in IPsec.

Security associations between the communicating entities are established and maintained by the security protocol used.

There are two security protocols defined by IPsec — Authentication Header (AH) and Encapsulating Security Payload (ESP).

Authentication Header

The AH protocol provides service of data integrity and origin authentication. It optionally caters for message replay resistance. However, it does not provide any form of confidentiality.

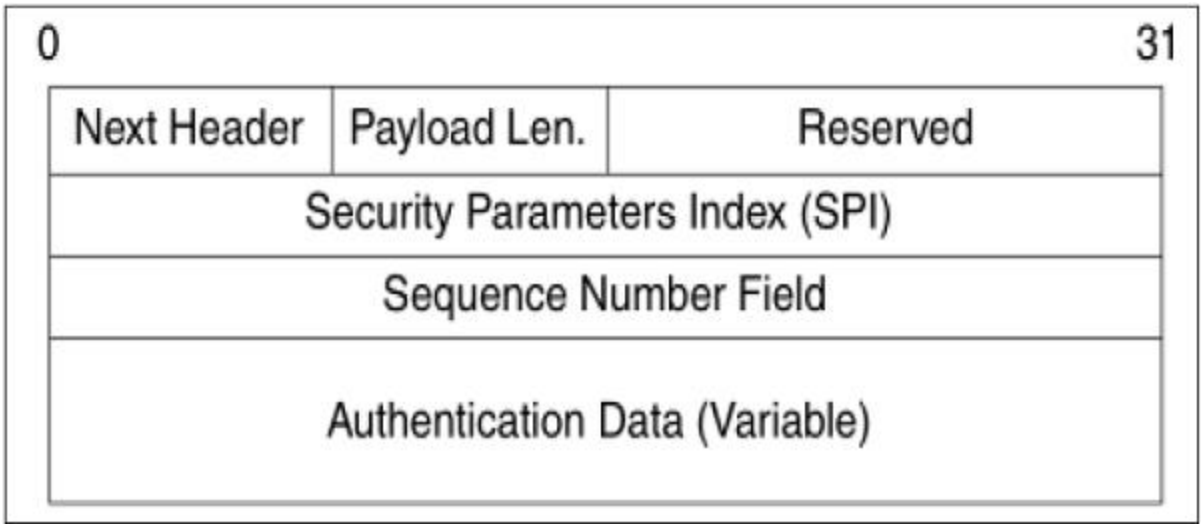
AH is a protocol that provides authentication of either all or part of the contents of a datagram by the addition of a header. The header is calculated based on the values in the datagram. What parts of the datagram are used for the calculation, and where to place the header, depends on the mode cooperation (tunnel or transport).

The operation of the AH protocol is surprisingly simple. It can be considered similar to the algorithms used to calculate checksums or perform CRC checks for error detection.

The concept behind AH is the same, except that instead of using a simple algorithm, AH uses special hashing algorithm and a secret key known only to the communicating parties. A security association between two devices is set up that specifies these particulars.

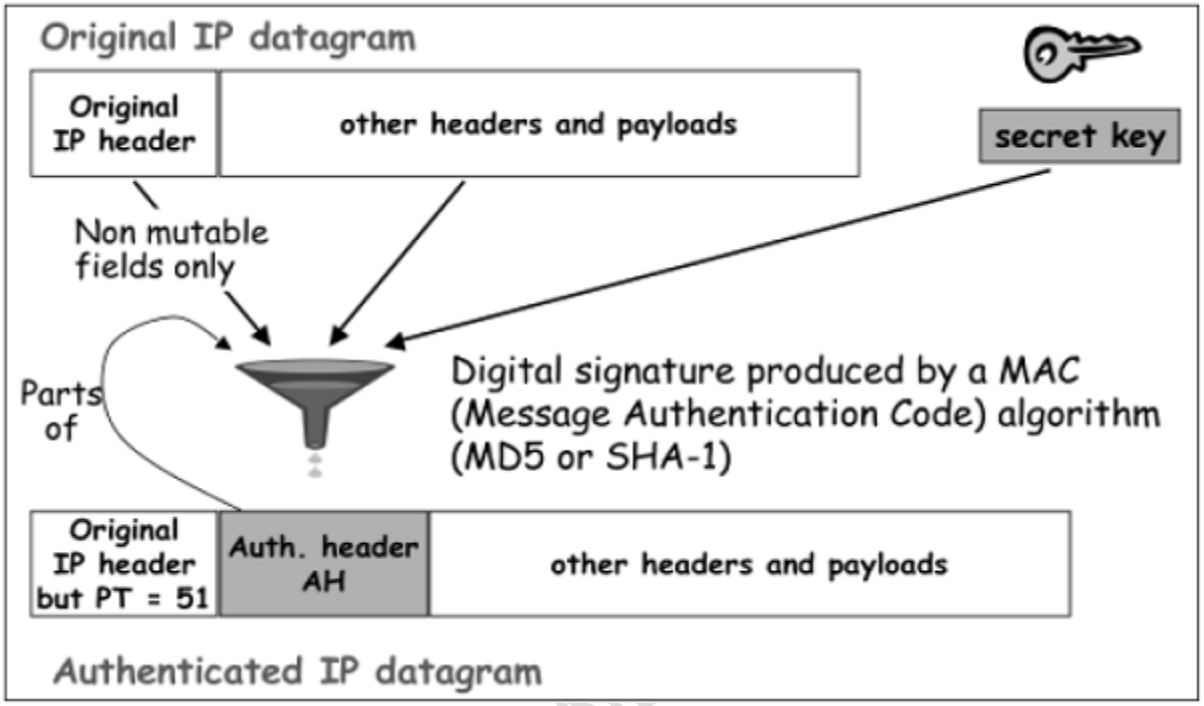
The process of AH goes through the following phases.

