

MCA-20301 INFORMATION SECURITY AND CRYPTOGRAPHY

Instruction: 4 Periods/week

Time: 3 Hours

Credits:4

Internal: 25 Marks

External: 75 Marks

Total: 100 Marks

UNIT I

Introduction: The need for security-security approaches-principles of security-Plain Text and Cipher Text-substitution and Transposition Techniques-Encryption and Decryption- Symmetric and Asymmetric Cryptography-Stenography-key range and key size-types of attacks.

Number Theory: Introduction to number theory- Modular Arithmetic, Euclidean algorithm, Euler theorem, Fermat Theorem, Totient Function, Multiplicative and Additive Inverse.

UNIT II

Symmetric Key Cryptographic Algorithms: Algorithm types and modes-overview of symmetric key cryptography – DES – IDEA – Blowfish – AES-Differential and Linear Cryptanalysis.

Asymmetric Key Cryptographic Algorithms: Overview of asymmetric key cryptography-RSA algorithm-symmetric and asymmetric key cryptography together-digital signatures.

UNIT III

User Authentication Mechanisms: Introduction-Authentication basics – passwords-authentication tokens-certificate based authentication-biometrics Authentication-Hash functions-SHA1.

System Security: Intruders, Viruses, Related Threats, Trusted Systems.

UNIT IV

Internet Security Protocols: Basic concepts-SSL-SHTTP-TSP-SET-SSL versus SET- 3D secure Protocol-Electronic Money-Email security-WAP security-security in GSM.

Network Security: Brief Introduction to TCP/IP -Firewalls -IP Security-Virtual Private Networks.

Text Books:

1. Cryptography and Network security, AtulKahate, Tata McGraw-Hill Pub companyLtd., NewDelhi
2. Network Security Essentials Applications and Standards, William Stallings, Pearson Education, New Delhi

Reference Books:

1. Network Security Private Communication in a public world, Charlie Kaufman, Radia Perlman & Mike Speciner, Prentice Hall of India Private Ltd., New Delhi
2. Network Security: The Complete Reference by Roberta Bragg, Mark Phodes -Ousley, Keith Strass berg TataMcGraw-Hill.

UNIT-1

INTRODUCTION

What is Information Security?

Information Security is not only about securing information from unauthorized access. Information Security is basically the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information. Information can be physical or electronic one. Information can be anything like your details or we can say your profile on social media, your data in mobile phone, your biometrics etc.

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 prevent 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.

To assess the security needs of an organization effectively, the manager responsible for security needs some systematic way of defining the requirements for security and characterization of approaches to satisfy those requirements. One approach is to consider three aspects of information security:

- Security Attack: Any action that compromises the security of information owned by an organization.
- Security Mechanism: A mechanism that is designed to detect, prevent or recover from a security attack.

- **Security Service:** A service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks and they make use of one or more security mechanisms to provide the service.

The need for Security

Information Security refers to the processes and methodologies which are designed and implemented to protect print, electronic, or any other form of confidential, private and sensitive information or data from unauthorized access, use, misuse, disclosure, destruction, modification, or disruption.

The network needs security against attackers and hackers. Network Security includes two basic securities. The first is the security of data information i.e. to protect the information from unauthorized access and loss. And the second is computer security i.e. to protect data and to thwart hackers.

Information system means to consider available countermeasures or controls stimulated through uncovered vulnerabilities and identify an area where more work is needed. The purpose of data security management is to make sure business continuity and scale back business injury by preventing and minimizing the impact of security incidents. The basic principle of Information Security is:

1. Confidentially
2. Authentication
3. Non-Repudiation
4. Integrity

The need for Information security:

- Protecting the functionality of the organization
- Enabling the safe operation of applications
- Protecting the data that the organization collect and use
- Safeguarding technology assets in organizations

Security Approaches

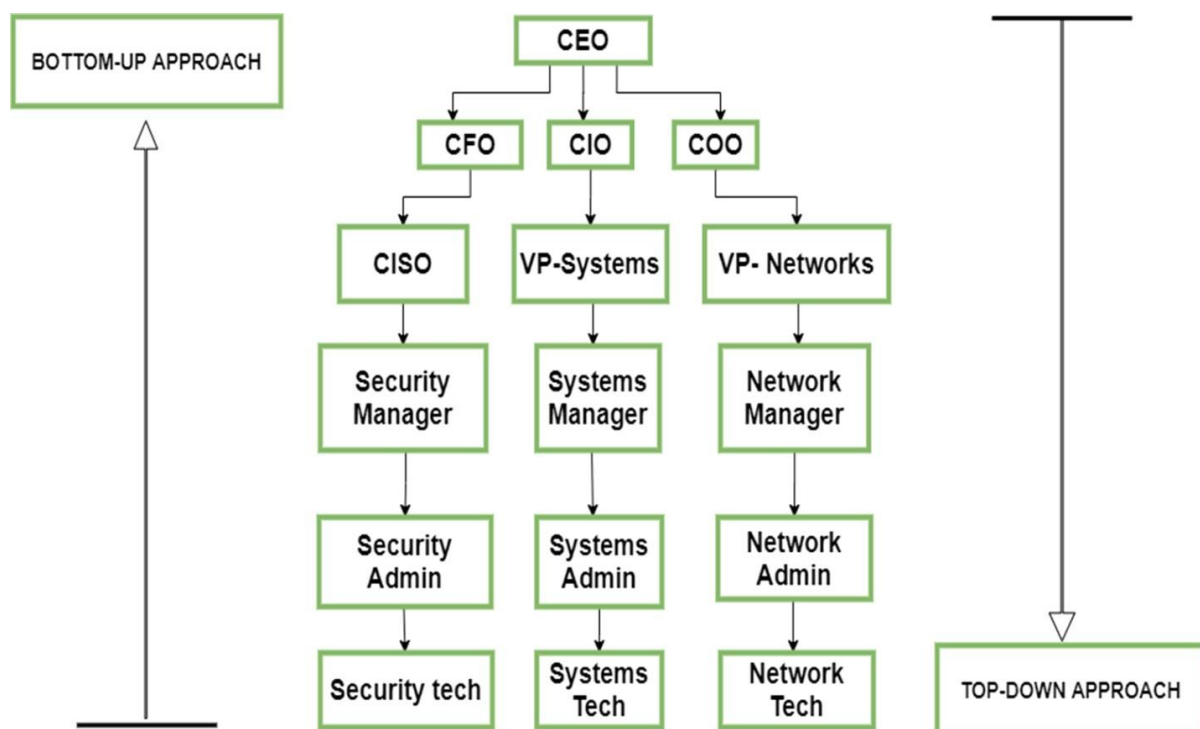
In order to determine the safety of data from potential violations and cyber-attacks. In order to ensure the integrity of the security model can be designed using two methods:

1. Bottom-Up Approach:

The company's security model is applied by system administrators or people who are working in network security or as cyber-engineers. The main idea behind this approach is for individuals working in this field of information systems to use their knowledge and experience in cyber security to guarantee the design of a highly secure information security model.

2. Top-Down Approach:

This type of approach is initialized and initiated by the executives of the organization. It is more likely to succeed. That strategy usually provides strong support from top management by committing resources, a consistent preparation and execution mechanism and opportunities to affect corporate culture.



Security Principles

A principle which is a core requirement of information security for the safe utilization, flow and storage of information is the CIA triad. CIA stands for confidentiality, integrity, and availability and these are the three main objectives of information security.

Confidentiality: This is equivalent to privacy, and it has a set of rules which limits access to information. It protects against disclosure of information to unintended recipients. It ensures that only the designated person gets the information and access will be restricted to those authorized to view the data in question.

Integrity: It involves maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle, and allows transferring accurate and desired information from senders to intended receivers. It ensures that data cannot be altered by unauthorized people (for example, in a breach of confidentiality).

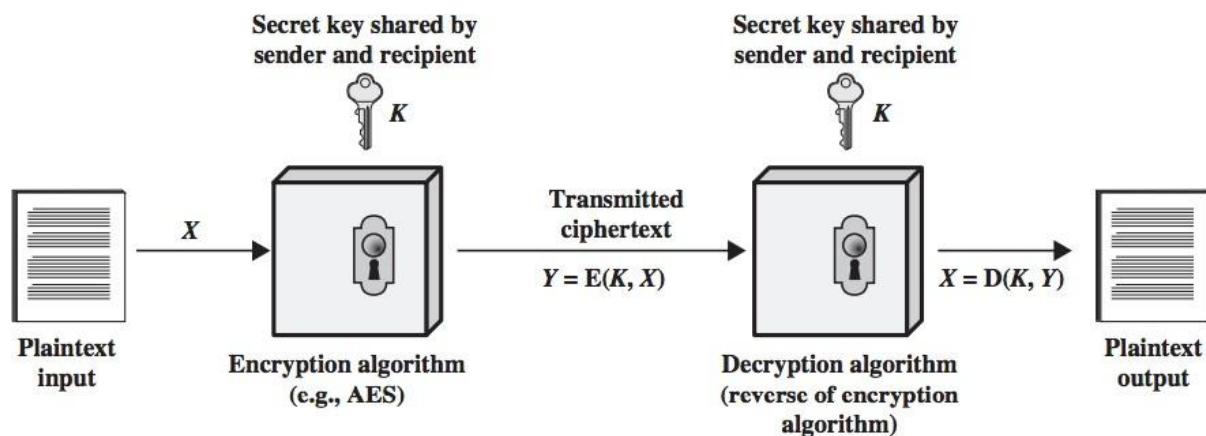
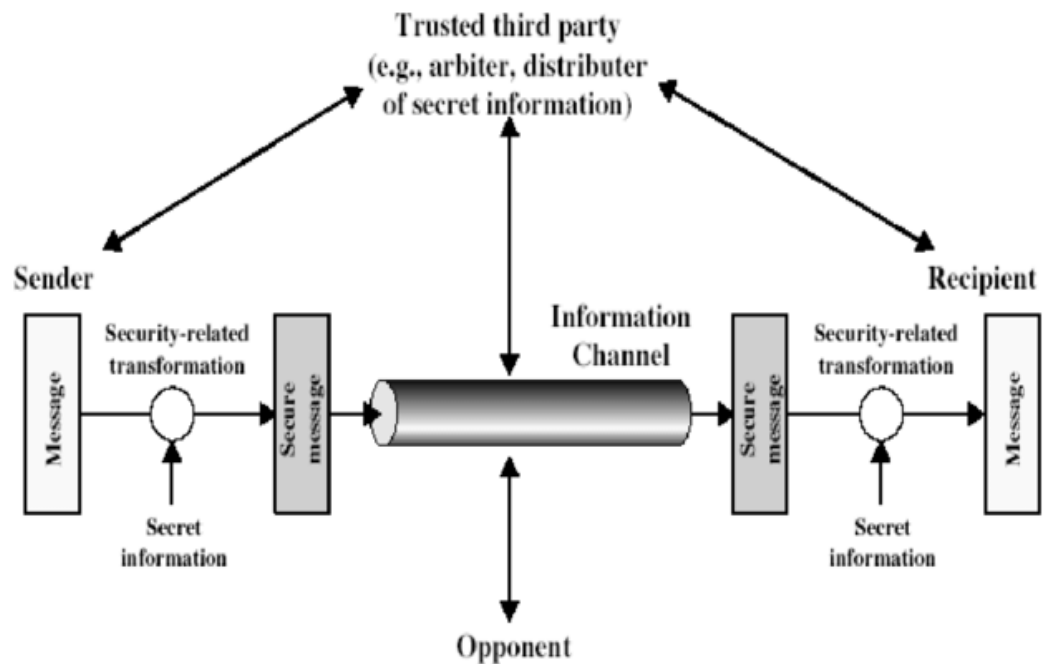
Authentication: This confirms the identity of a user and allows a user to have confidence that the information he receives originated from specific known sources.

Authorization: It specifies access rights to the users, based on the user role.

Availability: Ensures the readiness of the information on requirement. To simplify, information must be available to authorized person(s) when they require it. Availability is best ensured by rigorously maintaining all hardware, performing hardware repairs immediately when needed and maintaining a correctly functioning operating system environment that is free of software conflicts.

Non-repudiation: This ensures there is no denial from the sender or the receiver for sent /received messages. It exchanges authentication information with provable time stamp, for example, 'session id' and so forth.

A Model for Network Security



The requirement in this scheme both sender and receiver should know the key in the same way.

Explanation:

1. **Plain Text:** This is the general English language which can be understood by any person.
2. **Cipher Text:** This is resultant after applying encryption algorithm on the plain text with respect to the key.
3. **Encryption:** This is the process by which the plain text is converted into the cipher text using the key.
4. **Decryption:** This is the process by which the cipher text is converted into the plain text.
5. **Key:** This is the secret code used by authorised person while encryption and decryption procedure is doing.

Plain Text and Cipher Text

Plain Text: This is information that can be directly read by humans or a machine. This is the original message. Plaintext is a historic term pre-dating computers, when encryption was only used for hardcopy text, nowadays it is associated with many formats including music, movies and computer programs.

$$C = E(P)$$

Cipher Text: Cipher text is the result of encryption performed on plaintext using an algorithm, called a cipher. Cipher text is also known as encrypted or encoded information because it contains a form of the original plaintext that is unreadable by a human or computer without the proper cipher to decrypt it. Decryption, the inverse of encryption, is the process of turning ciphertext into readable plaintext. Cipher text is not to be confused with codetext because the latter is a result of a code, not a cipher.

$$P = D(C)$$

Substitution and Transposition Techniques

Substitution techniques

A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.

Types of Substitution Techniques

Caesar Cipher Algorithm:

This is the simplest substitution technique developed by Caesar. It involves replacing each letter with an alphabet standing 3 position, further to it. Note that the alphabets are wrapped around (A after Z).

Here the alphabets are assigned numerical values (a=1, b=2,, z=26). So, we can use the following substitutions.

PT	a	b	c	d	w	x	y	z
CT	d	e	f	g	z	a	b	c

Where PT refers plaintext and CT refers ciphertext. We can use the following expression for Caesar Cipher algorithm.

$$C = (P+3) \bmod 26$$

The Caesar Cipher decryption algorithm uses the following expression.

$$P = (C-3) \bmod 26$$

A slight variation to the Caesar Cipher algorithm is called “Captain Midnight Code”. The difference is in Caesar Cipher algorithm the key is always ‘3’, where as in Captain Midnight Code the key is any value between 1 and 25.

The expression for Captain Midnight Code is:

$$C = (P+K) \bmod 26, \text{ where } k=1 \text{ to } 25$$

$$P = (C-K) \bmod 26, \text{ where } k=1 \text{ to } 25 \text{ (decryption)}$$

2. Play fair Algorithm:

This is the simplest multi letter encryption algorithm. It treats diagrams in the plain text as a single unit and converts it into cipher text diagram.

This algorithm is based on a 5 x 5 matrix. Construction by using the key shared between the parties.

Let us use the keyword MONARCHY.

The 5 x 5 matrix is constructed in the following way.

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

Now we use the following rules to encrypt the given message.

Plain text letters in the diagram that fall in the single or same row of the above matrix are replaced with next letters in the same row.(The letters are wrapped around).

Ex: ON becomes NA, FG becomes GI/GJ, LQ becomes PS

Plain text letters in the diagram that fall in the same column of the matrix are replaced with next letter in the same column

Ex: HF becomes FP, BI becomes IS/JS, DT becomes KZ


The plain text letters that are not in same row or not in the same column or each plain text letter is replaced by the letter the lies in the same row and other letters column.

Ex: MH becomes OC, HK becomes DF, PX becomes SV

3. Mono-alphabetic Cipher:

Predictability of Caesar Cipher was its weakness once any key replacement of a single alphabet is known then, the whole message can be deciphered and almost 25 attempts are required to break it.

In this technique, we simply substitute any random key for each alphabet letter, that is 'A' can be replaced with any letter from B to Z and 'B' can be changed to rest of the Alphabets but itself and so on. Let's say we substitute A with E that doesn't mean that B will be replaced by F.



Monoalphabetic Cipher

- It is an improvement of Caesar Cipher.
- Each plaintext letter is replaced with different random cipher text letter.
- So the key is 26 letters long in size.

Plain:	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
Cipher:	y	n	l	k	x	b	s	h	m	i	w	d	p	j	o	q	v	f	e	a	u	g	t	z	c	

Example

- Plaintext: C O M P U T E R
- Ciphertext: L R P O A E X V

Twinkl Patel MGITER (033)

4. Homophonic Substitution Cipher:

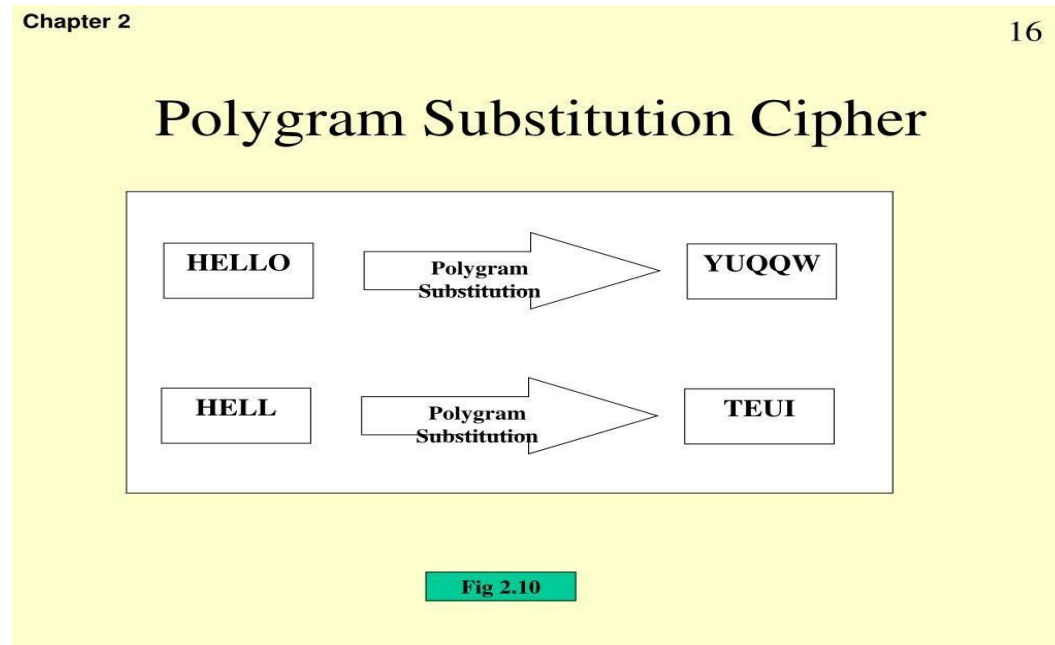
The Homophonic substitution and mono-alphabetic substitution are very much alike. Like in plain cipher substitution we replace an alphabet with a key but in case of Homophonic Substitution, we map an alphabet with a set of fixed keys (more than one key). For instance, A can be replaced with H, J, O, P and B will replace with any of the following in spite of A's key set D, I, W, Z etc.

5. Polygram Substitution Cipher:

In Polygram substitution cipher, instead of replacing one plain-text alphabet we simply replace a block of the word with another block of a word. Example, 'INCLUDEHELP' will change to 'WDSAEQTGTAI' whereas 'HELP' will replace to

'RYCV'. This is true that the last four letters are the same but still different in both words.

Example:



6. Polyalphabetic Substitution Cipher:

Polyalphabetic Substitution cipher was introduced by Leon Battista in the year 1568, and its prominent examples are Vigenère cipher and Beaufort cipher.

We use multiple one-character keys, each key encrypts one plain-text character. This first key encrypts the first plain-text character, the second the key encrypt the second plain-text character and so on, after all, keys are used then they are recycled. If 50 one-letter keys, every 50th character in the plain text would be placed with the same key and this number (in our case, 50) is period of the cipher.

The key points of the polyalphabetic substitution cipher are the following:

It uses a set of related mono-alphabetic substitution rules.