- When IP packet is received from upper protocol stack, IPsec determine the associated Security Association (SA) from available information in the packet; for example, IP address (source and destination).
- From SA, once it is identified that security protocol is AH, the parameters of AH header are calculated. The AH header consists of the following parameters −

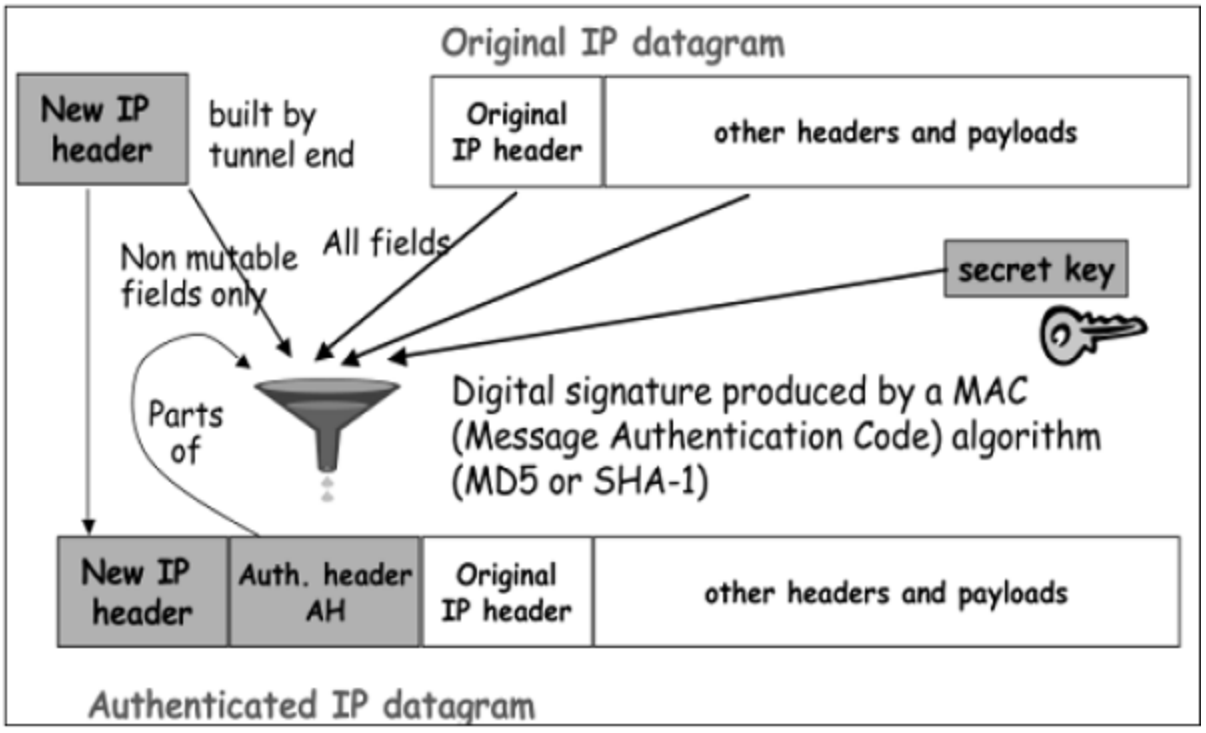


- The header field specifies the protocol of packet following AH header. Sequence Parameter Index (SPI) is obtained from SA existing between communicating parties.
- Sequence Number is calculated and inserted. These numbers provide optional capability to AH to resist replay attack.

- Authentication data is calculated differently depending upon the communication mode.
- In transport mode, the calculation of authentication data and assembling of final IP packet for transmission is depicted in the following diagram. In original IP header, change is made only in protocol number as 51 to indicated application of AH.



- In Tunnel mode, the above process takes place as depicted in the following diagram.



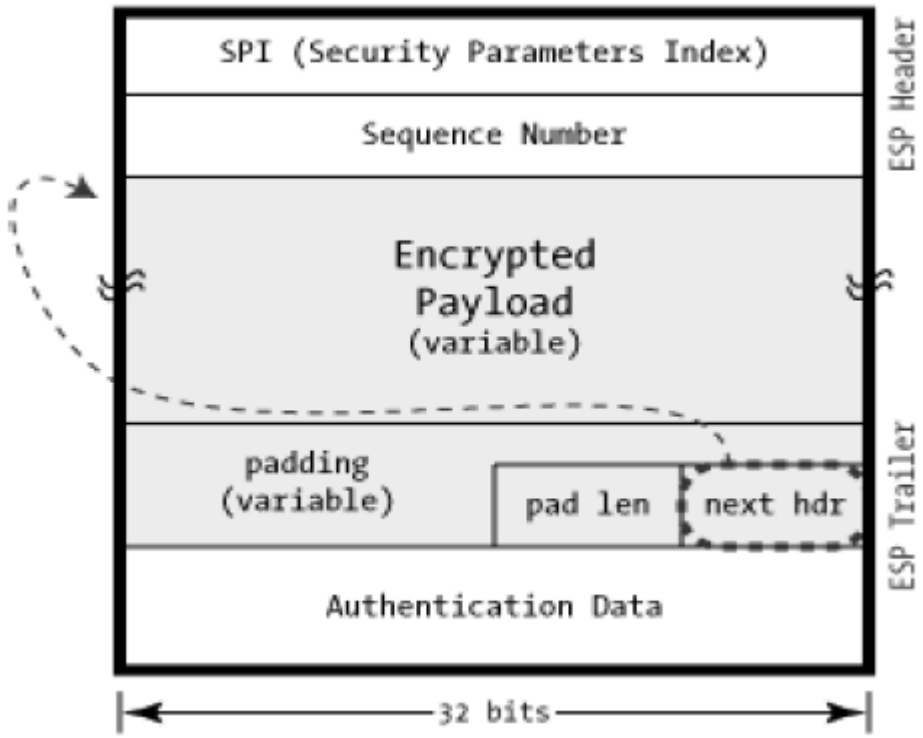
Encapsulation Security Protocol (ESP)

ESP provides security services such as confidentiality, integrity, origin authentication, and optional replay resistance. The set of services provided depends on options selected at the time of Security Association (SA) establishment.

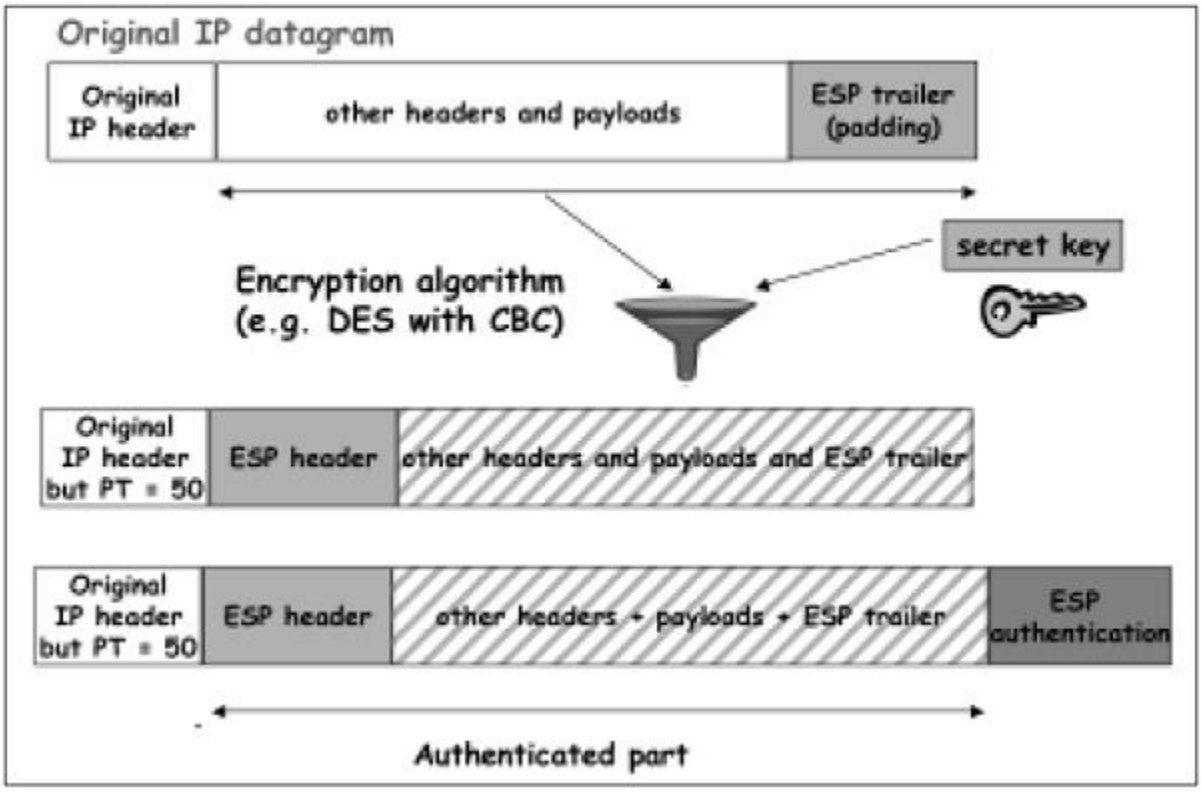
In ESP, algorithms used for encryption and generating authenticator are determined by the attributes used to create the SA.

The process of ESP is as follows. The first two steps are similar to process of AH as stated above.

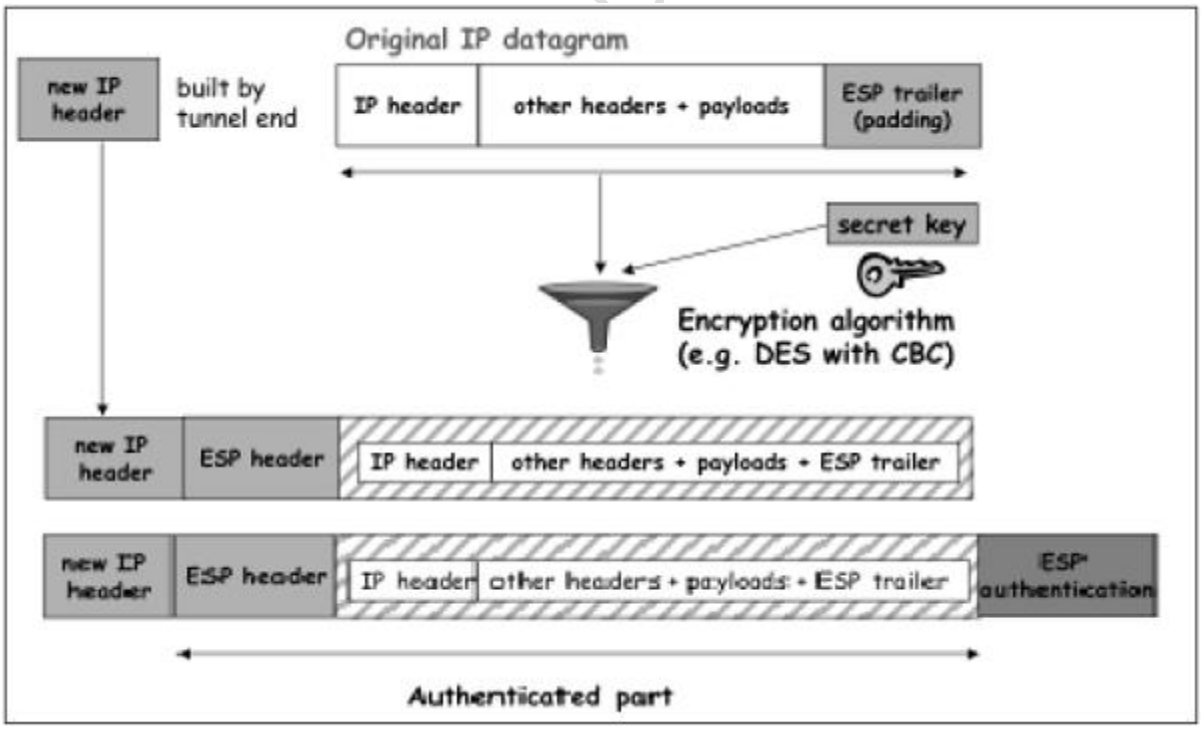
- Once it is determined that ESP is involved, the fields of ESP packet are calculated. The ESP field arrangement is depicted in the following diagram.



- Encryption and authentication process in transport mode is depicted in the following diagram.



- In case of Tunnel mode, the encryption and authentication process is as depicted in the following diagram.



Although authentication and confidentiality are the primary services provided by ESP, both are optional. Technically, we can use NULL encryption without authentication. However, in practice, one of the two must be implemented to use ESP effectively.

The basic concept is to use ESP when one wants authentication and encryption, and to use AH when one wants extended authentication without encryption.

Security Associations in IPsec

Security Association (SA) is the foundation of an IPsec communication. The features of SA are −