The rule used for transformations determined by the key it uses.

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

Input : Plaintext : GEEKSFORGEES

Keyword : AYUSH

Output : Ciphertext : GCYCZFMLYLEIM

For generating key, the given keyword is repeated in a circular manner until it matches the length of the plain text.

The keyword "AYUSH" generates the key "AYUSHAYUSHAYU"

The plain text is then encrypted using the process explained below.

Transposition Techniques

In Cryptography, a transposition cipher is a method of encryption by which the positions held by units of plain text are shifted according to a regular system, so that the cipher text constitutes a permutation of the plain text.

There are three types of transposition techniques. They are

- Rail Fence Cipher
- Columnar transposition
- Double Transposition

1) Rail-Fence Technique:

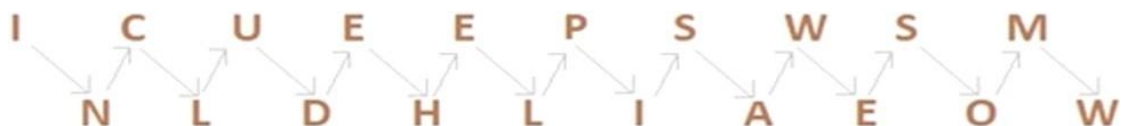
This technique is a type of Transposition technique and does is write the plain text as a sequence of diagonals and changing the order according to each row.

It uses a simple algorithm,

1. Writing down the plaintext message into a sequence of diagonals.

Example,

Let's say, we take an example of "INCLUDE HELP IS AWESOME". 2.Row-wise writing the plain-text written from above step.



So the Cipher-text are, ICUEEPSWSMNLDLIAEOW.

First, we write the message in a zigzag manner then read it out direct row-wise to change it to cipher-text.

Now as we can see, Rail-Fence Technique is very to break by any cryptanalyst.

2) Columnar Transition Technique:

It is a slight variation to the Rail-fence technique, let's see its algorithm:

In a rectangle of pre-defined size, write the plain-text message row by row.

Read the plain message in random order in a column-wise fashion. It can be any order such as 2, 1, 3 etc.

Thus Cipher-text is obtained.

Let's see an example:

Original message: "INCLUDEHELP IS AWESOME".

Now we apply the above algorithm and create the rectangle of 4 columns.

Column 1	Column 2	Column 3	Column 4
I	N	C	L
U	D	E	H
E	L	P	I
S	A	W	E
S	O	M	E

Now let's decide on an order for the column as 4, 1, 3 and 2 and now we will read the text in column-wise.

Cipher-text: LHIEEIUESSCEPWMNDLAO

3) Double Transposition:

Double Transposition consists of two applications of columnar transposition to a message. The two applications may use the same key for each of the two steps, or they may use different keys.

Columnar transposition works like this: First pick a keyword, such as DESCRIBE, then write the message under it in rows:

DESCRIBE

YOURMOTH

ERWASAHA

MSTERAND

YOURFATH

ERSMELTO

FELDERBE

RRIES

Now number the letters in the keyword in alphabetical order.

3 4 8 2 7 6 1 5

DESCRIBE

YOURMOTH

ERWASAHA

MSTERAND

YOURFATH

ERSMELTO

FELDERBE

RRIES

Then read the cipher off by columns, starting with the lowest-numbered column: Column 1 is THNTTB, followed by RAERMDE YEMYEFR ORSORER HADHOE OAAALR MSRFEE UWTUSLI. This completes the first columnar transposition. Next, select and number a second keyword, and write this intermediate ciphertext under it in rows:

2 7 1 8 9 5 4 6 3

COASTLINE

THN TTBR AE

RMDEYEM YE

FRORSORER

HADHOEOAA

ALRMSRFEE

SUWTUSLI

Finally, take it off by columns again and put it into five-letter groups for transmission.

NDODR WTRFH ASEER AERMROFLBE OERSA YEAEI

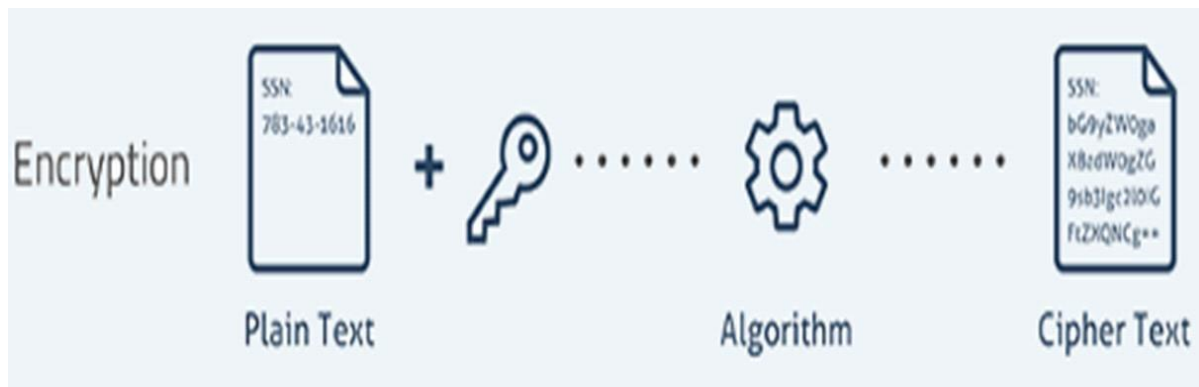
HMRAL UTERH MTTYS OSU

To decrypt a double transposition, construct a block with the right number of rows under the keyword, blocking off the short columns. Write the cipher in by columns, and read it out by rows. Lather, rinse, repeat.

Encryption and Decryption

Encryption

Encryption is a process which transforms the original information into an unrecognizable form. This new form of the message is entirely different from the original message. That's why a hacker is not able to read the data as senders use an encryption algorithm. Data is encrypted to make it safe from stealing.



Decryption

Decryption is a process of converting encoded/encrypted data in a form that is readable and understood by a human or a computer. This method is performed by un-encrypting the text manually or by using keys used to encrypt the original data.



Symmetric and Asymmetric Cryptography

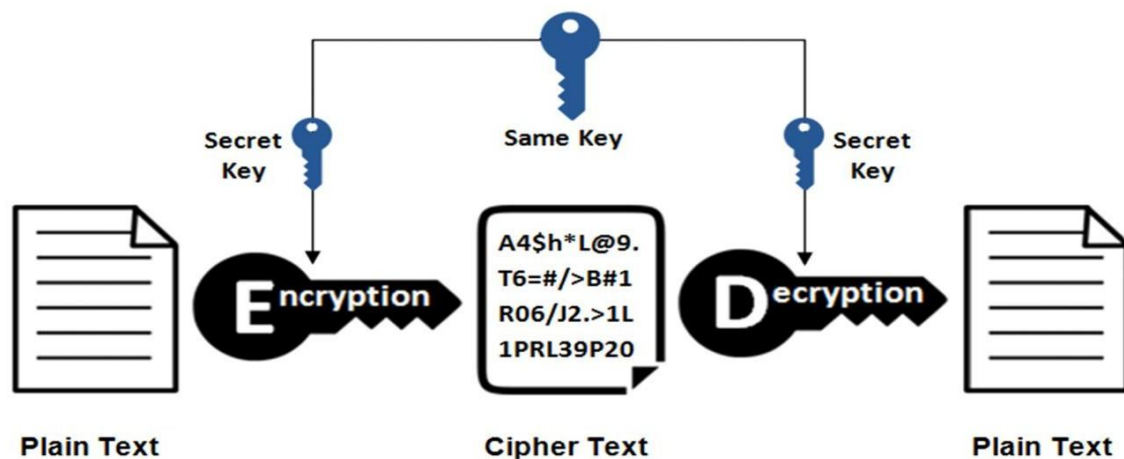
Symmetric encryption uses a single key that needs to be shared among the people who need to receive the message while Asymmetrical encryption uses a pair of public key and a private key to encrypt and decrypt messages when communicating.

The main features of symmetric cryptography are as follows –

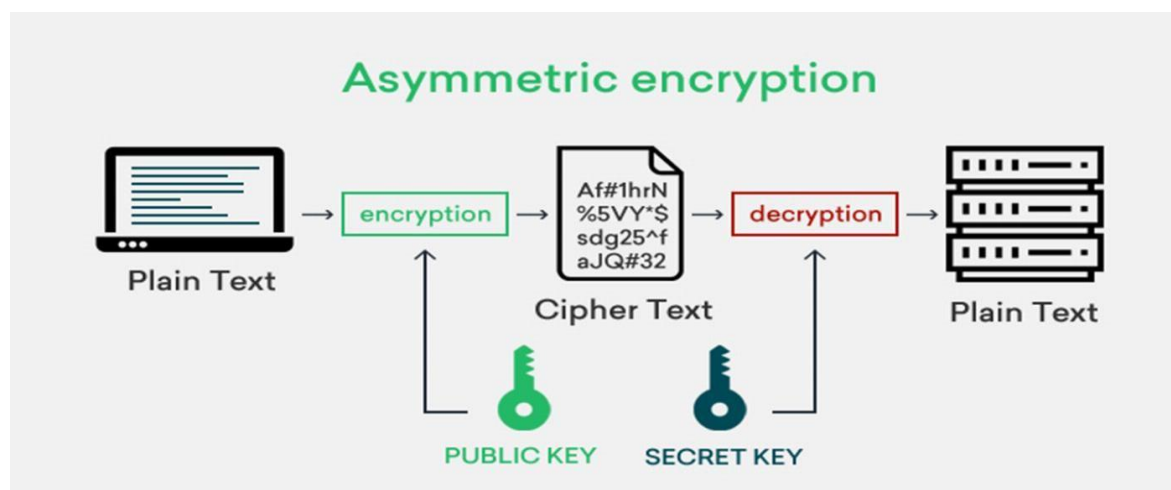
It is simpler and faster.

The two parties exchange the key in a secure way.

Symmetric Encryption



In asymmetric cryptography the public key is used for encrypting and the private key is used for decrypting.



Steganography

A plaintext messages may be hidden in any one of the two ways. The methods of steganography conceal the existence of the message, Whereas the methods of cryptography render the message unintelligible to outsiders by various transformation of the text.

A simple form of steganography, but one that is time consuming to construct is one in which an arrangement of words or letters which an apparently innocuous text spells out the real message.

Ex: 1 The sequence of first letters of each word of the overall message spells out the real hidden message.

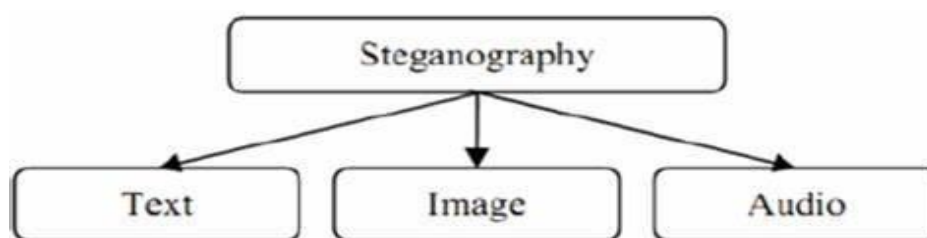
Ex: 2 subset of the words of the overall message is used to convey the hidden message.

Various other techniques have been used historically, some of them are:

1. **Character marking:** Selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held to an angle to bright light.
2. **Invisible ink:** A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to the paper.
3. **Pin punctures:** Small pin punctures on selected letters are ordinarily not visible unless the paper is held in front of the light. Typewritten correction ribbon-used between the lines typed with a black ribbon, the results of typing with the correction types are visible only under a strong light.

Drawbacks of Steganography:

1. Requires a lot of overhead to hide a relatively few bits of information.
2. Once the system is discovered, it becomes virtually worthless.



Key range and Key size

The concept of key range and key-size are related to each other. Key Range is total number of keys from smallest to largest available key. An attacker usually is armed with the knowledge of the cryptographic algorithm and the encrypted message, so only the actual key value remains the challenge for the attacker.

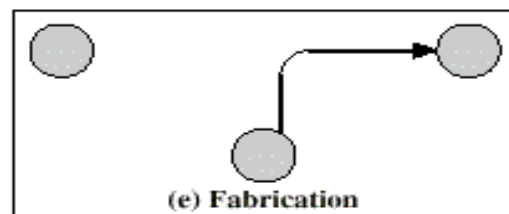
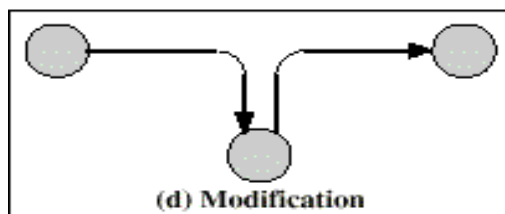
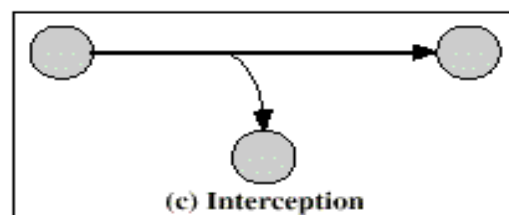
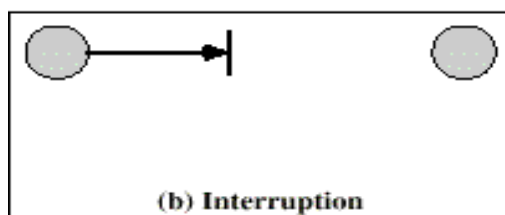
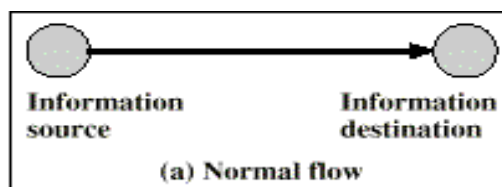
1. If the key is found, the attacker can get original plain text message. In the brute force attack, every possible key in the key-range is tried, until we get the right key.
2. In the best case, the right key is found in the first attempt, in the worst case, the key is found in the last attempt. On an average, the right key is found after trying half of the possible keys in the key range. Therefore by expanding the key range to a large extent, longer it will take for an attacker to find the key using brute-force attack.
3. The concept of key range leads to the principle of the key size. The strength of a cryptographic key is measured with the key size.
4. Key Range is total number of keys from smallest to largest available key. Key size is measured in bits and is represented using binary number system. Thus if the key range from 0 to 8, then the key size is 3 bits or in other words we can say if the size is bits then the key range is 0 to 256. Key size may be varying, depending upon the applications and the cryptographic algorithm being used, it can be 40 bits, 56 bits, 128 bits and so on.
5. From a practical viewpoint, a 40-bit key takes about 3 hours to crack, however a 41-bit key would take 6 hours and 42-bit key would take 12 hours and so on. This means every additional bit double the amount of time required to crack the key. We can assume the 128 bit key is quite safe, considering the capabilities of today's computers. However as the computing power and techniques improve, these numbers will change in future.

Types of Attacks

Security Attacks:

An attack on the security of a computer system may be defined as “Threat”. There are four categories of attacks which are listed below:

1. **Interruption:** Here the message is destroyed in the middle and is made unavailable to the receiver. This is also called as an “Attack on Availability”.
2. **Interception:** Here the message is accessed by an unauthorized user during the process of the transmission. Here the receiver receives the message and not unaware of the introducers. This is called attack on “Confidentiality”.
3. **Modification:** Here the intruders or unauthorized party gains the access over the communication channel and retrieves the message from the sender and then modifies it, and send it to receiver. This is called as an “Attack on Integrity”.
4. **Fabrication:** Here an intruder will insert a message into the communication channel and send it to the receiver. This is also called as an “Attack on Authentication”.



Cryptographic Attacks (OR) Types of Attacks:

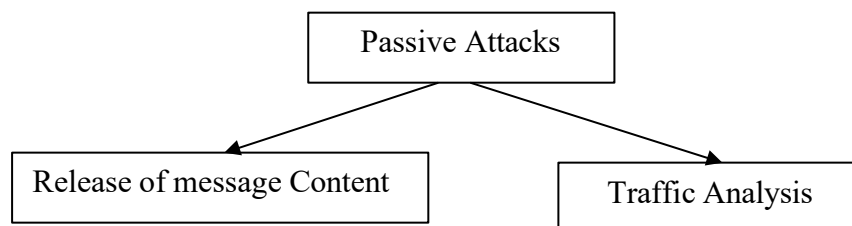
An attack on computer system security is defined as threat or attack. In a technologist's point of view, the attacks are classified into two categories.

They are

1. Passive attacks or Passive threats.
2. Active attacks or Active threats.

1. Passive Attacks

Passive attacks are in the nature of eavesdropping (spy) on, or monitoring of transmissions. The goal of passive threats is to know the message contents which are being transmitted. Here the message is not altered in the middle. There are two types of passive attacks are the 'Release of Message Content' and 'Traffic Analysis'.



a. Release of Message:

Here the unauthorized user will know the message transmission.

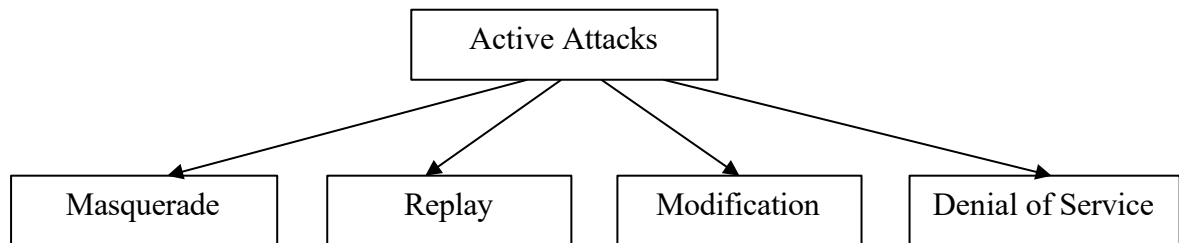
Ex: A telephone conversation or content of e-mail message etc.

b. Traffic Analysis:

In the traffic analysis the unauthorized person always monitors the authorized person, analysis the no. of communications between them and also the length of the message transmission. He also analysis how many times the receiver going to respond and passing on these factors the unauthorized person guesses the message transmission.

2. Active Attacks

In the Active attacks the message is altered in the middle. These threats can be classified as four categories: Masquerade, Replay, Modification of Messages, and Denial of Service.



1. Masquerade:

Here an unauthorized party or an intruder pretends to act like an authorized user and sends the message to all the other authorized users.

2. Replay

Here an unauthorized person copies the previous transmission between the authorized parties and send it to receiver no. of times as replay.

3. Modification of messages

Here the message is being altered by the unauthorized party in the process of transmission and send to the receiver.

4. Denial of Service

It prevents the normal use communication facilities like description of the entire network etc.,

Number Theory

Introduction to number theory

A number of concepts from number theory are essential in the design of public-key cryptographic algorithms.

Prime numbers:

“A prime number is an integer that can only be divided without remainder by positive and negative values of itself and 1. Prime numbers play a critical role both in number theory and in cryptography.”

Ex: 2,3,5,7 are prime, 4,6,8,9,10 are not.

Relatively Prime Numbers & GCD:

Two numbers a, b are **relatively prime** if there is **no common divisors** apart from 1.

Ex: 8 & 15 are relatively prime since factors of 8 are 1,2,4,8 and of 15 are 1,3,5,15 and 1 is the only common factor.

Conversely, we can determine the greatest common divisor (GCD) by comparing their prime factorizations and using least powers.

Ex: $300=2^2 \times 3^1 \times 5^2$ $18=2^1 \times 3^2$ hence $\text{GCD}(8,300)=2^1 \times 3^1 \times 5^0=6$

Two theorems that play important roles in public-key cryptography are Fermat's theorem and Euler's theorem.