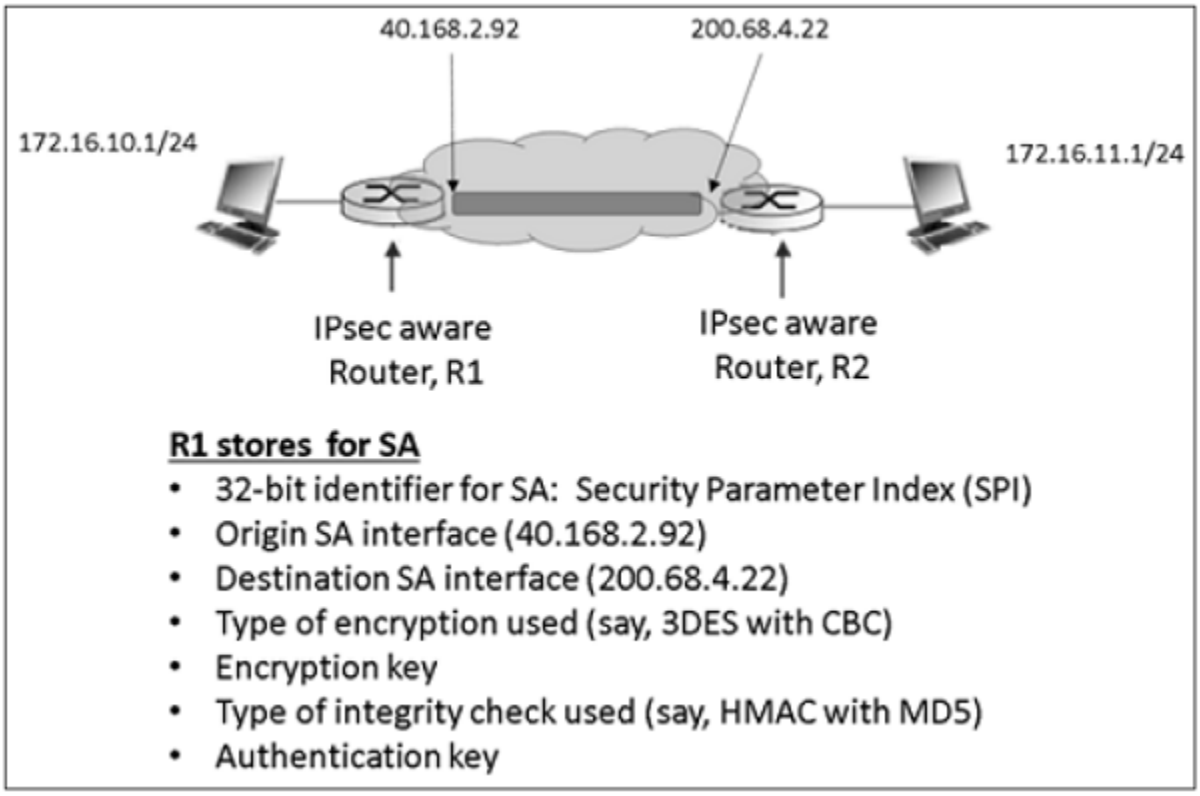
- Before sending data, a virtual connection is established between the sending entity and the receiving entity, called "Security Association (SA)".
- IPsec provides many options for performing network encryption and authentication. Each IPsec connection can provide encryption, integrity, authenticity, or all three services. When the security service is determined, the two IPsec peer entities must determine exactly which algorithms to use (for example, DES or 3DES for encryption; MD5 or SHA-1 for integrity). After deciding on the algorithms, the two devices must share session keys.
- SA is a set of above communication parameters that provides a relationship between two or more systems to build an IPsec session.
- SA is simple in nature and hence two SAs are required for bi-directional communications.
- SAs are identified by a Security Parameter Index (SPI) number that exists in the security protocol header.
- Both sending and receiving entities maintain state information about the SA. It is similar to TCP endpoints which also maintain state information. IPsec is connection-oriented like TCP.

Parameters of SA

Any SA is uniquely identified by the following three parameters −

- Security Parameters Index (SPI).
    - It is a 32-bit value assigned to SA. It is used to distinguish among different SAs terminating at the same destination and using the same IPsec protocol.
    - Every packet of IPsec carries a header containing SPI field. The SPI is provided to map the incoming packet to an SA.
    - The SPI is a random number generated by the sender to identify the SA to the recipient.
- **Destination IP Address** − It can be IP address of end router.
- **Security Protocol Identifier** − It indicates whether the association is an AH or ESP SA.

Example of SA between two router involved in IPsec communication is shown in the following diagram.



Security Administrative Databases

In IPsec, there are two databases that control the processing of IPsec datagram. One is the Security Association Database (SAD) and the other is the Security Policy Database (SPD). Each communicating endpoint using IPsec should have a logically separate SAD and SPD.

Security Association Database

In IPsec communication, endpoint holds SA state in Security Association Database (SAD). Each SA entry in SAD database contains nine parameters as shown in the following table −

Sr.No.	Parameters & Description
1	**Sequence Number Counter** For outbound communications. This is the 32-bit sequence number provided in the AH or ESP headers.
2	**Sequence Number Overflow Counter** Sets an option flag to prevent further communications utilizing the specific SA
3	**32-bit anti-replay window** Used to determine whether an inbound AH or ESP packet is a replay

4	**Lifetime of the SA** Time till SA remain active
5	**Algorithm - AH** Used in the AH and the associated key
6	**Algorithm - ESP Auth** Used in the authenticating portion of the ESP header
7	**Algorithm - ESP Encryption** Used in the encryption of the ESP and its associated key information
8	**IPsec mode of operation** Transport or tunnel mode
9	**Path MTU(PMTU)** Any observed path maximum transmission unit (to avoid fragmentation)

All SA entries in the SAD are indexed by the three SA parameters: Destination IP address, Security Protocol Identifier, and SPI.

Security Policy Database

SPD is used for processing outgoing packets. It helps in deciding what SAD entries should be used. If no SAD entry exists, SPD is used to create new ones.

Any SPD entry would contain −

- Pointer to active SA held in SAD.
- Selector fields – Field in incoming packet from upper layer used to decide application of IPsec. Selectors can include source and destination address, port numbers if relevant, application IDs, protocols, etc.

Outgoing IP datagrams go from the SPD entry to the specific SA, to get encoding parameters. Incoming IPsec datagram get to the correct SA directly using the SPI/DEST IP/Protocol triple, and from there extracts the associated SAD entry.

SPD can also specify traffic that should bypass IPsec. SPD can be considered as a packet filter where the actions decided upon are the activation of SA processes.

System Security

INTRUDERS

One of the most publicized attacks to security is the intruder, generally referred to as hacker or cracker. Three classes of intruders are as follows:

- **Masquerader** – an individual who is not authorized to use the computer and who penetrates a system‟s access controls to exploit a legitimate user‟s account.

- **Misfeasor** – a legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuse his or her privileges.

- **Clandestine user** – an individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

The masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider.

Intruder attacks range from the benign to the serious. At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there. At the serious end are individuals who are attempting to read privileged data, perform unauthorized modifications to data, or disrupt the system. Benign intruders might be tolerable, although they do consume resources and may slow performance for legitimate users. However there is no way in advance to know whether an intruder will be benign or malign.

An analysis of previous attack revealed that there were two levels of hackers:

- The high levels were sophisticated users with a thorough knowledge of the technology.

- The low levels were the „foot soldiers‟ who merely use the supplied cracking programs with little understanding of how they work.

one of the results of the growing awareness of the intruder problem has been the establishment of a number of Computer Emergency Response Teams (CERT). these co-operative ventures collect information about system vulnerabilities and disseminate it to systems managers. Unfortunately, hackers can also gain access to CERT reports.

In addition to running password cracking programs, the intruders attempted to modify login software to enable them to capture passwords of users logging onto the systems.

Intrusion techniques

The objective of the intruders is to gain access to a system or to increase the range of privileges accessible on a system. Generally, this requires the intruders to acquire information that should be protected. In most cases, the information is in the form of a user password.

Typically, a system must maintain a file that associates a password with each authorized user. If such a file is stored with no protection, then it is an easy matter to gain access to it. The password files can be protected in one of the two ways:

- **One way encryption** – the system stores only an encrypted form of user‟s password. In practice, the system usually performs a one way transformation (not reversible) in which the password is used to generate a key for the encryption function and in which a fixed length output is produced.

- **Access control** – access to the password file is limited to one or a very few accounts.

The following techniques are used for learning passwords.

- Try default passwords used with standard accounts that are shipped with the system. Many administrators do not bother to change these defaults.

- Exhaustively try all short passwords.

- Try words in the system‟s online dictionary or a list of likely passwords.

- Collect information about users such as their full names, the name of their spouse and children, pictures in their office and books in their office that are related to hobbies.

- Try user‟s phone number, social security numbers and room numbers.

- Try all legitimate license plate numbers.

- Use a torjan horse to bypass restriction on access.

- Tap the line between a remote user and the host system.

Two principle countermeasures:

Detection – concerned with learning of an attack, either before or after its success.Θ

Prevention – challenging security goal and an uphill bottle at all times.

INTRUSION DETECTION:

Inevitably, the best intrusion prevention system will fail. A system's second line of defense is intrusion detection, and this has been the focus of much research in recent years. This interest is motivated by a number of considerations, including the following:

- If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised.

- An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions.

· Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified.

Figure 5.2.1 suggests, in very abstract terms, the nature of the task confronting the designer of an intrusion detection system. Although the typical behavior of an intruder differs from the typical behavior of an authorized user, there is an overlap in these behaviors. Thus, a loose interpretation of intruder behavior, which will catch more intruders, will also lead to a number of "false positives," or authorized users identified as intruders. On the other hand, an attempt to limit false positives by a tight interpretation of intruder behavior will lead to an increase in false negatives, or intruders not identified as intruders. Thus, there is an element of compromise and art in the practice of intrusion detection.

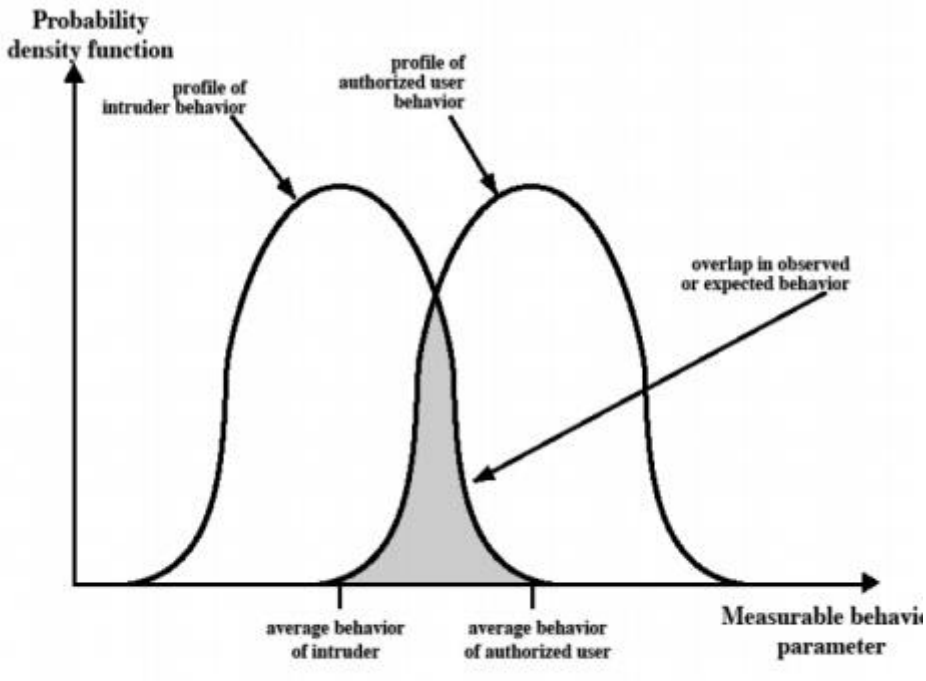


Fig. 5.2.1Profilesofbehavior of intruders and authorized users

1. The approaches to intrusion detection:

Statistical anomaly detection: Involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed

behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

Threshold detection: This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.

Profile based: A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

j) **Rule-based detection**: Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.

Anomaly detection: Rules are developed to detect deviation from previous usage patterns.

Penetration identification: An expert system approach that searches for suspicious behavior.

In terms of the types of attackers listed earlier, statistical anomaly detection is effective against masqueraders. On the other hand, such techniques may be unable to deal with misfeasors. For such attacks, rule-based approaches may be able to recognize events and sequences that, in context, reveal penetration. In practice, a system may exhibit a combination of both approaches to be effective against a broad range of attacks.

System Audit

It is an investigation to review the performance of an operational system. The objectives of conducting a system audit are as follows −

- To compare actual and planned performance.
- To verify that the stated objectives of system are still valid in current environment.
- To evaluate the achievement of stated objectives.
- To ensure the reliability of computer based financial and other information.
- To ensure all records included while processing.

- To ensure protection from frauds.

Audit of Computer System Usage

Data processing auditors audits the usage of computer system in order to control it. The auditor need control data which is obtained by computer system itself.

The System Auditor

The role of auditor begins at the initial stage of system development so that resulting system is secure. It describes an idea of utilization of system that can be recorded which helps in load planning and deciding on hardware and software specifications. It gives an indication of wise use of the computer system and possible misuse of the system.

Audit Trial

An audit trial or audit log is a security record which is comprised of who has accessed a computer system and what operations are performed during a given period of time. Audit trials are used to do detailed tracing of how data on the system has changed.

It provides documentary evidence of various control techniques that a transaction is subject to during its processing. Audit trials do not exist independently. They are carried out as a part of accounting for recovering lost transactions.

Audit Methods

Auditing can be done in two different ways −

Auditing around the Computer

- Take sample inputs and manually apply processing rules.
- Compare outputs with computer outputs.

Auditing through the Computer

- Establish audit trial which allows examining selected intermediate results.
- Control totals provide intermediate checks.