Modular Arithmetic

Most of the public key algorithms are based on modular arithmetic. Modular arithmetic uses the non negative integers less than some positive integer ‘ n ’, performs ordinary arithmetic operations such as addition and multiplication and then replaces the result with its remainder when divided by ‘ n ’. The result is said to be **modulo n** or **mod n** .

a) Modular Addition:

+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

For example,

a) $7 + 6 = 13$ in regular arithmetic, but the mod 10 answer is 3.

b) $5 + 5 = 0$

c) $3 + 9 = 2$

Like regular arithmetic, subtracting x can be done by adding $-x$, also known as x 's **Additive Inverse**. An additive inverse of x is the number, we have to add to x to get 0 (zero).

For example, 4's inverse will be 6, because in mod 10 arithmetic $4 + 6 = 0$. If the secret key is 4, then to encrypt we have 4 (mod 10) and decrypt we have 6 (mod 10).

b) Modular Multiplication:

Z_n^* Theorem

The function **Z_n^*** is closed under **multiplication mod n**. This means that the **Z_n^*** modular multiplication table only includes the numbers which are relatively prime to **n**.

*	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	1	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	1	8	5	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	1

For example,

- a) $7 * 6 = 42$ in regular arithmetic, but the mod 10 answer is 2.
- b) $5 * 5 = 5$
- c) $3 * 9 = 7$

The **Multiplicative Inverse** of x (x^{-1}) is the number by which we have to multiply x to get 1 (one). Only the numbers $\{1,3,5,7,9\}$ have multiplicative inverses mod 10.

For example, 7 is the multiplicative inverse of 3. So encryption could be performed by multiplying by 3, and decryption could be performed by multiplying 7. 9 is its own inverse. 1 is its own inverse.

Euclidean or Euclid's Algorithm

The Euclidean or Euclid's algorithm is a way to find the greatest common divisor of two positive integers.

Formal description of the Euclidean algorithm

Input: Two positive integers, a and b .

Output: The greatest common divisor, g , of a and b .

1. The greatest common divisor is the largest integer that evenly divides into both integers.
2. The algorithm repeatedly replaces the original numbers with smaller numbers that have the same greatest common divisor until one of them is zero. The remainder is the greatest common divisor.

This algorithm is based on the following equation. For any non-negative integer a and any positive integer b ,

$$\text{gcd}(a,b) = \text{gcd}(b, a \bmod b)$$

Ex: $\text{gcd}(55,22) = \text{gcd}(22, 55 \bmod 22)$
 $= \text{gcd}(22,11)$
 $= \text{gcd}(11,22 \bmod 11)$
 $= \text{gcd}(11,0) = 11$

Algorithm:

Step 1: Select 2 integers x and y which we believe has a common divisor.

Step 2: Sort x and y in descending order.

Step 3: If $y = 0$, then return $\text{gcd}(x,y) = x$ and stop.

Step 4: The remainder is $r = x \bmod y$

Step 5: Assign $x = y$ and $y = r$

Step 6: Go to Step 3.

Ex: $\text{gcd}(77,22)$

- i) $x = 77$ and $y = 22 \neq 0$
- ii) $r = 77 \bmod 22 = 11$
- iii) $x = 22$ and $y = 11 \neq 0$
- iv) $r = 22 \bmod 11 = 0$
- v) $x = 11$ and $y = 0$, therefore $\text{gcd}(77,22) = x = 11$

Euler theorem and Totient Function

The Euler's totient function, or phi (ϕ) function is a very important number theoretic function having a deep relationship to prime numbers and the so-called order of integers.

This function produces 'm' number of positive integer values which are less than 'n' and they are relatively prime to 'n'. This is denoted using the symbol ' ϕ '.

- $\Phi(n) = m$ is the number of positive integers less than n and relatively prime to n .
- If n is a prime number, then $\phi(n) = n-1$.
- If n is not a prime, then let p and q are prime numbers, with $p \neq q$, we can write as

$$\Phi(n) = \Phi(pq) = \Phi(p) \times \Phi(q) = (p-1) \times (q-1)$$

Ex: a) Determine $\Phi(37)$

$n = 37$ is a prime number. By using the above function $\Phi(37) = 36$.

i.e., all the +ve integers from 1 through 36 are relatively prime to 37.

b) Determine $\Phi(35)$.

$n = 35$ is not a prime number. By using the above function,

$p = 7$ and $q = 5$, $7 \times 5 = 35$, with $p \neq q$.

$\Phi(35) = \phi(7 \times 5) = \phi(7) \times \phi(5) = (7-1) \times (5-1) = 6 \times 4 = 24$ positive integer numbers relatively prime to 35.

Fermat Theorem

Fermat's little theorem is a fundamental theorem in elementary number theory, which helps compute powers of integers modulo prime numbers. It is a special case of Euler's theorem, and is important in applications of elementary number theory, including primality testing and public-key cryptography.

If **p** is prime and **a** is a positive integer not divisible by p, then

$$a^{p-1} = 1(\text{mod } p) \Rightarrow \mathbf{a^{p-1} \bmod p = 1}, \text{ where } p \text{ is prime and } \gcd(a, p) = 1.$$

This also known as Fermat's Little Theorem.

Ex: $a=7, p=19$

$$\text{Therefore } a^{p-1} = 7^{19-1} = 7^{18}$$

Now Calculate

$$7^2 = 49 = 11(\text{mod } 19)$$

$$7^4 = 121 = 7(\text{mod } 19)$$

$$7^8 = 49 = 11(\text{mod } 19)$$

$$7^{16} = 121 = 7(\text{mod } 19)$$

$$a^{p-1} = 7^{18} = 7^{16} \times 7^2 = 7 \times 11 = 1(\text{mod } 19)$$

Multiplicative and Additive Inverse

Multiplicative Inverse

A **multiplicative inverse** or **reciprocal** for a number x denoted by $1/x$ or x^{-1} , is a number which when multiplied by x yields the multiplicative identity 1. The multiplicative inverse of a fraction a/b is b/a . For the multiplicative inverse of a real number, divide 1 by the number.

For example, the reciprocal of 5 is one fifth ($1/5$ or 0.2), and the reciprocal of 0.25 is 1 divided by 0.25, or 4.

Hence, $a \times (1/a) = (1/a) \times a = 1$, where a can be rational number or natural number or integer.

Example:

$$3 \times 1/3 = 1$$

$$(-4/5) \times (1/(-4/5)) = (-4/5) \times (-5/4) = 1$$

Additive Inverse:

The additive inverse is defined as its inverse element under the binary operation of addition, which allows a broad generalization to mathematical objects other than numbers i.e., additive inverse of a number a is the number that when added to a yields Zero. This number is also known as the opposite (number), sign change, and negation. For a real number, it reverses its sign: the opposite to a positive number is negative, and the opposite to a negative number is positive. Zero is the additive inverse of itself.

Ex: The additive inverse of a is denoted by unary minus: $-a$.

For example, the additive inverse of 7 is -7, because $7+(-7) = 0$, and the additive inverse of -0.3 is 0.3, because $-0.3+0.3 = 0$.

UNIT-II

Symmetric Key Cryptographic Algorithms

Symmetric Algorithm Types:

Symmetric-key algorithms are algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link.

Algorithm	Key Length
Data Encryption Standard	56-Bit Key
Triple DES	Three DES Operations, 168-Bit Key
Advanced Encryption Standard (AES)	Variable Key Lengths
International Data Encryption Algorithm (IDEA)	128-Bit Key
Blowfish	Variable Key Lengths
RC4	Variable Key Lengths

Block Cipher Principles

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

Stream Cipher: A Stream Cipher is one that encrypts a digital data stream one bit or one byte at a time.

Block Cipher: A Block Cipher is one in which a block of plain text 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.

Symmetric Algorithms Modes or Block Cipher Modes of Operations

The modes of operation of a block cipher: These are procedural rules for a generic block cipher. Interestingly, the different modes result in different properties being achieved which add to the security of the underlying block cipher.

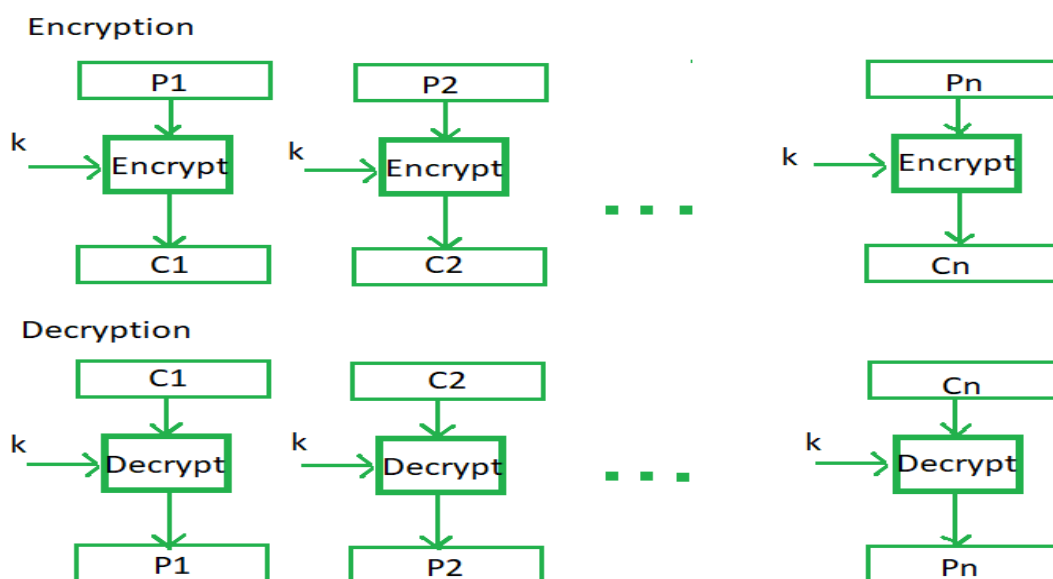
While encrypting the message of length larger than 64-bits, we use the following block cipher modes:

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

1. Electronic Code Book (ECB) Mode

This mode is a most straightforward way of processing a series of sequentially listed message blocks. This is the simplest mode. Initially the message is divided into 64-bit blocks. If it is necessary, the last block is padded on right side with 0s (zeros) to get exactly 64-bits. Now each block is encrypted separately with key, we will get a cipher text block for each plain text block and it is combined to get cipher text of the given message. We have to note that we use the same algorithm and key for each encryption.

The receiver receives the cipher text. He divides it into 64-bits ciphers. Each block is decrypted separately to get plain text blocks. All the blocks are combined to get the original message. The decryption process uses the same key which is used in encryption.



In the above diagram,

Encryption: $C_1 = E_k[M_1]$

$C_2 = E_k[M_2]$

....

$C_n = E_k[M_n]$

Decryption: $M_1 = D_k[C_1]$

$M_2 = D_k[C_2]$

....

$M_n = D_k[C_n]$

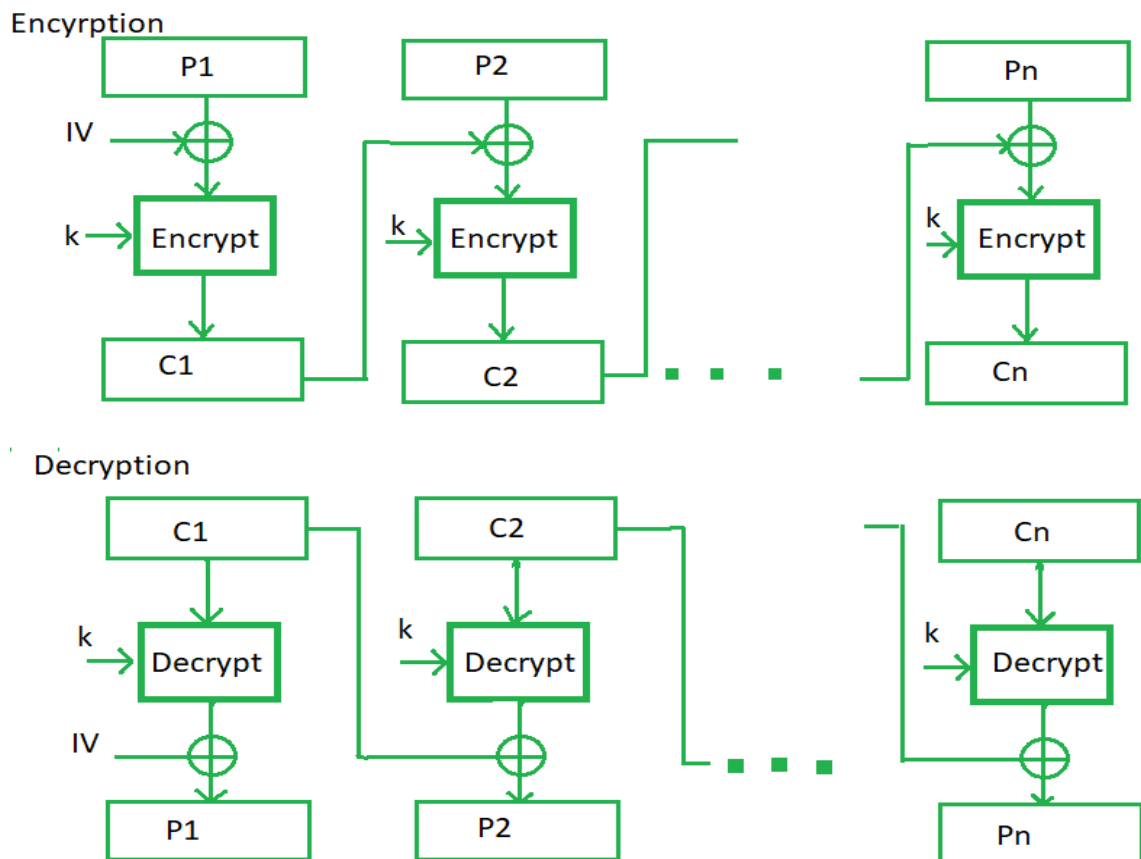
2. Cipher Block Chaining (CBC) Mode

CBC is a method of avoiding some of the problem in ECB. In CBC, though the same block repeats in the plain text, it will not cause repeats in the cipher text.

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

The operation of CBC mode is depicted in the following illustration. The steps are as follows –

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



Encryption:

$$C_1 = E_k[IV (+) M_1]$$

$$C_2 = E_k[C_1 (+) M_2]$$

$$C_3 = E_k[C_2 (+) M_3]$$

.....

$$C_n = E_k[C_{n-1} (+) M_n]$$

Decryption:

$$M_1 = IV (+) D_k[C_1]$$

$$M_2 = C_1 (+) D_k[C_2]$$

$$M_3 = C_2 (+) D_k[C_3]$$

.....

$$M_n = C_{n-1} (+) D_k[C_n]$$

3. Cipher Feedback (CFB) Mode

In this mode, each cipher text block gets 'fed back' into the encryption process in order to encrypt the next plaintext block.

Operation

The operation of CFB mode is depicted in the following illustration.

For example, in the present system, a message block has a size 's' bits where

$1 < s < n$. The CFB mode requires an initialization vector (IV) as the initial random n-bit input block. The IV need not be secret. Steps of operation are –

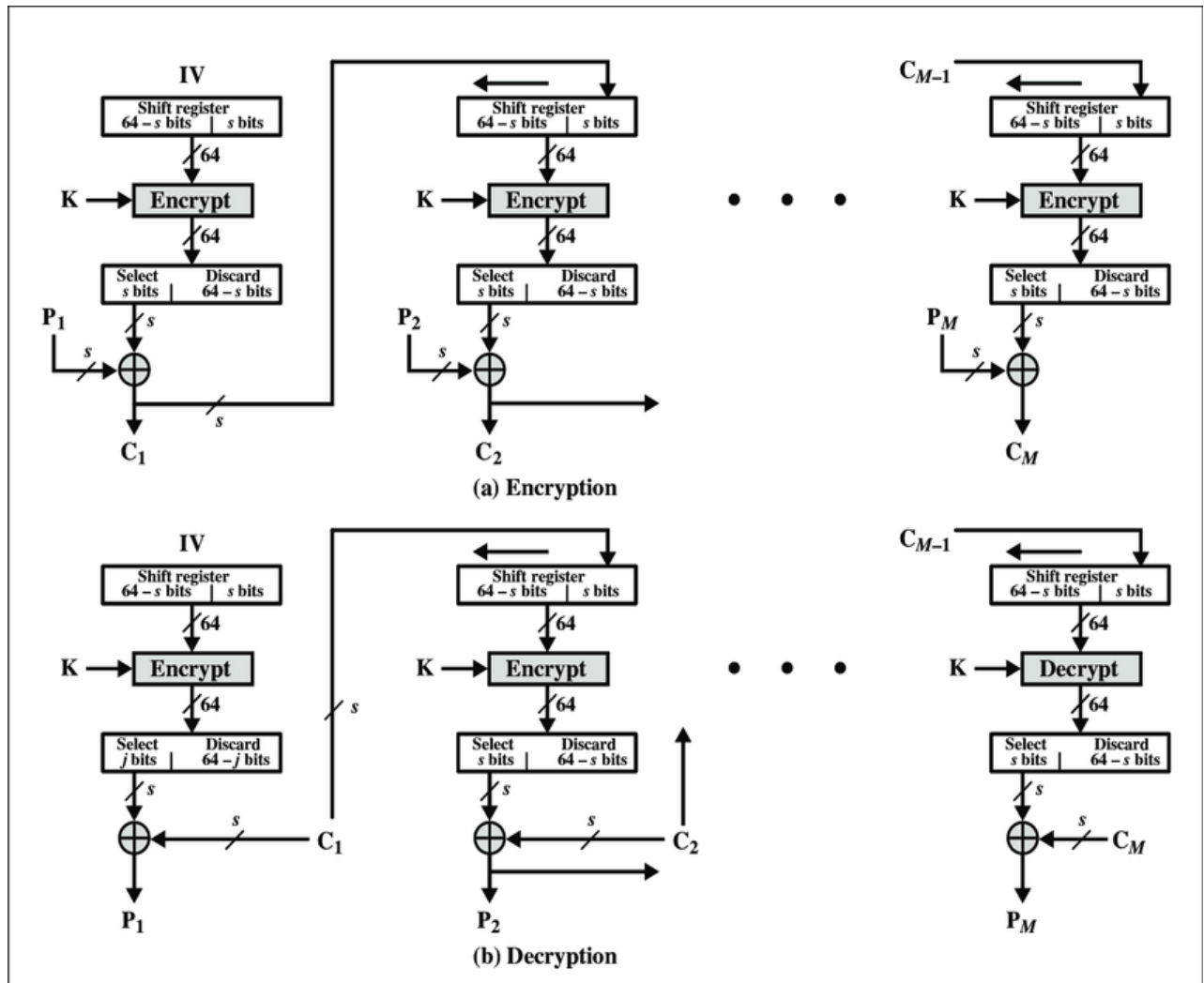
- Load the IV in the top register.
- Encrypt the data value in top register with underlying block cipher with key K.
- Take only 's' number of most significant bits (left bits) of output of encryption process and XOR them with 's' bit plaintext message block to generate cipher text block.
- Feed cipher text block into top register by shifting already present data to the left and continue the operation till all plaintext blocks are processed.
- Essentially, the previous cipher text block is encrypted with the key, and then the result is XORed to the current plaintext block.
- Similar steps are followed for decryption. Pre-decided IV is initially loaded at the start of decryption.