Audit Considerations

Audit considerations examine the results of the analysis by using both the narratives and models to identify the problems caused due to misplaced functions, split processes or functions, broken data flows, missing data, redundant or incomplete processing, and nonaddressed automation opportunities.

The activities under this phase are as follows −

- Identification of the current environment problems
- Identification of problem causes
- Identification of alternative solutions

- Evaluation and feasibility analysis of each solution
- Selection and recommendation of most practical and appropriate solution
- Project cost estimation and cost benefit analysis
- **Distributed Intrusion Detection**
-
- Until recently, work on intrusion detection systems focused on single-system stand-alone facilities. The typical organization, however, needs to defend a distributed collection of hosts supported by a LAN Porras points out the following major issues in the design of a distributed intrusion detection system
-
- A distributed intrusion detection system may need to deal with different audit record formats. In a heterogeneous environment, different systems will employ different native audit collection systems and, if using intrusion detection, may employ different formats for security-related audit records.
-
- One or more nodes in the network will serve as collection and analysis points for the data from the systems on the network. Thus, either raw audit data or summary data must be transmitted across the network. Therefore, there is a requirement to assure the integrity and confidentiality of these data.
-
- Either a centralized or decentralized architecture can be used.
-
- Below figure shows the overall architecture, which consists of three main components:
-
- · **Host agent module:** An audit collection module operating as a background process on a monitored system. Its purpose is to collect data on security-related events on the host and transmit these to the central manager.
- ·
- · **LAN monitor agent module:** Operates in the same fashion as a host agent module except that it analyzes LAN traffic and reports the results to the central manager.
- ·
-
- · **Central manager module:** Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion.

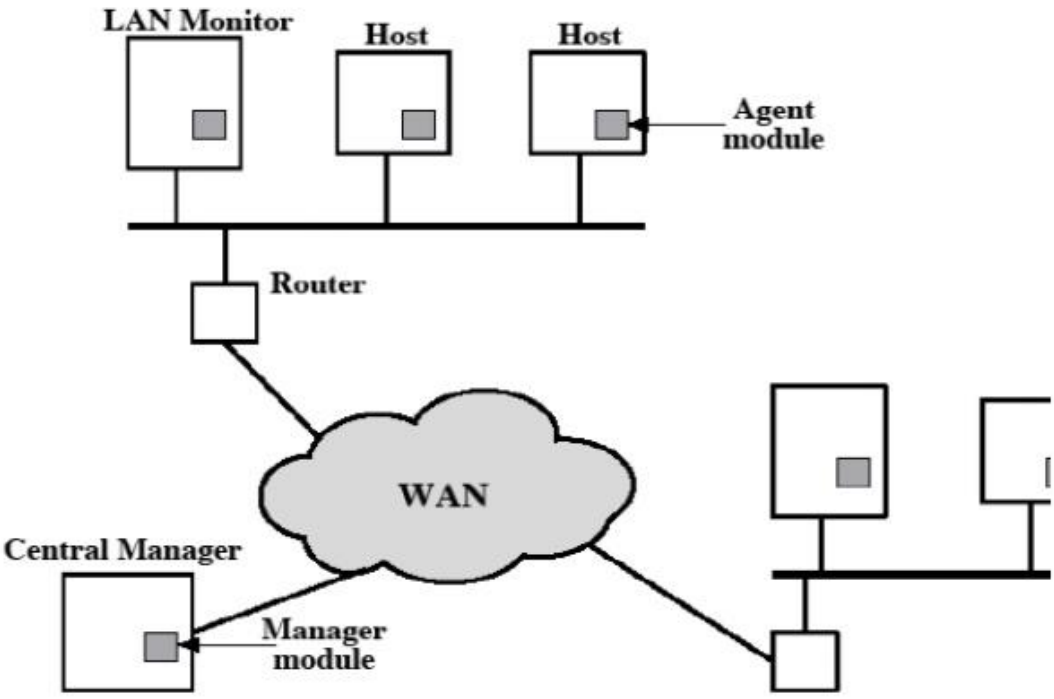
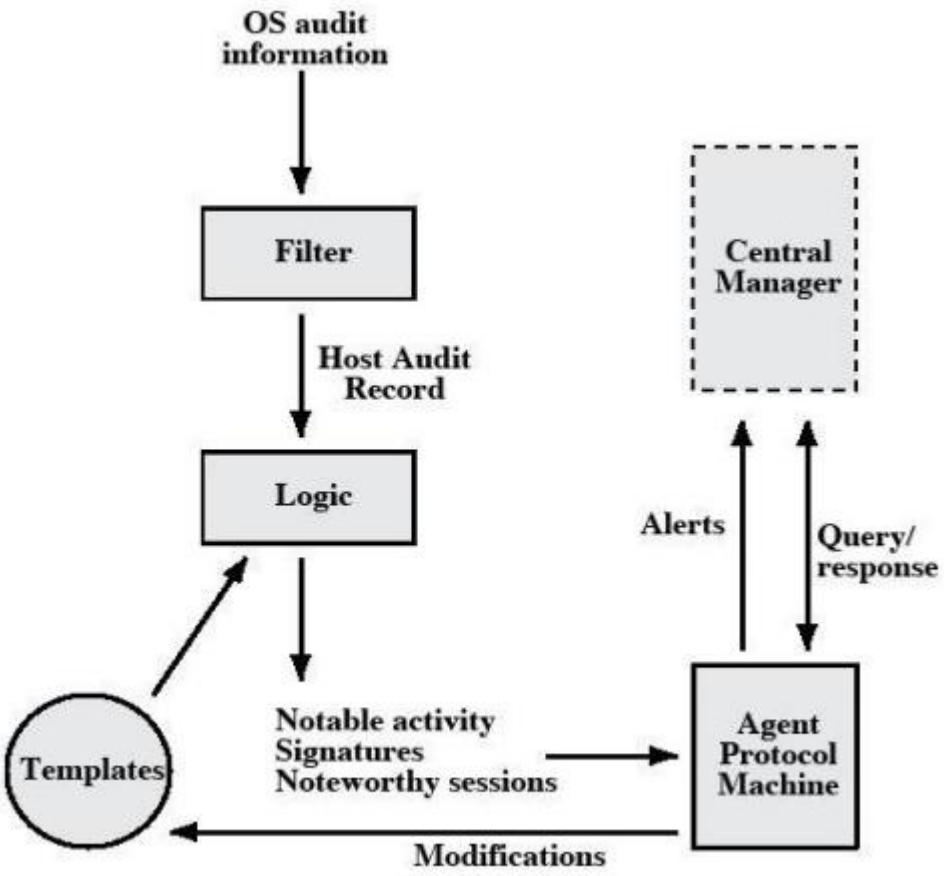