Encryption:

$$C_1 = M_1 (+) [S_k[E_k[IV]]]$$

Decryption:

$$M_1 = C_1 (+) S_k[D_k[IV]]$$

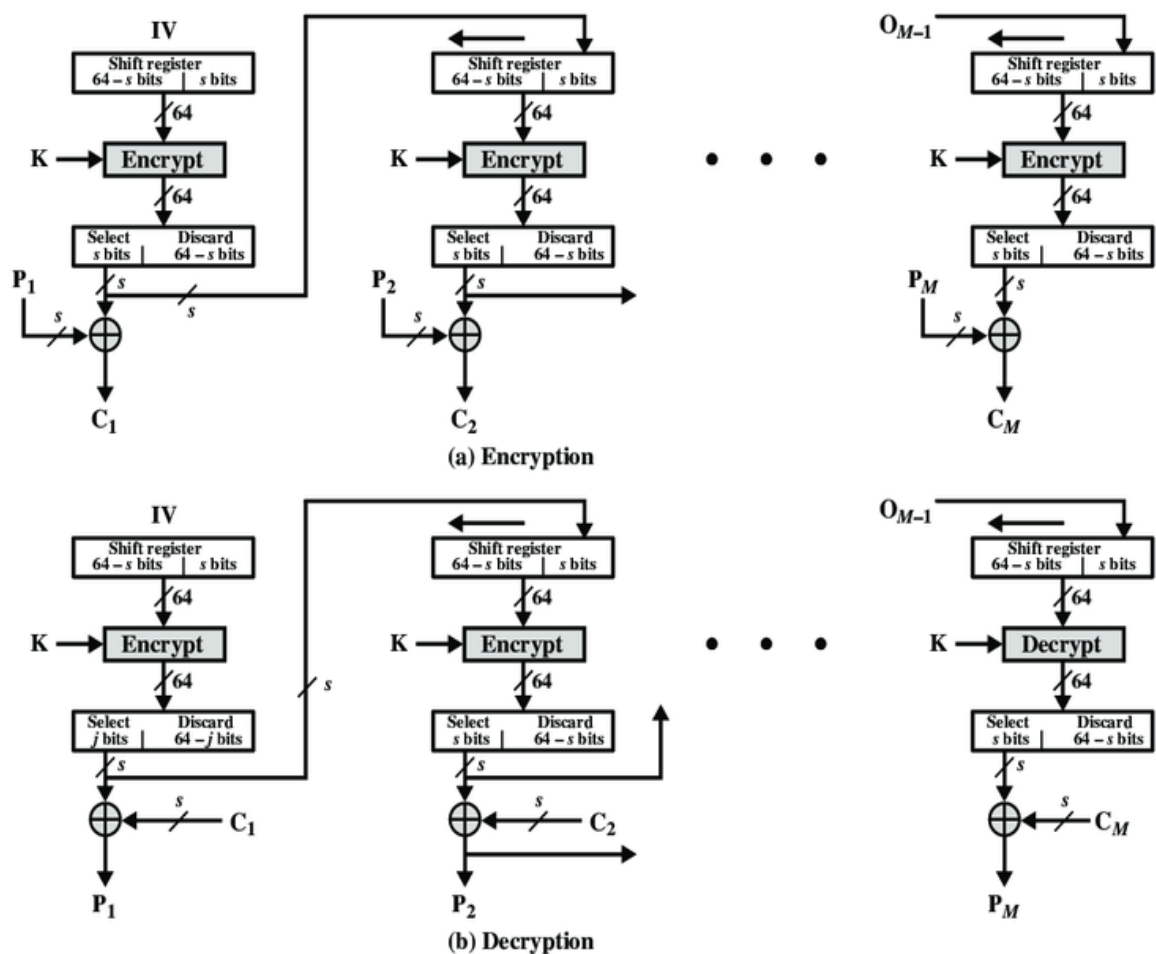


4. Output Feedback (OFB) Mode

It involves feeding the successive output blocks from the underlying block cipher back to it. These feedback blocks provide string of bits to feed the encryption algorithm which act as the key-stream generator as in case of CFB mode.

The key stream generated is XOR-ed with the plaintext blocks. The OFB mode requires an IV as the initial random n -bit input block. The IV need not be secret.

The operation is depicted in the following illustration –



Encryption:

$$C_1 = M_1 (+) [S_k[E_k[IV]]]$$

Decryption:

$$M_1 = C_1 (+) S_k[D_k[IV]]$$

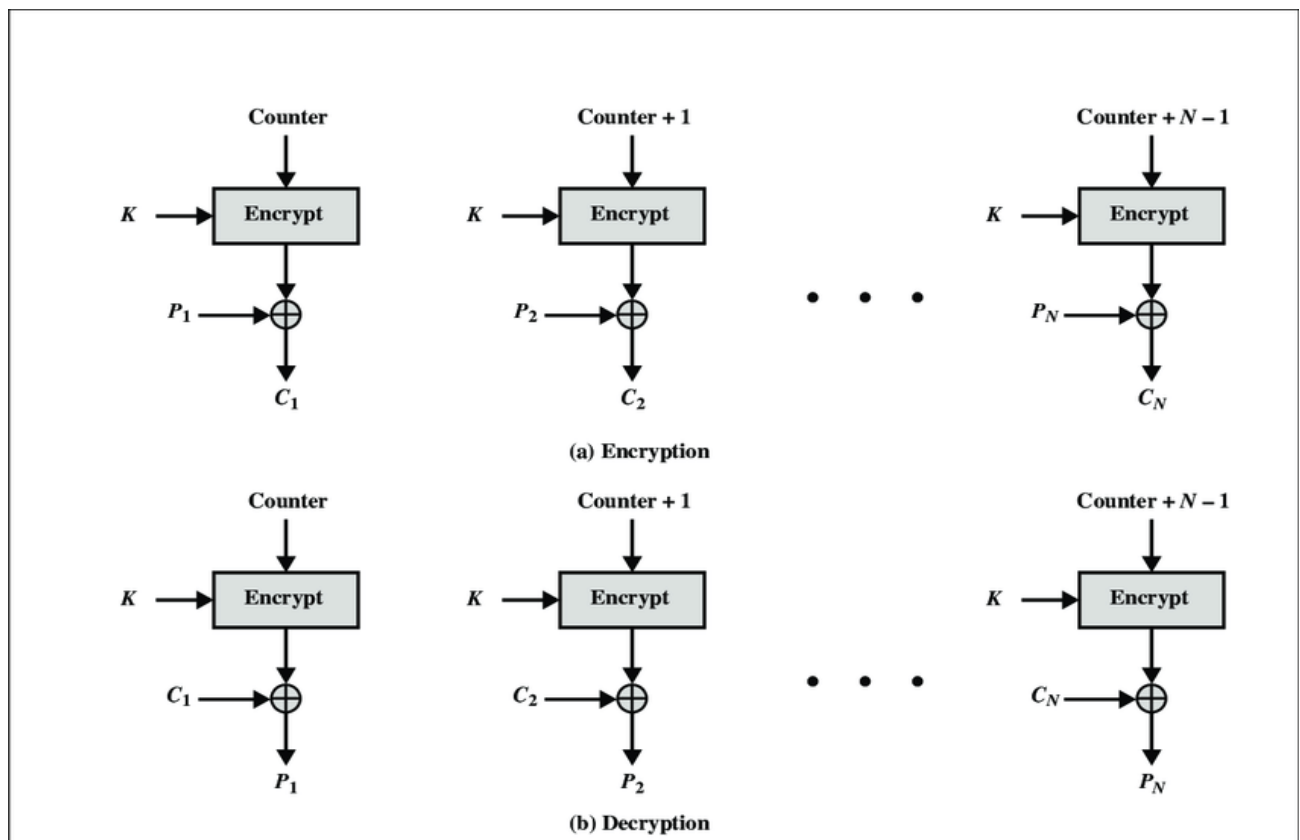
5. Counter (CTR) Mode

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

Operation

Both encryption and decryption in CTR mode are depicted in the following illustration. Steps in operation are –

- Load the initial counter value in the top register is the same for both the sender and the receiver. It plays the same role as the IV in CFB (and CBC) mode.
- Encrypt the contents of the counter with the key and place the result in the bottom register.
- Take the first plaintext block P_1 and XOR this to the contents of the bottom register. The result of this is C_1 . Send C_1 to the receiver and update the counter. The counter update replaces the cipher text feedback in CFB mode.
- Continue in this manner until the last plaintext block has been encrypted.
- The decryption is the reverse process. The cipher text block is XORed with the output of encrypted contents of counter value. After decryption of each cipher text block counter is updated as in case of encryption.



Encryption:

$$C_1 = M_1 (+) [S_k[E_k[IV]]]$$

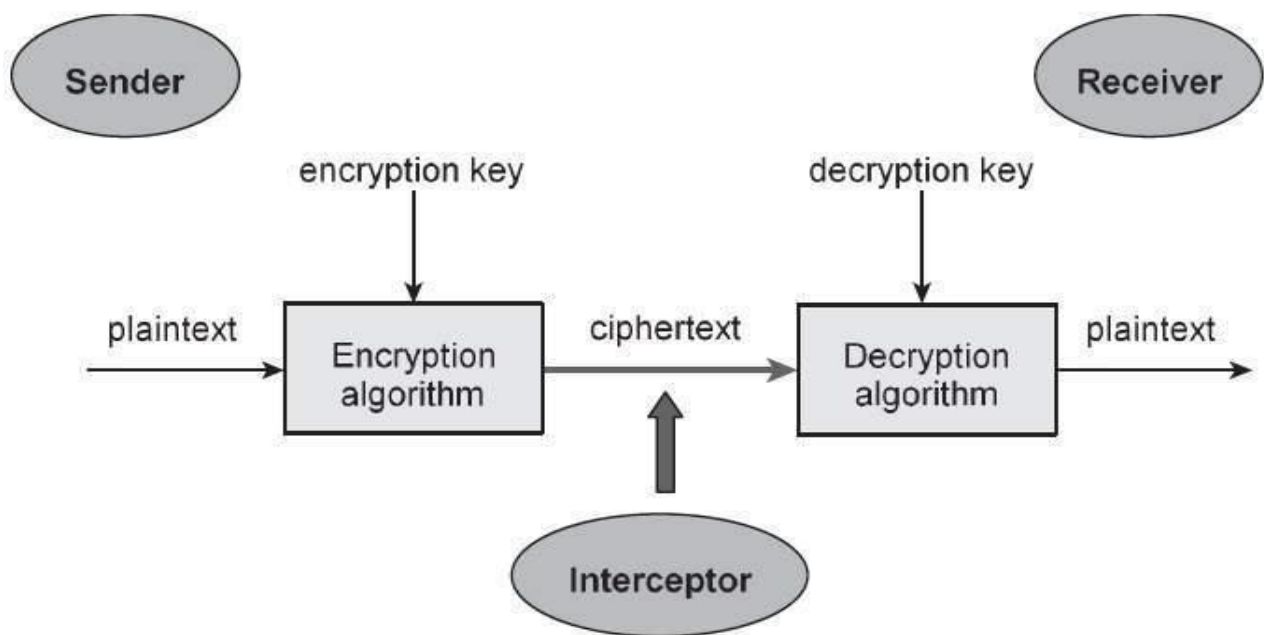
Decryption:

$$M_1 = C_1 (+) S_k[D_k[IV]]$$

Overview of Symmetric Key Cryptography

A cryptosystem is an implementation of cryptographic techniques and their accompanying infrastructure to provide information security services. A cryptosystem is also referred to as a **cipher system**.

A simple model of a cryptosystem that provides confidentiality to the information being transmitted. This basic model is depicted in the illustration below:



Components of a Cryptosystem

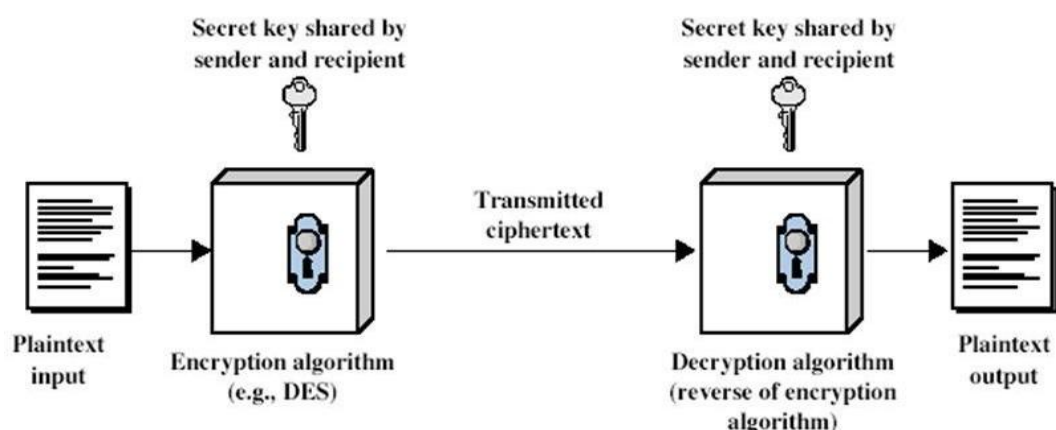
The various components of a basic cryptosystem are as follows –

- **Plaintext.** It is the data to be protected during transmission.
- **Encryption Algorithm.** It is a mathematical process that produces a cipher text for any given plaintext and encryption key. It is a cryptographic algorithm that takes plaintext and an encryption key as input and produces a ciphertext.
- **Cipher text.** It is the scrambled version of the plaintext produced by the encryption algorithm using a specific the encryption key. The ciphertext is not guarded. It flows on public channel. It can be intercepted or compromised by anyone who has access to the communication channel.

- **Decryption Algorithm,** It is a mathematical process, that produces a unique plaintext for any given ciphertext and decryption key. It is a cryptographic algorithm that takes a ciphertext and a decryption key as input, and outputs a plaintext. The decryption algorithm essentially reverses the encryption algorithm and is thus closely related to it.
- **Encryption Key.** It is a value that is known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext in order to compute the cipher text.
- **Decryption Key.** It is a value that is known to the receiver. The decryption key is related to the encryption key, but is not always identical to it. The receiver inputs the decryption key into the decryption algorithm along with the cipher text in order to compute the plaintext.

Symmetric Key Cryptography is a type of encryption where only one key (a secret key) is used to both encrypt and decrypt electronic information. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process. This encryption method differs from asymmetric encryption where a pair of keys, one public and one private, is used to encrypt and decrypt messages.

Symmetric Cipher Model



source: William Stallings

3

There are two types of symmetric encryption algorithms:

1. **Block algorithms.** Set lengths of bits are encrypted in blocks of electronic data with the use of a specific secret key. As the data is being encrypted, the system holds the data in its memory as it waits for complete blocks.
2. **Stream algorithms.** Data is encrypted as it streams instead of being retained in the system's memory.

Some examples of symmetric encryption algorithms include:

- AES (Advanced Encryption Standard)
- DES (Data Encryption Standard)
- IDEA (International Data Encryption Algorithm)
- Blowfish (Drop-in replacement for DES or IDEA)
- RC4 (Rivest Cipher 4)
- RC5 (Rivest Cipher 5)
- RC6 (Rivest Cipher 6)

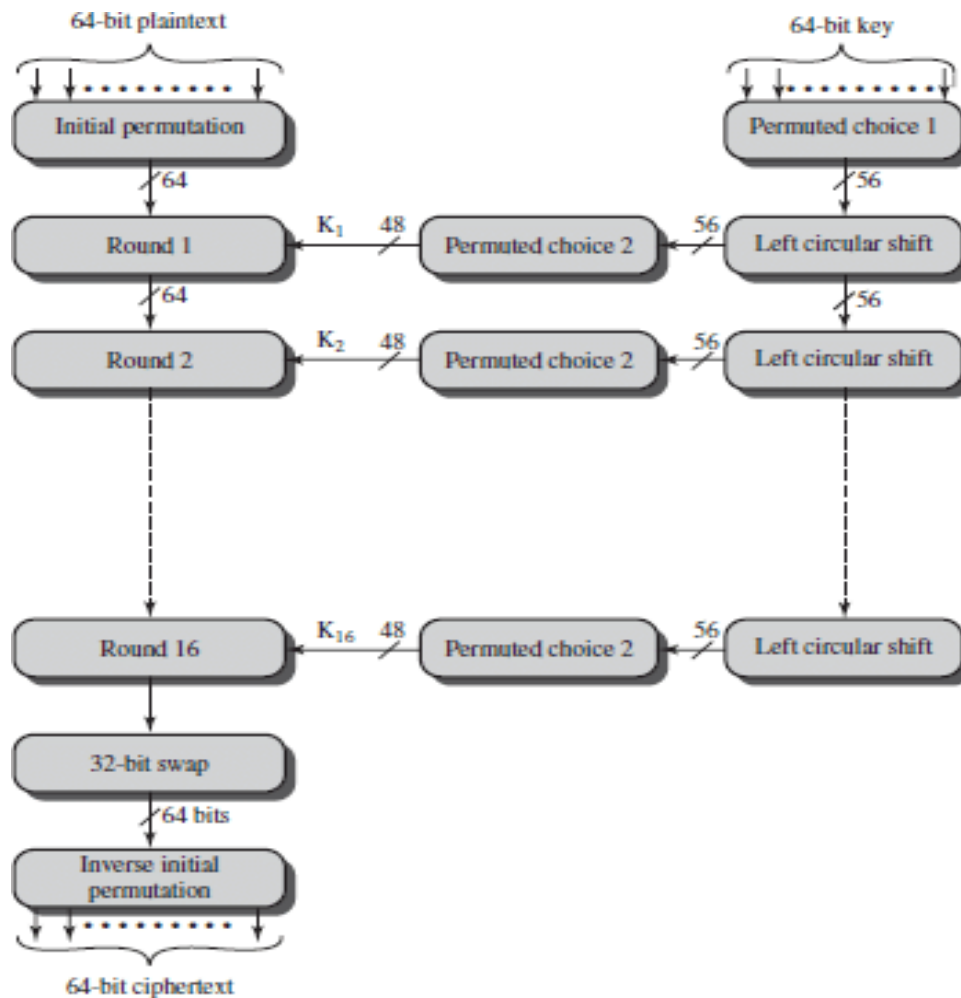
AES, DES, IDEA, Blowfish, RC5 and RC6 are block ciphers. RC4 is stream cipher.

DES Algorithm

The most widely used encryption scheme is based on the DES adopted by the National Bureau of Standards (NBS or NIST- National Institute of Standards and Technology), as Federal Information Standard 46 (FIPS PUB 46).

Data Encryption Standard (DES) is a block cipher algorithm that takes plain text in blocks of 64 bits and converts them to cipher text using keys of 56 bits. It is a symmetric key algorithm, which means that the same key is used for encrypting and decrypting data.

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



The initial plain text is 64-bits, it is written in the form of 8 x 8 matrix. It is given to the IP function. Let our message X (plain text) is

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}
M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}	M_{39}	M_{40}
M_{41}	M_{42}	M_{43}	M_{44}	M_{45}	M_{46}	M_{47}	M_{48}
M_{49}	M_{50}	M_{51}	M_{52}	M_{53}	M_{54}	M_{55}	M_{56}
M_{57}	M_{58}	M_{59}	M_{60}	M_{61}	M_{62}	M_{63}	M_{64}

where M_i is a binary digit. Then the permutation $X = IP(M)$

The DES encryption algorithm involves five functions. They are

1. Initial Permutation
2. Inverse Initial Permutation
3. 32-bit swap
4. Generation of Sub Keys
5. Details of Single Round

1. Initial Permutation (IP):

The plaintext 64-bit block passed to IP function and the bit positions will be changed. The IP function is described as following:

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

2. Inverse Initial Permutation (IP⁻¹):

The output of 32-bit swap is passed to IP⁻¹ function and the output of IP⁻¹ is the 64-bit Cipher text. The format of IP⁻¹ is as follows:

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

3. 32-bit Swap:

The output of 16th round is divided into two 32-bit halves (left half and right half). Now they are swapped and generated 64-bit output.

4. Generation of Sub keys:

A total of 16 sub keys. Each of 48-bits are used in 16 rounds of DES algorithm. These are generated from 56-bit key.

The actual key of DES algorithm looks like 64-bit long. But each and every 8th bit is called the odd parity. So, we exclude these bits i.e., we exclude 8,16,24,32,40,48,56,64. From remaining 56-bits generate the 16 sub keys.

Initially the 56-bits are permuted and divided into two halves (C_0 , D_0). These are as given below:

Let us consider the 56-bit key.

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

57	49	41	33	25	17	9	63	55	47	39	31	23	15		
$C_0 =$	1	58	50	42	34	26	18	$D_0 =$	7	62	54	46	38	30	22
	10	2	59	51	43	35	27		14	6	61	53	45	37	29
	19	11	3	60	52	44	36		21	13	5	28	20	12	4

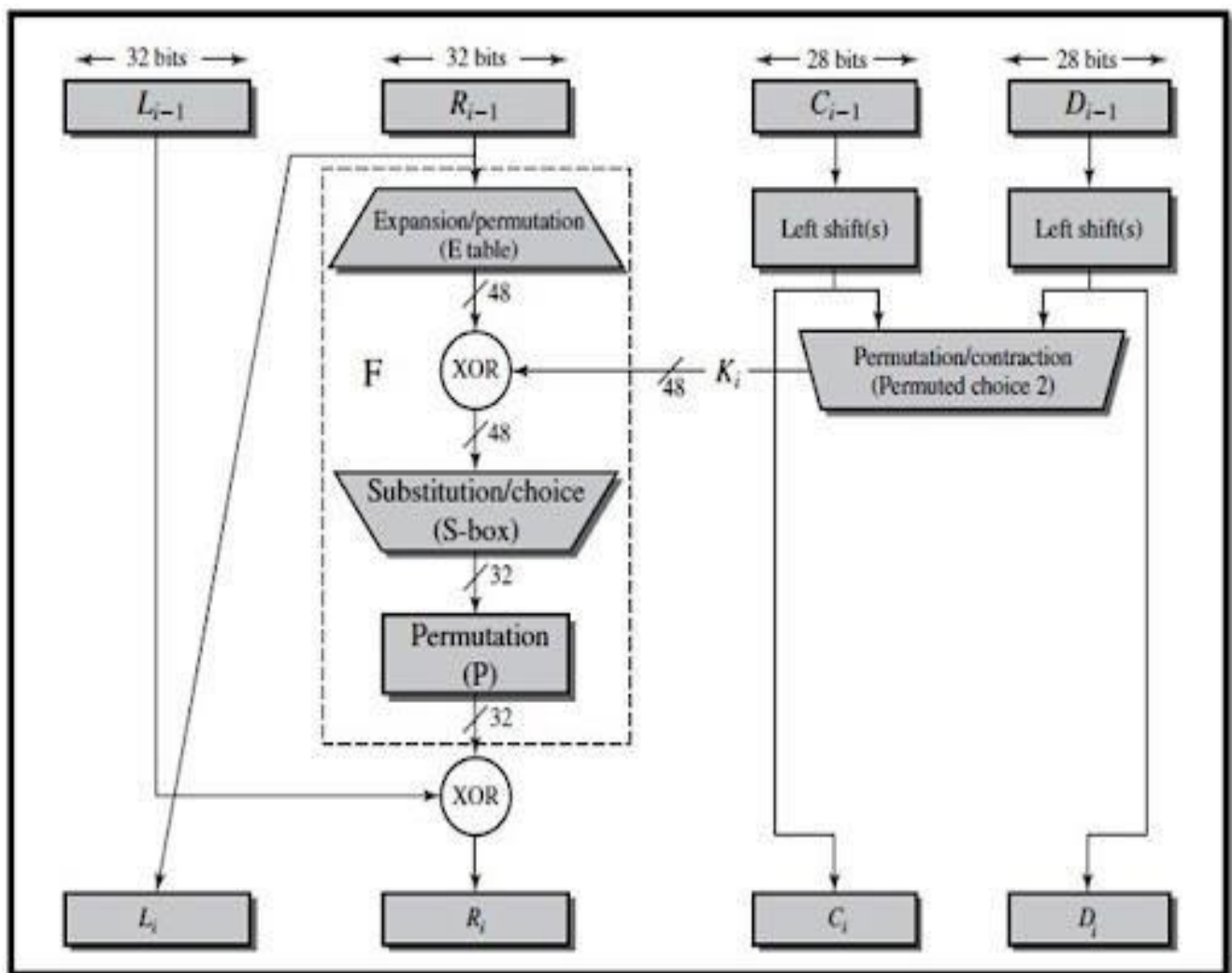
Reduced Permutation:

The two 28-bit blocks are grouped into 56-bit block. This passes through a reduced permutation giving 48-bit block output, representing the key K_i . The general format of reduced permutation function is:

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

5. Details of Single Round:

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. The following diagram describes internal structure of a single round. The 64-bit intermediate value is divided into 32-bit two halves.



The overall processing at each round can be summarized in the following formulas:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} (+) F(R_{i-1}, K_i)$$

The round key K_i is 48 bits. The R input is 32 bits. The R input is first expanded to 48 bits by using a table that defines a permutation plus an expansion that involves duplication of 16 of the R bits. The resulting 48 bits are XORed with K_i .

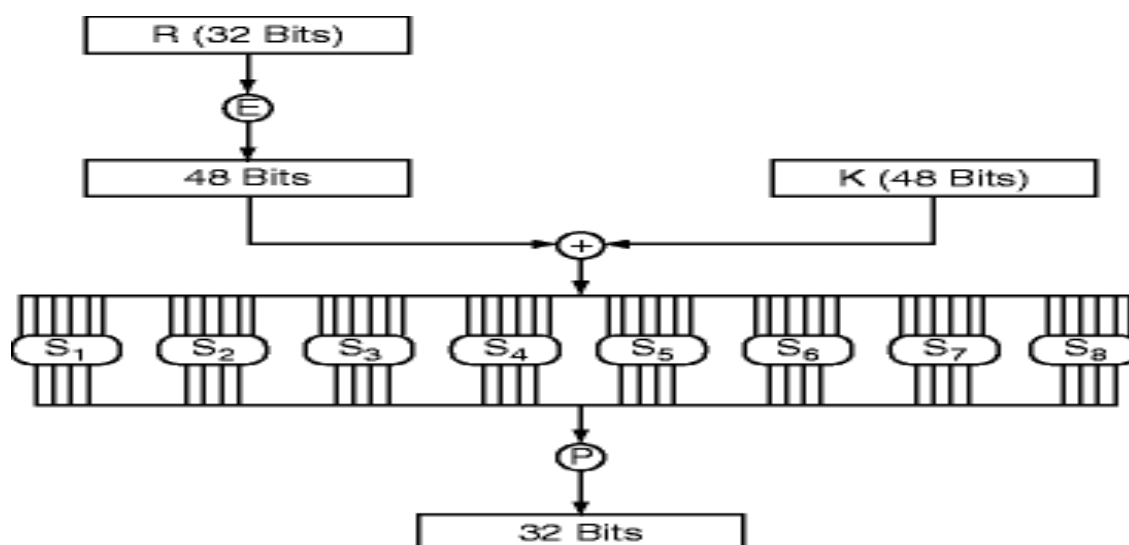
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

This 48-bit result passes through a substitution function that produces a 32-bit output, which is permuted as defined by the following table.

Permutation Function (F):

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

The role of the S-Boxes in the function F is explained in the following diagram. The substitution consists of a set of 8 S-Boxes, each of which accepts 6 bits as input and produces 4bits as output.



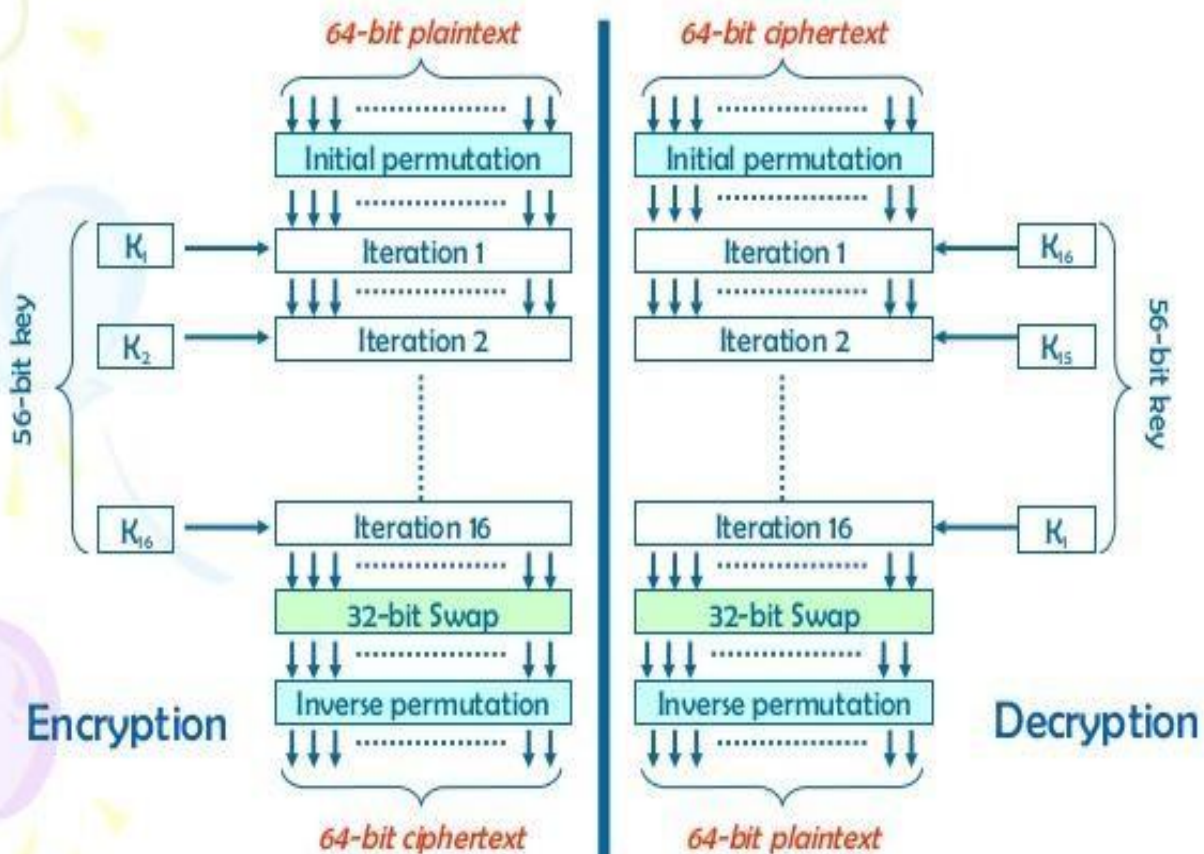
S-Box (Substitution Box):

The S-Boxes do the real mixing. DES uses 8 S-Boxes, each with 6-bit input and a 4-bit output.

DES Decryption:

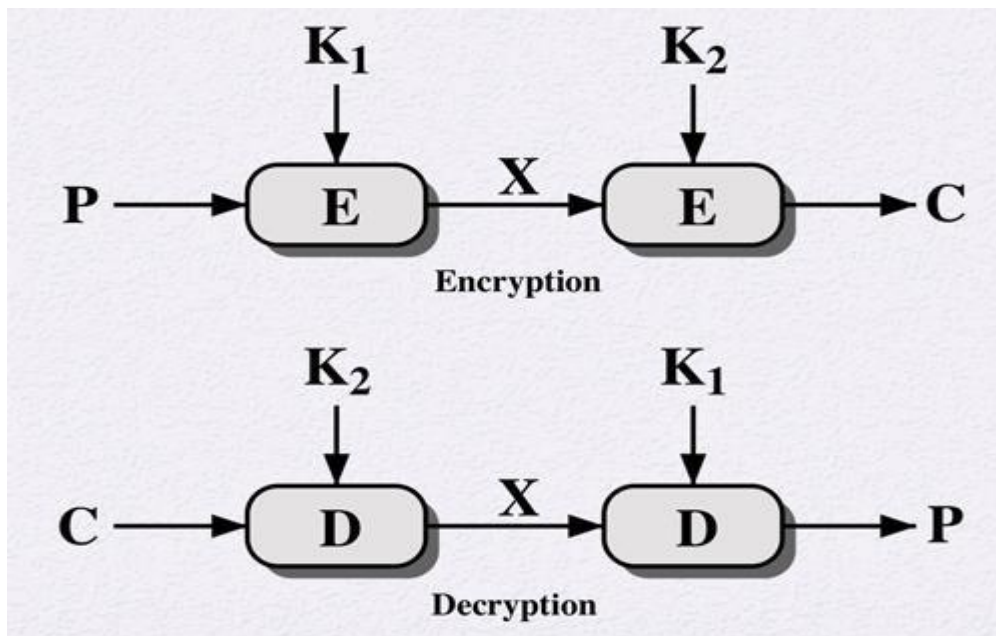
The decryption uses the same algorithm as encryption, except that the application of the sub keys is reversed.

DES Encryption & Decryption



Double DES

In Double DES we have multiple stages of encryption for the given plain text. In the double DES we use encryption twice. The following diagram explains the idea.



Encryption: $C = E_{K_2} (E_{K_1} (P))$

Decryption: $P = D_{K_1} (D_{K_2} (C))$

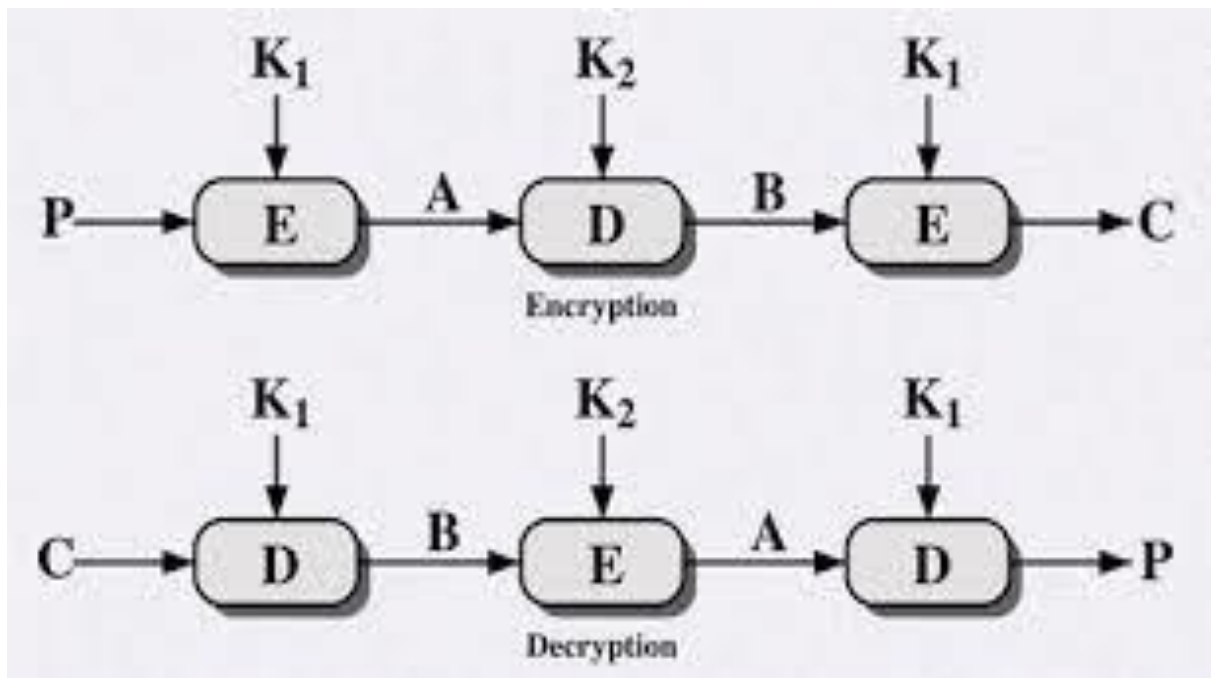
This scheme involves a key length of $56 \times 2 = 112$ bits. It will increase the key space and security.

Triple DES

As an improvement of data encryption standard in the late 1970's IBM developed the Triple DES. Triple DES is simply the DES used three times in succession. It is the successive use which makes 3DES much harder to crack than DES. 3DES solve the problem of the too-short 56 bit key length used in DES by using a key length of 168 bits.

It overcomes the drawback of Double DES algorithm.

The following diagram explains the encryption and decryption used in TDES.



The equations are:

$$C = E_{K_1} ((D_{K_2} (E_{K_1} (P))))$$

$$P = D_{K_1} ((E_{K_2} (D_{K_1} (C))))$$

IDEA

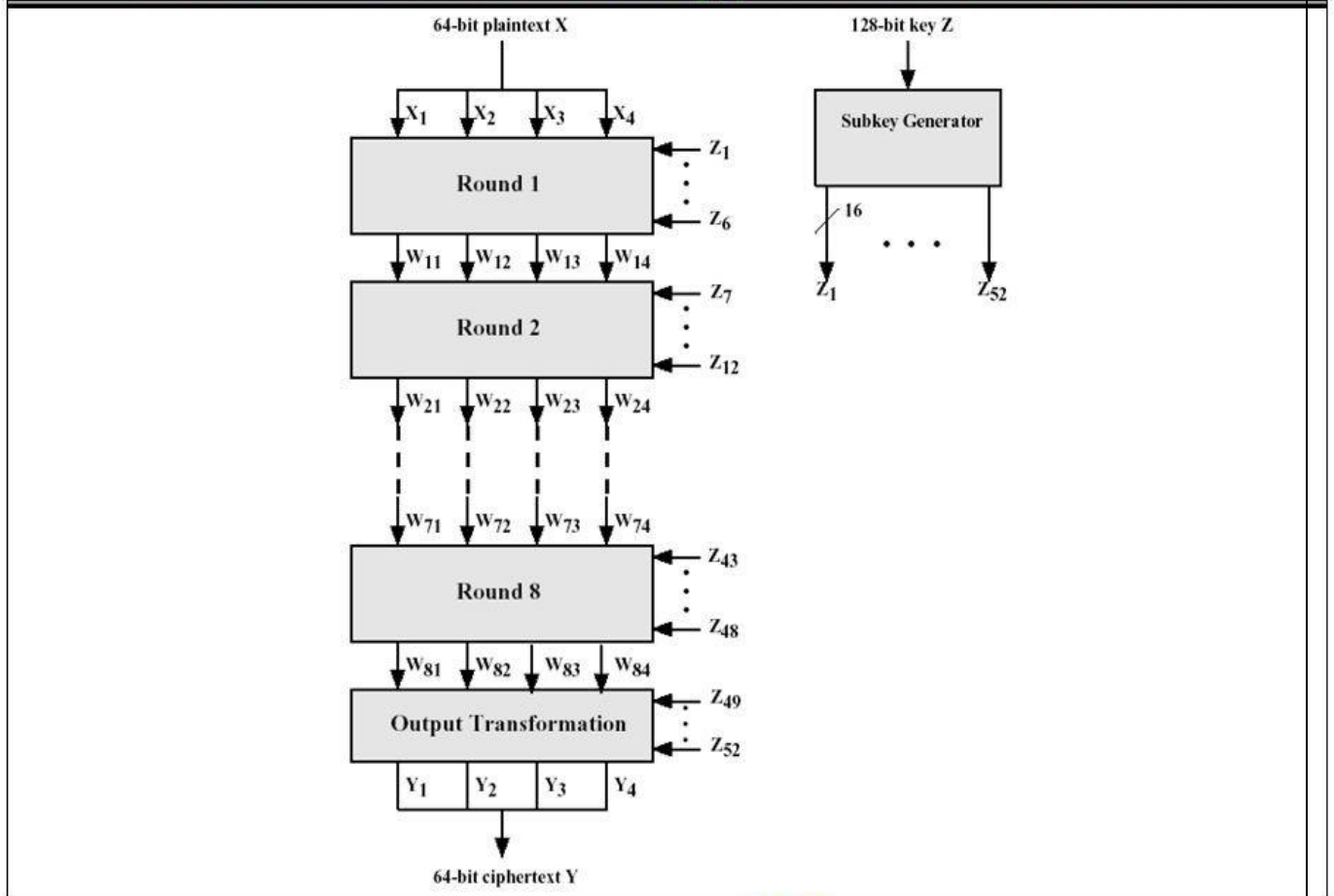
International Data Encryption Algorithm (IDEA):

In cryptography, the **International Data Encryption Algorithm (IDEA)**, originally called **Improved Proposed Encryption Standard (IPES)**, is a symmetric-key block cipher designed by James Massey of ETH Zurich and Xuejia Lai and was first described in 1991.