Fig. 5.2.3.1 Architecture of Distributed intrusion detection

- **Honeypots**
- A relatively recent innovation in intrusion detection technology is the honeypot. Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems. Honeypots are designed to
- · divert an attacker from accessing critical systems
- · collect information about the attacker's activity
- · encourage the attacker to stay on the system long enough for administrators to respond



- These systems are filled with fabricated information designed to appear valuable but that a legitimate user of the system wouldn't access. Thus, any access to the honeypot is suspect.
- **FIREWALLS**
- **1. Firewall design principles**
- Internet connectivity is no longer an option for most organizations. However, while internet access provides benefits to the organization, it enables the outside world to reach and interact with local network assets. This creates the threat to the organization. While it is possible to equip each workstation and server on the premises network with strong security features, such as intrusion protection, this is not a practical approach. The alternative, increasingly accepted, is the firewall.
- The firewall is inserted between the premise network and internet to establish a controlled link and to erect an outer security wall or perimeter. The aim of this perimeter is to protect the premises network from internet based attacks and to provide a single choke point where security and audit can be imposed. The firewall

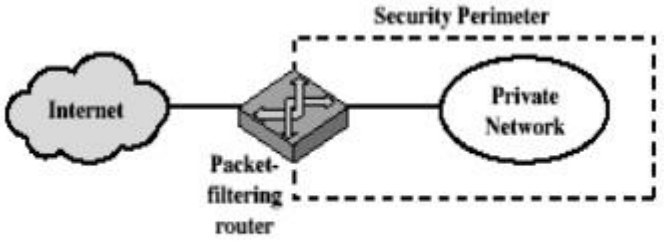
can be a single computer system or a set of two or more systems that cooperate to perform the firewall function.

- **2. Firewall characteristics:**
- · 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.
- · Various configurations are possible.
- ·
- · 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.
- ·
- · The firewall itself is immune to penetration. This implies that use of a trusted system with a secure operating system. This implies that use of a trusted system with a secure operating system.
- Four techniques that firewall use to control access and enforce the site‟s security policy is as follows:
- 1. Service control – determines the type of internet services that can be accessed, inbound or outbound. The firewall may filter traffic on this basis of IP address and TCP port number; may provide proxy software that receives and interprets each service request before passing it on; or may host the server software itself, such as web or mail service.
- 2. Direction control – determines the direction in which particular service request may be initiated and allowed to flow through the firewall.
- 3. User control – controls access to a service according to which user is attempting to access it.
- 4. Behavior control – controls how particular services are used.
- **Capabilities of firewall**
- A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving

the network, and provides protection from various kinds of IP spoofing and routing attacks.

- A firewall provides a location for monitoring security related events. Audits and alarms can be implemented on the firewall system.
- A firewall is a convenient platform for several internet functions that are not security related.
- A firewall can serve as the platform for IPsec.
- **3. Limitations of firewall**
- · 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.
- · The firewall does not protect against internal threats. The firewall does not protect against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.
- · The firewall cannot protect against the transfer of virus-infected programs or files. Because of the variety of operating systems and applications supported inside the perimeter, it would be impractical and perhaps impossible for the firewall to scan all incoming files, e-mail, and messages for viruses.
- **4 Types of firewalls**
- There are 3 common types of firewalls.
- · Packet filters
- ·
- · Application-level gateways
- ·
- · Circuit-level gateways
- **Packet filtering router**
- A packet filtering router applies a set of rules to each incoming IP packet and then forwards or discards the packet. The router is typically configured to filter packets going in both directions. Filtering rules are based on the information contained in a network packet:

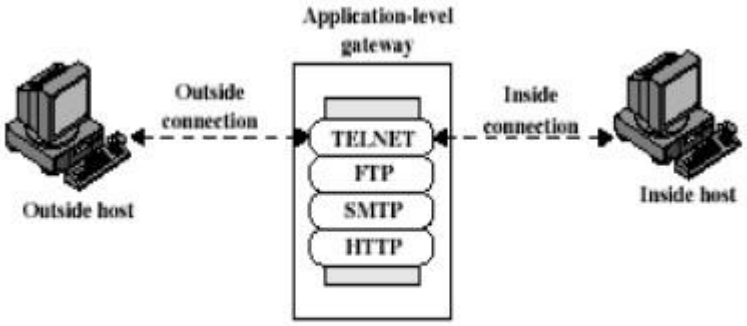
- · Source IP address – IP address of the system that originated the IP packet. Destination IP address – IP address of the system, the IP is trying to reach. Source and destination transport level address – transport level port number. IP protocol field – defines the transport protocol.
- · Interface – for a router with three or more ports, which interface of the router the packet come from or which interface of the router the packet is destined for.



(a) Packet-filtering router

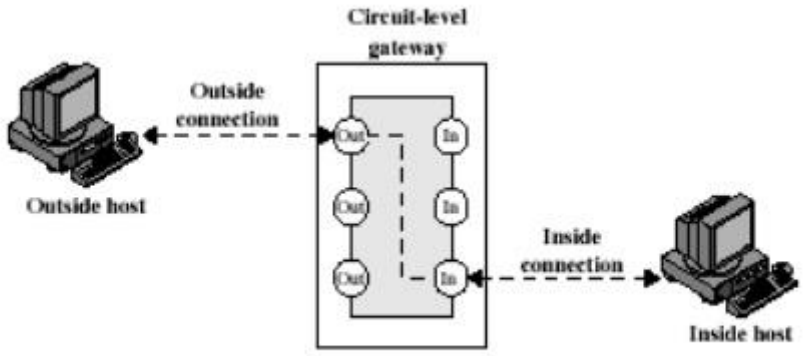
- The packet filter is typically set up as a list of rules based on matches to fields in the IP or TCP header. If there is a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet. If there is no match to any rule, then a default action is taken.
- Two default policies are possible:
- · Default = discard: That which is not expressly permitted is prohibited.
- · Default = forward: That which is not expressly prohibited is permitted.
- The default discard policy is the more conservative. Initially everything is blocked, and services must be added on a case-by-case basis. This policy is more visible to users, who are most likely to see the firewall as a hindrance. The default forward policy increases ease of use for end users but provides reduced security.
- **5. Advantages of packet filter router**
- · Simple
- · Transparent to users
- Very fast
- **Weakness of packet filter firewalls**

- · Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application specific vulnerabilities or functions.
- . Because of the limited information available to the firewall, the logging functionality present in packet filter firewall is limited.
- . It does not support advanced user authentication schemes.
- · They are generally vulnerable to attacks such as layer address spoofing.
- Some of the attacks that can be made on packet filtering routers and the appropriate counter measures are the following:
- · IP address spoofing – the intruders transmit packets from the outside with a source IP address field containing an address of an internal host.
- Countermeasure: to discard packet with an inside source address if the packet arrives on an external interface.
- · Source routing attacks – the source station specifies the route that a packet should take as it crosses the internet; i.e., it will bypass the firewall.
- · Tiny fragment attacks – the intruder create extremely small fragments and force the TCP header information into a separate packet fragment. The attacker hopes that only the first fragment is examined and the remaining fragments are passed through.
- Countermeasure: to discard all packets where the protocol type is TCP and the IP fragment offset is equal to 1.
- **6. Application level gateway**
- An Application level gateway, also called a proxy server, acts as a relay of application level traffic. The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints.
- Application level gateways tend to be more secure than packet filters. It is easy to log and audit all incoming traffic at the application level. A prime disadvantage is the additional processing overhead on each connection.