IDEA operates on 64-bit blocks using a 128-bit key and consists of a series of 8 identical transformations (a *round*, see the illustration) and an output transformation (the *half-round*). The processes for encryption and decryption are similar.

Here we generate 52 sub keys of 16 bits from the 128-bits key. Each round uses 6 sub keys and the output transformation uses 4 sub keys. The following is the block diagram.

Overall IDEA Encryption Structure



Explanation:

Sub key Generation: First we consider the 128-bits key. It is divided into 8 equal parts. The first part is called Z_1 , second part is called Z_2 and so on. The last part is Z_8 . In this way we generate Z_1 to Z_8 keys, i.e., $Z_1(1..16)$, $Z_2(17..32)$, $Z_3(33..48)$, $Z_4(49..64)$, $Z_5(65..80)$, $Z_6(81..96)$, $Z_7(97..112)$, $Z_8(113..128)$.

Now we perform circular left shift of 25-bits on the given key to get 26..128, 1..25. We divide these bits into 8 equal parts and call them as Z_9 to Z_{16} , i.e., $Z_9[26..41]$, $Z_{10}[42..57]$, $Z_{11}[57..73]$, $Z_{12}[74..89]$, $Z_{13}[90..105]$, $Z_{14}[106..121]$, $Z_{15}[122,123,124,125,126,127,128,1..9]$, $Z_{16}[10..25]$.

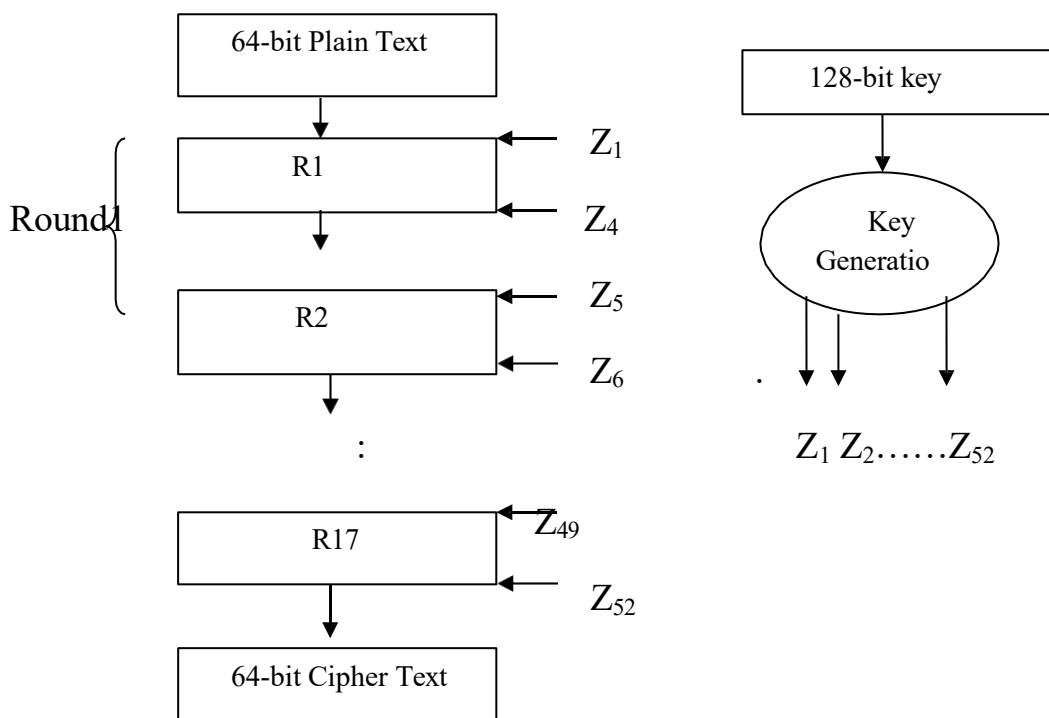
Now again we perform circular left shift of 25-bits on the above input to get

51..128, 1..25, 26..50.

From this we generate the text 8 sub keys from Z17 to Z24. Same procedure is repeated until we get 52 sub keys.

Internal Organization of Rounds:

In IDEA algorithm we have 8 rounds of similar structure and an output transformation function. Each round is further divided into 2 sub rounds. Hence we will get totally 17 different rounds. Out of these 17 rounds, all rounds having similar structure. Now the IDEA block diagram can be viewed as follows:



From the above diagram we observe that each round takes a 64-bit input and produces 64-bit output. Each odd round takes 4 keys whereas each even round takes 2 keys.

Each round takes four 16-bit values as output. The following is the structure of odd round. The odd round 'i' takes four 16-bit values namely Xa, Xb, Xc, Xd as inputs with keys Ka, Kb, Kc, Kd and produce Xa, Xb, Xc, Xd as outputs. The following diagram explains this concept.

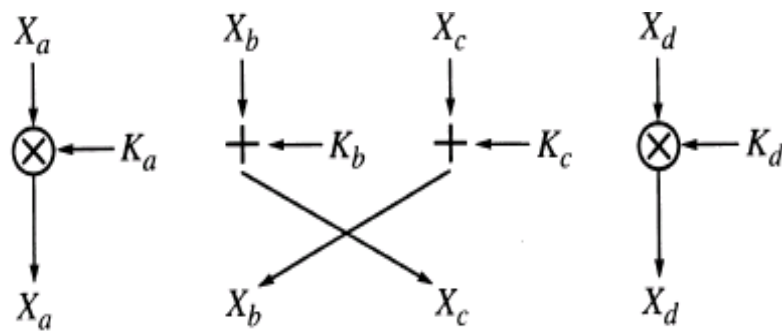


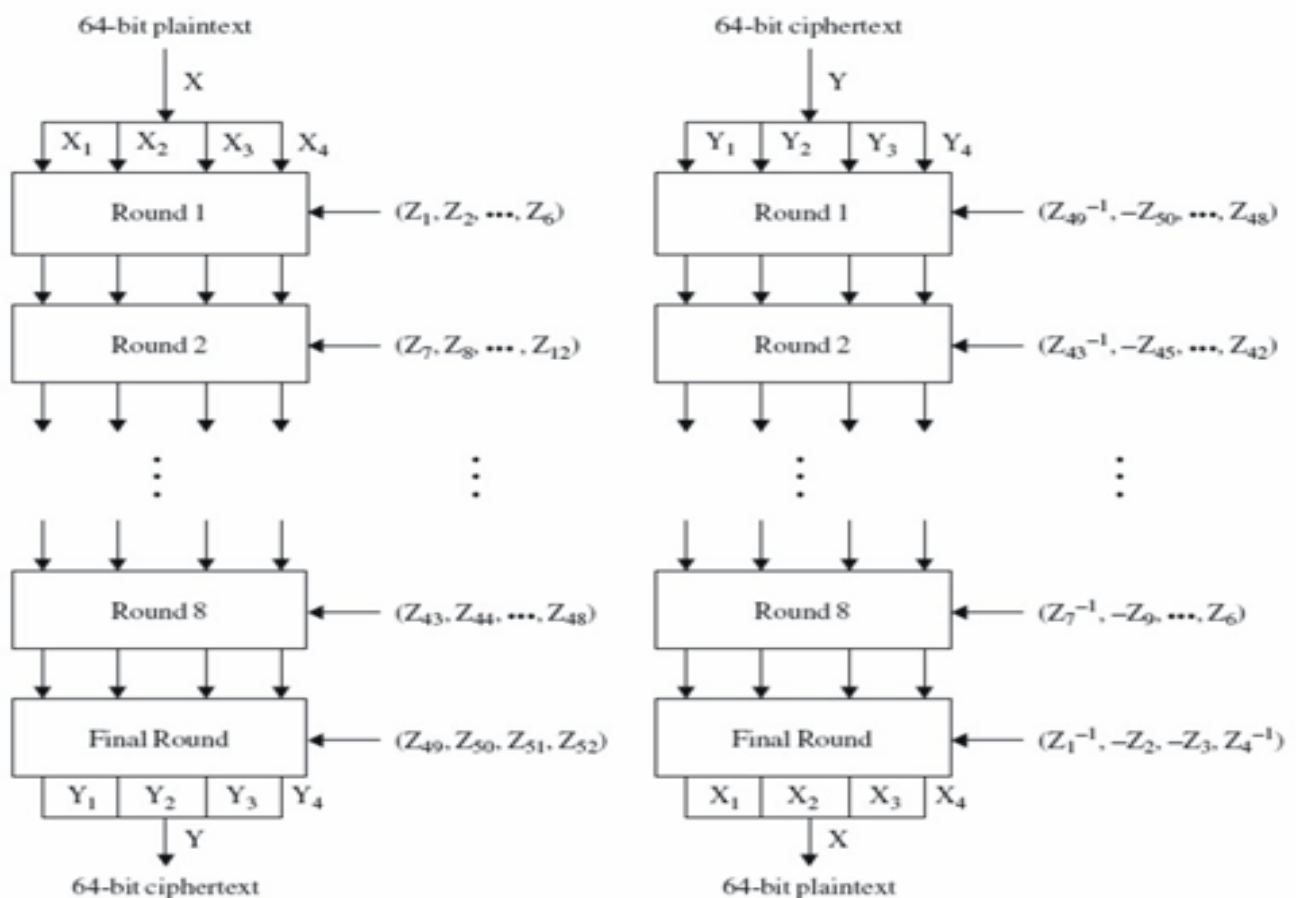
Figure 3-21. IDEA Odd Round

Structure of even round:

The even rounds takes four 16-bit inputs X_a , X_b , X_c , X_d and two key values K_e , K_f and generate four 16-bit values output as X_a , X_b , X_c , X_d .

IDEA Decryption:

In IDEA decryption the 64-bit cipher text is converted into 64-bits plain text using the same 128-bits key.



AES Algorithm

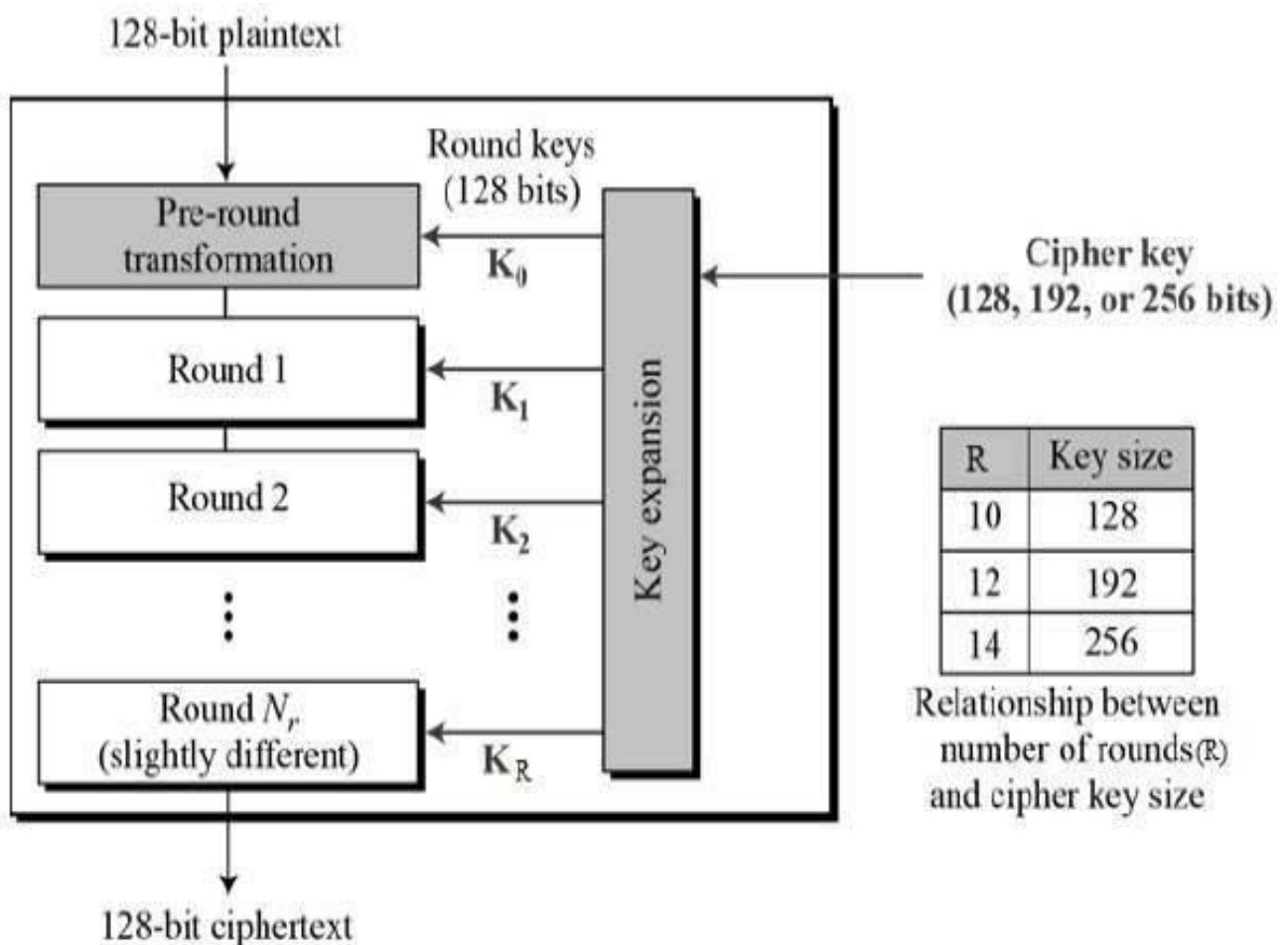
Advanced Encryption Standard (AES)

AES is an iterative rather than Feistel cipher. This block cipher algorithm is proposed by Rijndael. This is a substitution permutation network. This relatively easy to implement when compared to DES and it occupies less memory space. This algorithm uses a block of plain text and converts it into a block of cipher text using a block key.

The block length is a multiple of 32 and in between 128 bits. This algorithm supports a large key in multiples of 32. The key is also a multiple of 32, between 128 and 256 bits.

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.

Block Diagram:



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

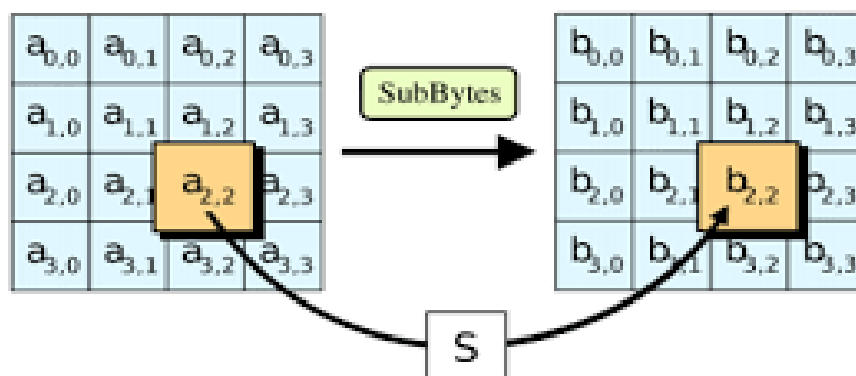
Encryption Process

Each round in the AES algorithm contains 4 stages except the last round. They are

- i) Sub Bytes Step
- ii) Shift Row Step
- iii) Mix Column Step
- iv) Add Round key Step

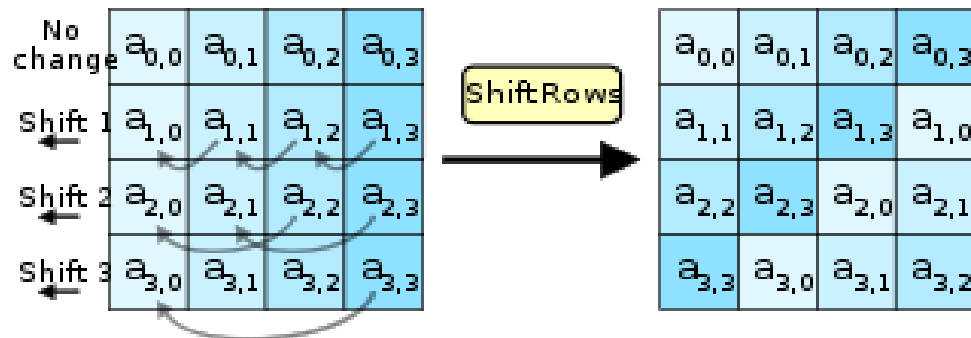
i) Sub Bytes Step:

In sub bytes step each byte of the input is updated using S-box to get an output byte. This operation provides a non-linearity in the input. The S-box is generally derived from the inverse functions. The following diagram explains this idea.



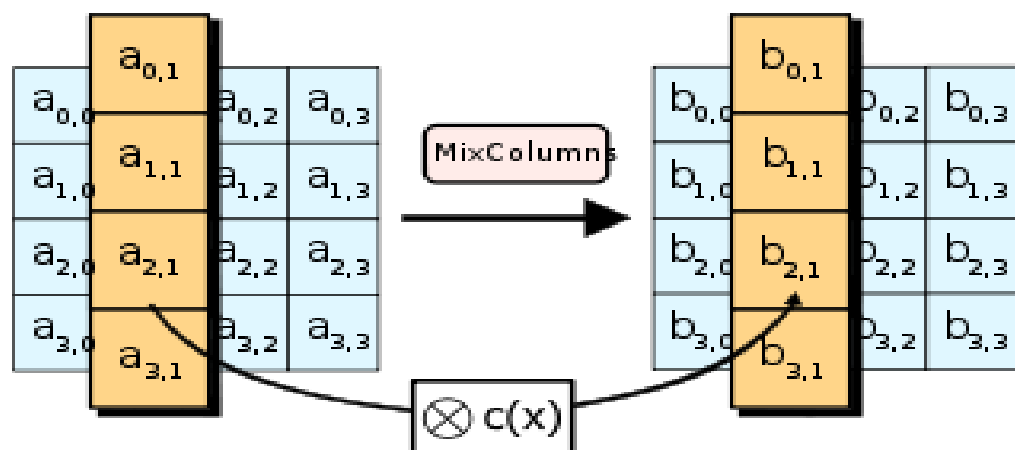
ii) Shift Row Step:

The output of first stage is given to shift rows step. It operates on the rows of the input. The row 1 is unchanged. The second row is shifted one to its left. The third row is shifted two to its left. The fourth row is shifted three to its left. This is shown in the following diagram.



iii) Mix Column Step:

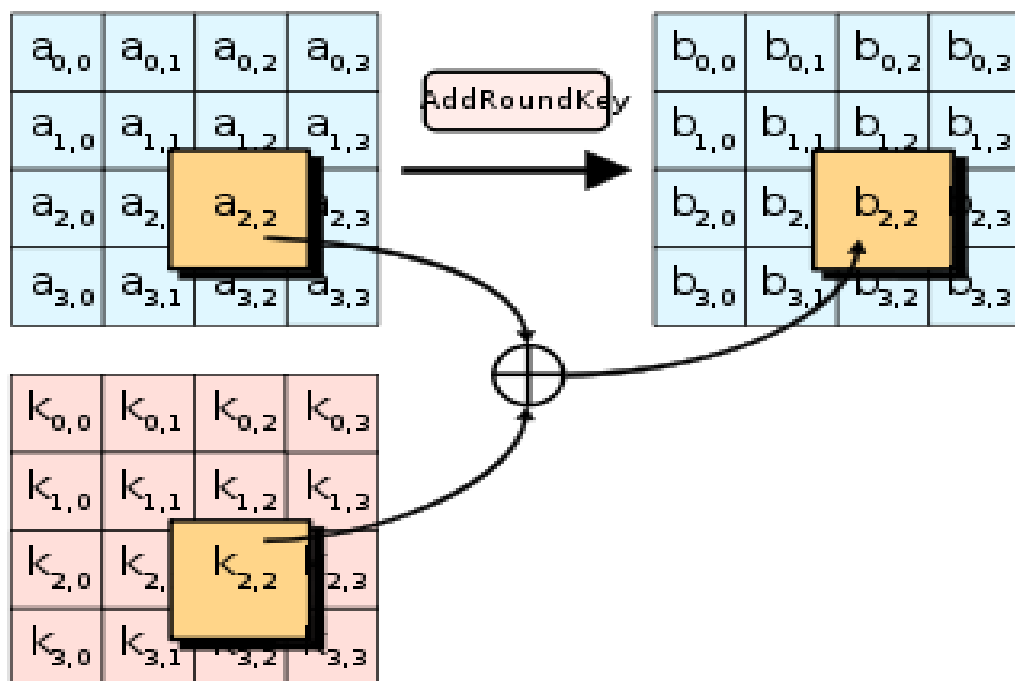
In this step the 4 bytes of each column are converted into transformations and multiply with a fixed polynomial (eg. x^4+1) to get the output. The input to this stage is 4 * 4 array of bytes (output of the shift rows). From this input, we get four groups. Each group is transformed into polynomial. Each such polynomial is multiplied with x^4+1 to get another polynomial. It is again retransformed to a group of 4-bytes.



iv) Add Round key Step:

In this step sub or round key is combined with each byte of the input to get the output. We require a sub key of 4×4 arrays of bytes for each round. These sub keys are generated from the given 128-bits key using key scheduler process. Each byte of the sub key is Exclusive Ored with corresponding byte of the input to get an output byte.