(b) Application-level gateway

-
-
- **7 Circuit level gateway**
-
- Circuit level gateway can be a stand-alone system or it can be a specified function performed by an application level gateway for certain applications. A Circuit level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outer host. Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.
-
- A typical use of Circuit level gateways is a situation in which the system administrator trusts the internal users. The gateway can be configured to support application level or proxy service on inbound connections and circuit level functions for outbound connections.

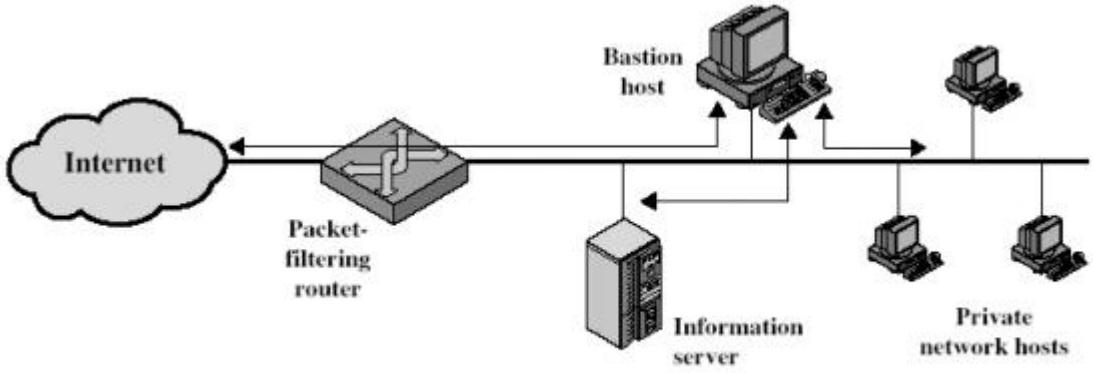


(c) Circuit-level gateway

-
-
- **Basiton host**
-
- It is a system identified by the firewall administrator as a critical strong point in the network‟s security. The Bastion host serves as a platform for an application level and circuit level gateway.
-

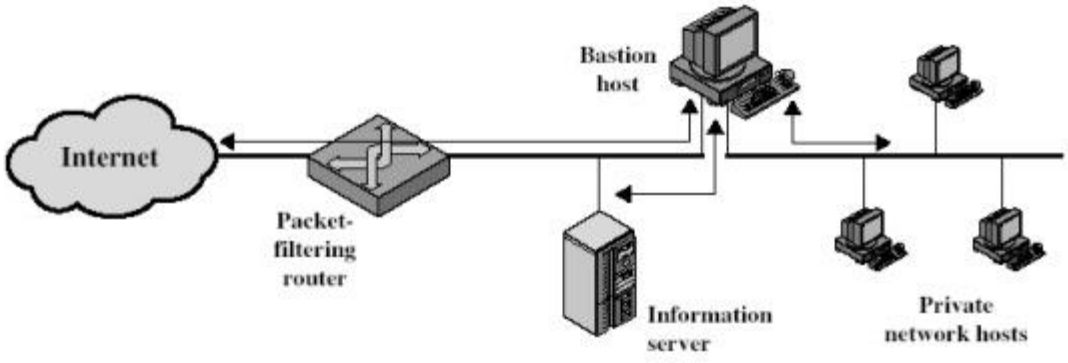
- Common characteristics of a Basiton host are as follows:
-
- · The Bastion host hardware platform executes a secure version of its operating system, making it a trusted system.
- · Only the services that the network administrator considers essential are installed on the Bastion host.
-
- · It may require additional authentication before a user is allowed access to the proxy services.
-
- · Each proxy is configured to support only a subset of standard application‟s command set.
- · Each proxy is configured to allow access only to specific host systems.
-
- · Each proxy maintains detailed audit information by logging all traffic, each connection
- · and the duration of each connection.
-
- · 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 on a non privileged user in a private and secured directory on the Bastion host.
- **Firewall configurations**
-
- There are 3 common firewall configurations.
- 1. Screened host firewall, single-homed basiton configuration
- 2. Screened host firewall, dual homed basiton configuration
- 3. Screened subnet firewall configuration
-
-
- **1. Screened host firewall, single-homed basiton configuration**
-
- In this configuration, the firewall consists of two systems: a packet filtering router and a bastion host. Typically, the router is configured so that
-
- o For traffic from the internet, only IP packets destined for the basiton host are allowed in.
-
- o For traffic from the internal network, only IP packets from the basiton host are allowed out.

-
- The basiton host performs authentication and proxy functions. This configuration has greater security than simply a packet filtering router or an application level gateway alone, for two reasons:
- · This configuration implements both packet level and application level filtering, allowing for considerable flexibility in defining security policy.
-
- · An intruder must generally penetrate two separate systems before the security of the internal network is compromised.
-



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

-
-
- **2. Screened host firewall, dual homed basiton configuration**
-
- In the previous configuration, if the packet filtering router is compromised, traffic could flow directly through the router between the internet and the other hosts on the private network. This configuration physically prevents such a security break.



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

-
-
- **3. Screened subnet firewall configuration**
-
- In this configuration, two packet filtering routers are used, one between the basiton host and internet and one between the basiton host and the internal network. This

configuration creates an isolated subnetwork, which may consist of simply the basiton host but may also include one or more information servers and modems for dial-in capability. Typically both the internet and the internal network have access to hosts on the screened subnet, but traffic across the screened subnet is blocked. This configuration offers several advantages:

- · There are now three levels of defense to thwart intruders.
-
- · The outside router advertises only the existence of the screened subnet to the internet; therefore the internal network is invisible to the internet.
-
- · Similarly, the inside router advertises only the existence of the screened subnet to the internal network; therefore the systems on the internal network cannot construct direct routes to the internet.