The last round does not have Mix Column step. But it has the remaining 3 stages in the same order.



AES Encryption Structure:

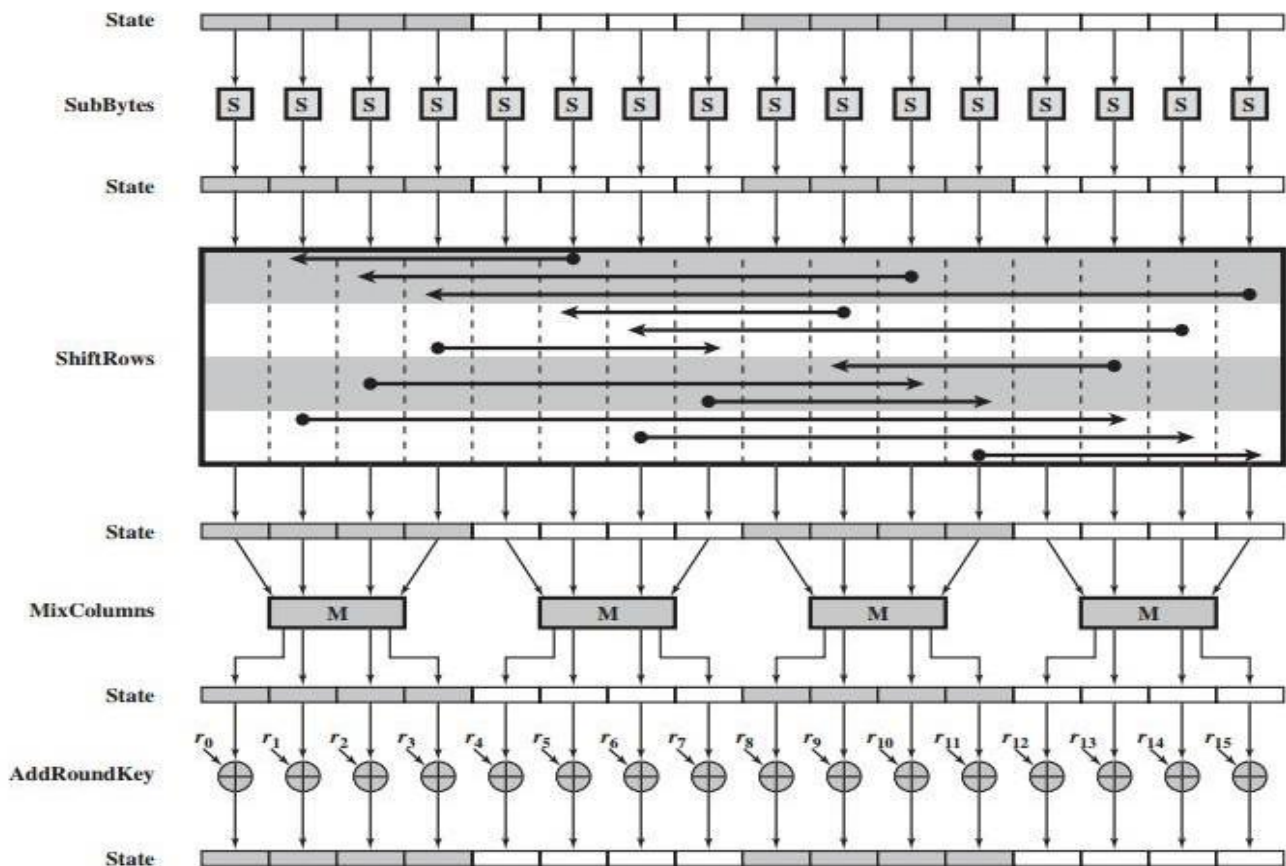


Figure 5.4 AES Encryption Round

AES Key Expansion or Sub Keys Generation:

The 128 bit key is expanded into 11 arrays of the same size as the state. The arrays are stored in array of structure (rk) named from rk[0] to rk[10]. Before expansion rk[0] is XORed into state byte-for-byte i.e., 16 bytes in a state is replaced by the XOR of itself and the corresponding byte in rk[0]. The other 10 arrays will be used during rounds with one array per round. These round keys are produced by repeated rotation and XORing of various groups of key bits.

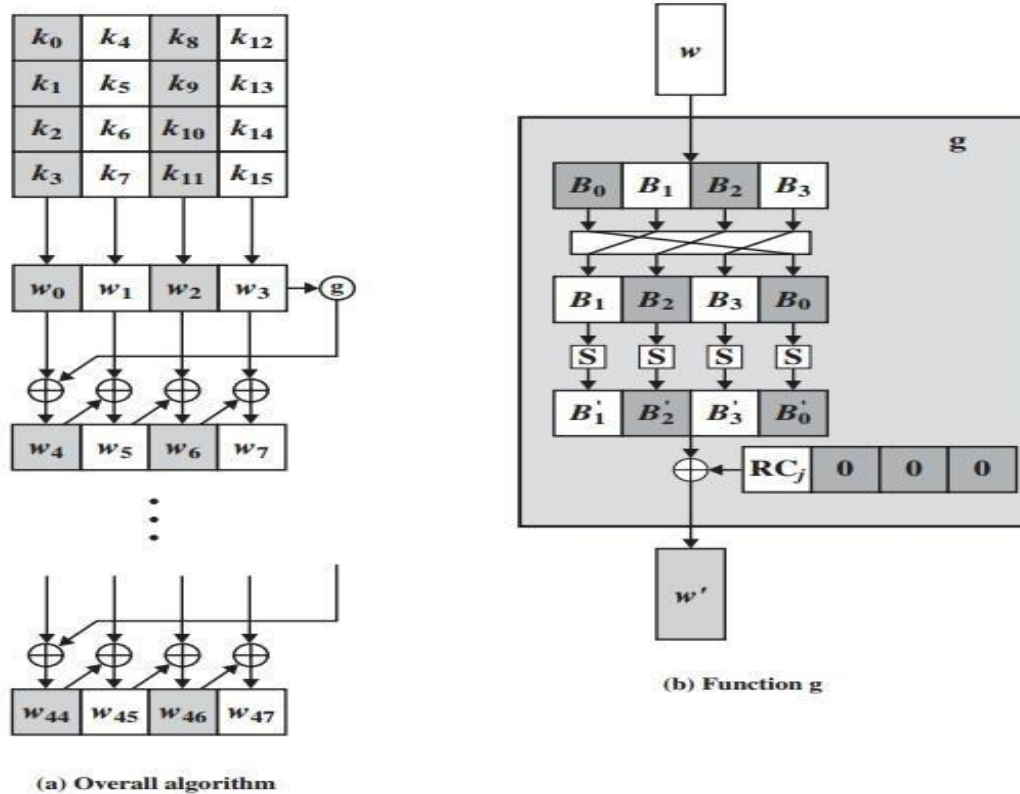


Figure 5.9 AES Key Expansion

Decryption Process

The process of decryption of an AES cipher text 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

Blowfish Algorithm

Blowfish Algorithm is developed by Bruce Schneier. Blowfish is a symmetric block encryption algorithm designed in consideration with:

1. **Fast:** It encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles per byte.
2. **Compact:** It can run in less than 5K memory.
3. **Simple:** It uses addition, XOR, lookup table with 32-bit operations.
4. **Secure:** The key length is variable, it can be in the range of 32 – 448 bits:
default 128 bits key length.

Description of Algorithm:

Blowfish encrypts 64-bit blocks of plain text into 64-bit blocks of cipher text. Blowfish symmetric block cipher algorithm encrypts block data of 64-bits at a time. It will follow the Feistel network and this algorithm is divided into two parts.

a. Key-Expansion

b. Data Encryption

a. Key- Expansion: It will convert a key of at most 448 bits into several sub key arrays totally 4168 bytes. Blowfish uses large number of sub keys. These keys are generated earlier to any data encryption or decryption.

The p-array consists of 18, 32-bit sub keys:

P_1, P_2, \dots, P_{18}

Four 32-bit S-Boxes consist of 256 entries each:

$S_{1,0}, S_{1,1}, \dots, S_{1,255}$

$S_{2,0}, S_{2,1}, \dots, S_{2,255}$

$S_{3,0}, S_{3,1}, \dots, S_{3,255}$

$S_{4,0}, S_{4,1}, \dots, S_{4,255}$

Generating the Sub Keys:

The subkeys are calculated using the Blowfish algorithm:

1. Initialize first the P-array then the four S-Boxes, in order, with a fixed string. This string consists of the hexadecimal digits of a constant pi (less than initial 3):
 $P_1=243F6A88, P_2=85a308d3, P_3=13198a2e, P_4=03707344$, etc.
2. XOR P_1 with the first 32 bits of the key, XOR P_2 with the second 32-bits of the key, and so on for all bits of the key. Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits.
3. Encrypt the all-zero string with the Blowfish algorithm, using the sub keys described in steps (1) and (2).
4. Replace P_1 and P_2 with the output of step (3).
5. Encrypt the output of step (3) using the Blowfish algorithm with the modified sub keys.
6. Replace P_3 and P_4 with the output of step (5).
7. Continue the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm.

In total, 521 iterations are required to generate all required sub keys. Applications can store the sub keys rather than execute this derivation process multiple times.

The update process can be summarised as follows:

$$P_1, P_2 = E_{P,S}[0]$$

$$P_3, P_4 = E_{P,S}[P_1 \parallel P_2]$$

...

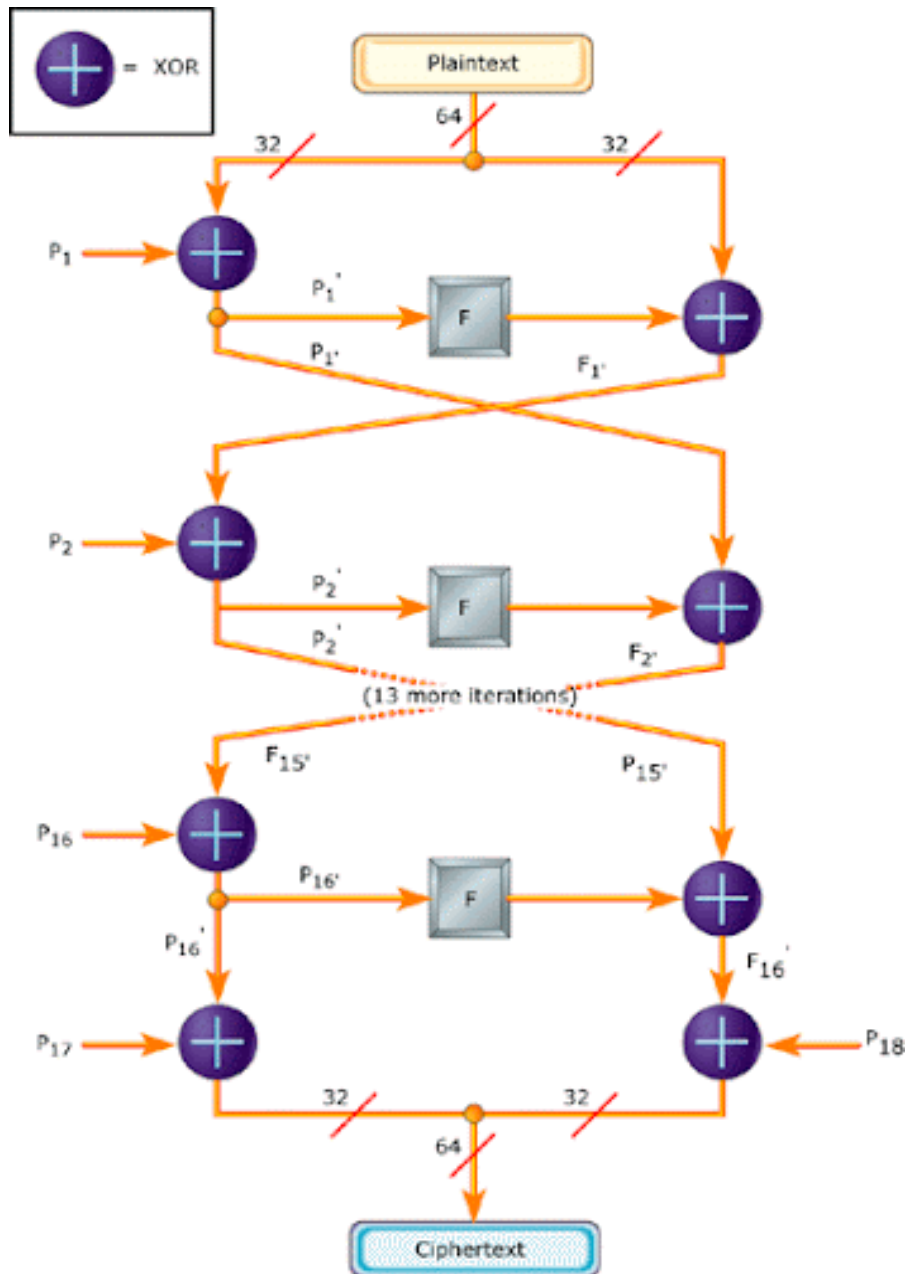
$$P_{17}, P_{18} = E_{P,S}[P_{15} \parallel P_{16}]$$

$$S_{1,0}, S_{1,1} = E_{P,S}[P_{17} \parallel P_{18}]$$

...

$$S_{4,254}, S_{4,255} = E_{P,S}[S_{4,252} \parallel S_{4,253}]$$

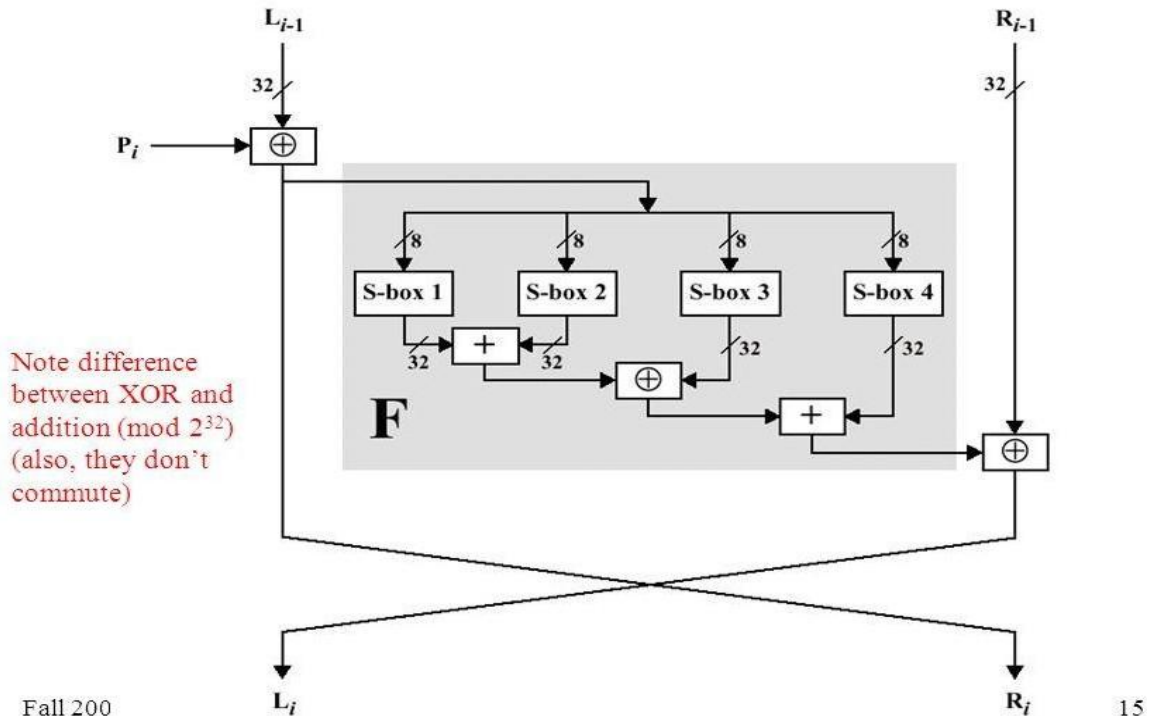
b. Data Encryption: It is having a function to iterate 16 times of network. Each round consists of key-dependent permutation and a key and data-dependent substitution. All operations are XORed and additions on 32-bit words. The only additional operations are four indexed array data lookup tables for each round.



The resulting cipher text is contained in the two variables. The function F is shown in the following figure. The 32-bit input to F is divided into 4 bytes. If we label those bytes a , b , c and d , then the function can be defined as follows:

$$F[a, b, c, d] = ((S_{1,a} + S_{2,b}) (+) S_{3,c}) + S_{4,d}$$

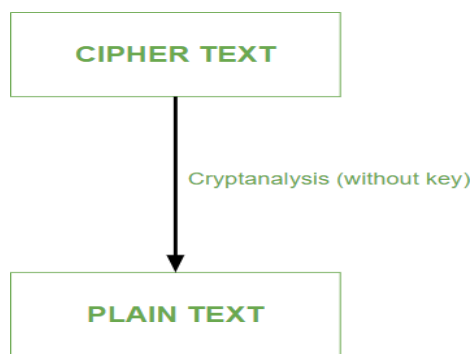
Single Blowfish Round



Decryption: In decryption, the 64-bits of cipher text are initially assigned to the two one-word variables. Blowfish decryption involves using the sub keys in reverse order.

Differential and Linear Cryptanalysis

Cryptanalysis: Cryptanalysis is the process of transforming or decoding communications from non-readable to readable format without having access to the real key. **Cryptoanalysis** is the art, science, or practice of decrypting encrypted messages. The secret key used for encryption and decoding is considered to be unknown to the cryptologists, mathematicians, and other scientists participating in the process. In contrast to a brute force attack, this form of analysis seeks vulnerabilities in a cryptosystem.



Cryptanalysis is used to break cryptographic security systems and gain access to the contents of the encrypted messages, even if the cryptographic key is unknown.

Types of Cryptanalytic Attacks:

1. Ciphertext only attack:

1. In this type of cryptanalytic attack, the attacker has the knowledge of only the ciphertext.
2. The attacker has to detect the plain text using the ciphertext only.
3. This type of attack is not very easy to be implemented.

2. Known plain text only attack:

1. In this type of cryptanalytic attack, the attacker has the knowledge of some plain text as well as ciphertext.
2. The attacker tries to decrypt the messages using these two.
3. This type of attack is somewhat easy to implement.

Different Forms of Cryptanalysis:

Cryptanalysis basically has two forms:

1. Linear Cryptanalysis:

Linear cryptanalysis is a general type of cryptanalysis based on discovering affine approximations to a cipher's action in cryptography. Block and stream ciphers have both been subjected to attacks. Linear cryptanalysis is one of the two most common attacks against block ciphers, with differential cryptanalysis being the other.

2. Differential Cryptanalysis:

Differential cryptanalysis is a sort of cryptanalysis that may be used to decrypt both block and stream ciphers, as well as cryptographic hash functions. In the widest sense, it is the study of how alterations in information intake might impact the following difference at the output. In the context of a block cipher, it refers to a collection of strategies for tracking differences across a network of transformations, finding where the cipher displays non-random behavior, and using such attributes to recover the secret key (cryptography key).

Difference between Linear Cryptanalysis and Differential Cryptanalysis

S.No.	Linear Cryptanalysis	Differential Cryptanalysis
1.	Linear cryptanalysis was basically invented by Matsui and Yamagishi in the year 1992.	Differential cryptanalysis was first defined in the year 1990 by Eli Biham and Adi Shamir.
2.	Linear cryptanalysis always works on a single bit (one bit at a time).	Differential cryptanalysis can work on multiple bits at a time.
3.	In the case of Linear cryptanalysis, ciphertext attack is a very big disadvantage.	In the case of differential cryptanalysis plain text attack is a very big disadvantage.

S.No. Linear Cryptanalysis

Differential Cryptanalysis

4. The use of linear cryptanalysis is to figure out what is the linear relationship present between some plaintext bits, ciphertext bits, and unknown key bits very easily.
5. Subsets of input attributes refer to the internal structures of a single input.
6. The cryptanalyst decrypts each ciphertext using all available subkeys and analyses the resultant intermediate ciphertext to determine the random outcome for one encryption cycle.

The use of differential cryptanalysis is to get clues about some critical bits, reducing the need for an extensive search.

The underlying structure of each individual input is unimportant in this case since the input attributes are differential.

After several encryption rounds, Cryptanalyst analyses the changes in the intermediate ciphertext obtained. The practice of combining assaults is known as differential linear cryptanalysis.

Asymmetric Key Cryptographic Algorithms

Overview of Asymmetric key Cryptography

Asymmetric Key Cryptography: The encryption process where different keys are used for encrypting and decrypting the information is known as Asymmetric Key Encryption. This is also called Public Key Cryptography. In this two key are used, Public Key and Private Key. Though the keys are different, they are mathematically related.

Any public key cryptographic algorithm has six elements as follow:

1. **Plain Text:** This is a readable message which is given as input to the algorithm. In a public key algorithm, the plain text is encrypted in blocks.
2. **Encryption Algorithm:** The encryption algorithm is implemented on the plain text which performs several transformations on plain text.
3. **Public and Private keys:** These are the set of keys among which if one is used for encryption the other would be used for decryption. The transformation of plain text by encryption algorithm depends on the key chosen from the set to encrypt the plain text.
4. **Cipher Text:** This is the output of encryption algorithm. The generated cipher text totally depends on the key selected from the set of the public and private key. Both of these keys, one at a time with plain text would produce different cipher texts.
5. **Decryption Algorithm:** This would accept the output of the encryption algorithm i.e. the cipher text and will apply the related key to produce the original plain text.

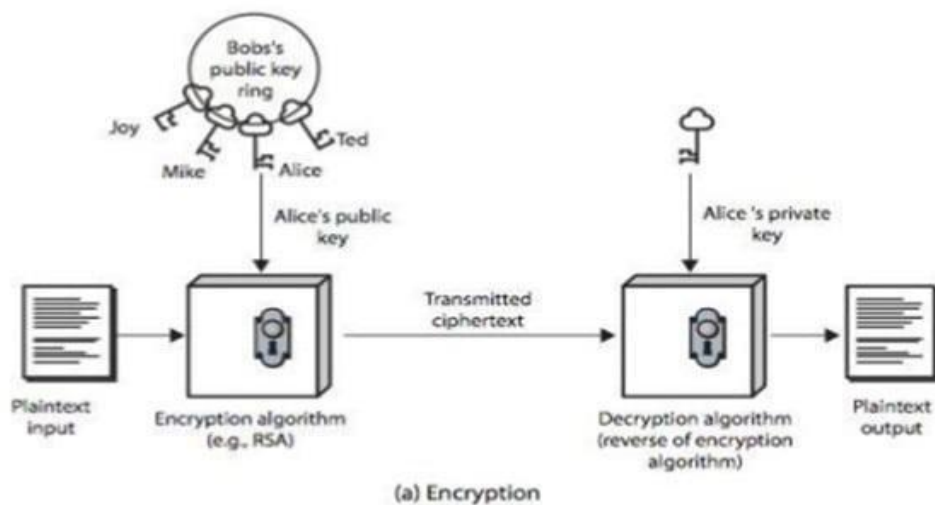
The essential steps are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private. As shown in the following figure, each user maintains a collection of public keys obtained from others.
3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.

4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

The process is depicted in the following illustration –

Asymmetric Cryptography



Asymmetric key algorithms are not quite as fast as symmetric key algorithms. This is partially due to the fact that asymmetric key algorithms are generally more complex, using a more sophisticated set of functions.

Asymmetric Key Algorithms

Asymmetric key algorithms aren't as widely used as their symmetric counterparts. So we'll just go over two of the big ones: Diffie-Hellman and RSA.

RSA Algorithm

RSA (Rivest–Shamir–Adleman) is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. Asymmetric means that there are two different keys. This is also called public key cryptography, because one of the keys can be given to anyone.

The RSA was developed in 1977 by Ron Rivest, Adi Shamir, Len Adleman at MIT. Since then, the Rivest-Shamir-Adleman (RSA) scheme has become the most widely accepted and implemented general purpose approach to public-key encryption.

This algorithm is used to encrypt integer data. In the RSA algorithm the integer message 'M' is encrypted by using the following equation:

$$C = M^e \bmod n$$

The receiver uses the following equation for decryption:

$$M = C^d \bmod n$$

Here both sender and receiver must know the integer value 'n'. The sender uses the public key $KU = \{e, n\}$ The receiver uses the private key $KR = \{d, n\}$.

The following are the requirements for the RSA algorithm:

- a) It is possible to find the values for e, d, n such that $M = M^{ed} \bmod n$
- b) It is easy to calculate M^e, C^d for all M, C.
- c) It is computationally infeasible to find 'd' even $\{e, n\}$ are known.

The RSA Algorithm is stated as follows:

Key Generation

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

Encryption

Plaintext	$M < n$
Ciphertext	$C = M^e \pmod{n}$

Decryption

Ciphertext	C
Plaintext	$M = C^d \pmod{n}$

Example 1:

1) Key Generation:

- i) Select two prime numbers p, q . Let $p = 7$ and $q = 17$
- ii) Calculate $n = 7 * 17 = 119$
- iii) Calculate $\phi(n) = (7-1) * (17-1) = 6 * 16 = 96$
- iv) Select 'e' such that $\gcd(\phi(n), e) = 1 \Rightarrow \gcd(96, e) = 1$ and let $e = 5$
- v) Calculate 'd' such that $de = 1 \pmod{\phi(n)}$
 $\Rightarrow de \pmod{\phi(n)} = 1$
 $\Rightarrow d5 \pmod{96} = 1$
 $\Rightarrow 5 * 77 \pmod{96} = 1$

vi) Form public key $KU = \{5, 119\}$

vii) Form private key $KR = \{77, 119\}$

Assume that $M = 19$

(Sender) Now $C = M^e \bmod n = 19^5 \bmod 119 = 66$

Now the receiver performs decryption in the following way:

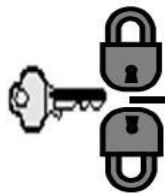
$$M = C^d \bmod n = 66^{77} \bmod 119 = 19$$

Calculation of $19^5 \bmod 119$:

$$19^2 \bmod 119 \Rightarrow 361 \bmod 119 = 4$$

$$19^5 = (19^2 * 19^2 * 19) = (4 * 4 * 19) \bmod 119 = 304 \bmod 119 = 66$$

Example 2:



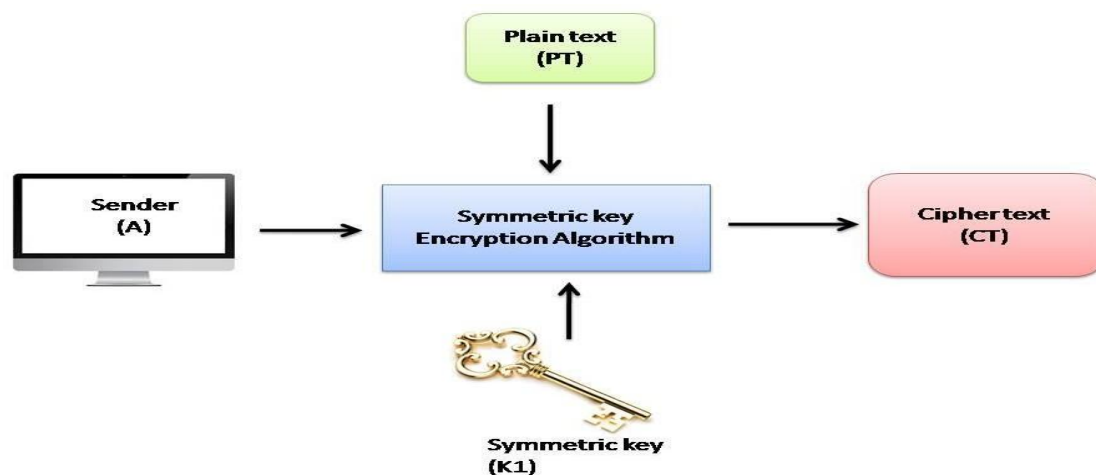
Examples on RSA

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

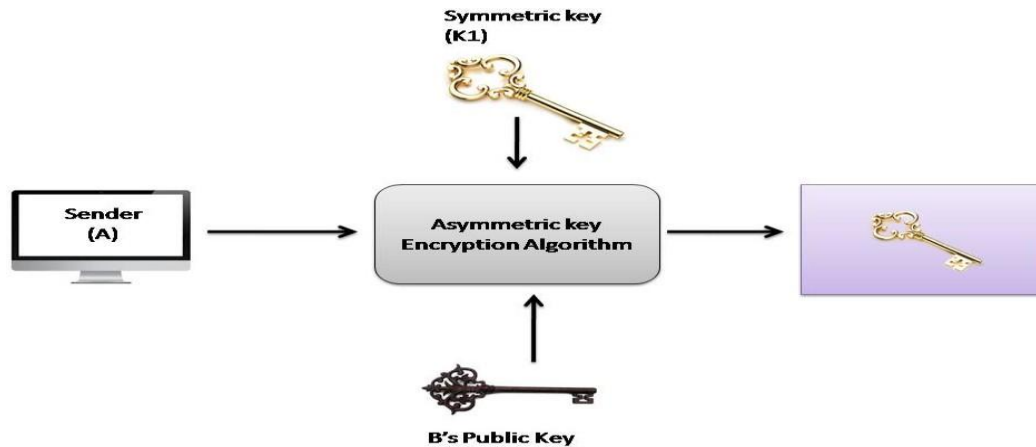
Symmetric and asymmetric key cryptography together

Symmetric-key cryptography and asymmetric-key cryptography are combined to have a very efficient security solution. The way it works is as follows, assuming that A is the sender of message and B is its receiver.

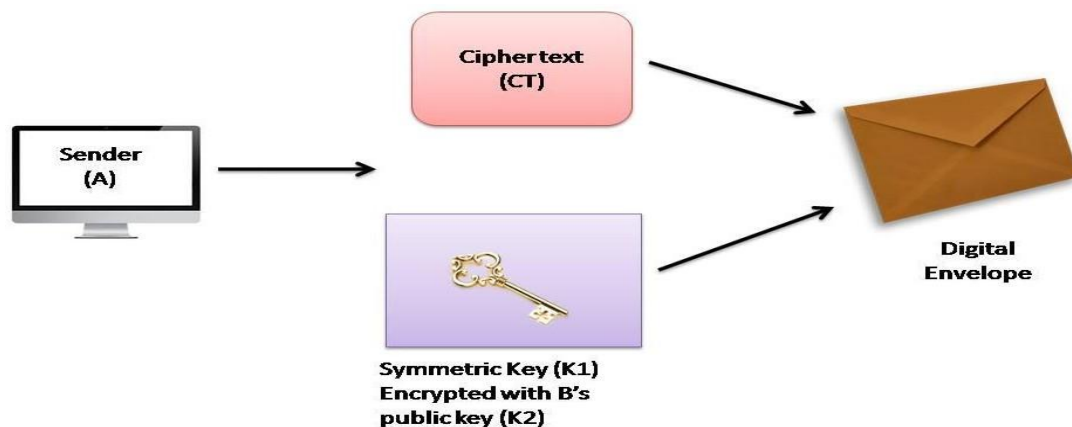
1. A's computer encrypts the original plain-text message (PT) with the help of a standard symmetric key cryptography algorithm, such DES, IDEA or RC5, etc. this produces a cipher-text message (CT) as shown in Fig. below. The key used in this operation (K1) is called one-time symmetric key, as it is used once and then discarded.



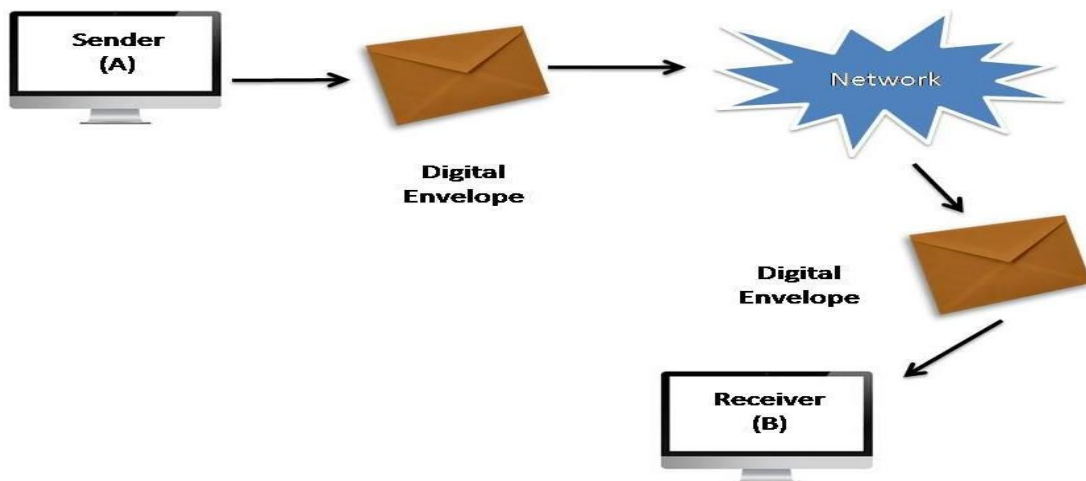
2. We would now think, we are back to square one! We have encrypted the plain text (PT) with a symmetric-key operation. We must now transport this one-time symmetric key (K1) to the server so that the server can decrypt the cipher text (CT) to get back the original plain-text message (PT). Does this not again lead us to the key-exchange problem? Well, a novel concept is used now. A now takes the one-time symmetric key of step 1 (i.e. K1), and encrypts K1 with B's public key (K2). This process is called key wrapping of the symmetric key, and is shown in fig. below. We have shown that the symmetric key K1 goes inside a logical box, which is sealed by B's public key (i.e. K2).



3. Now, A puts the cipher text CT1 and the encrypted symmetric key together inside a digital envelope. This is shown in fig. below



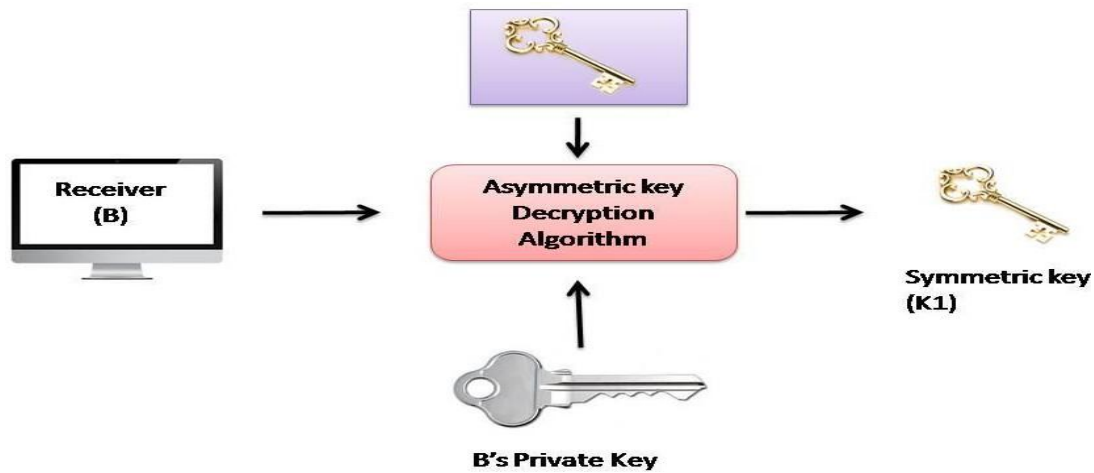
4. The sender (A) now sends the digital envelope [which contains the cipher text (CT) and the onetime symmetric key (K1) encrypted with B's public key, (K2)] to B using the underlying transport mechanism (network). This is shown in fig. we do not show the contents of the envelope, and assume that the envelope contains the two entities, as discussed.



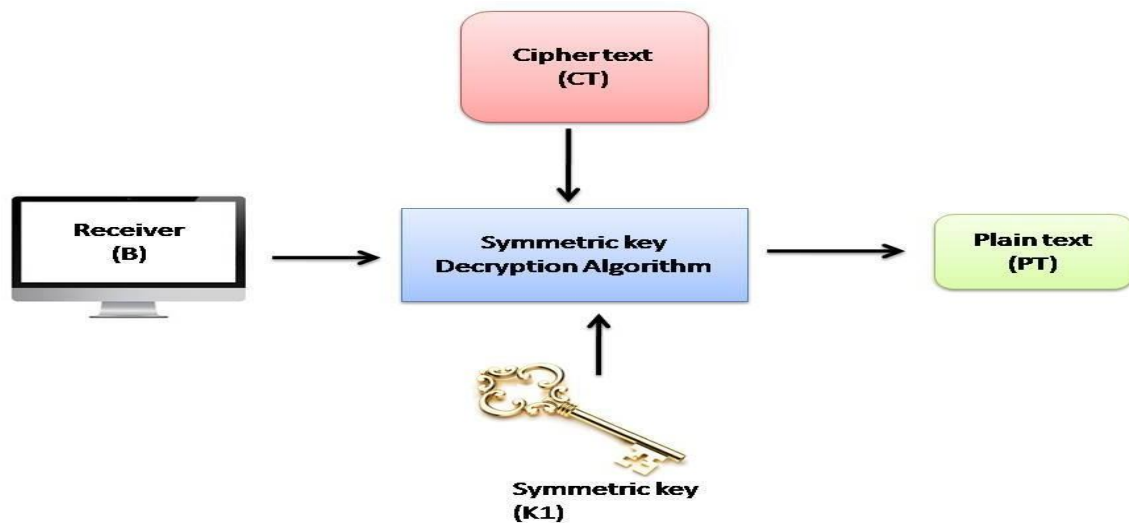
5. B receives digital envelope and opens it . After B opens this digital envelope, he gets 2 things first is cipher text (CT) and another one is the one-time session key (K1) which is encrypted using B's public key (K2). This is shown in fig. below.



6. B now uses the same asymmetric-key algorithm as was used by A and her private key (K3) to decrypt (i.e. open up) the logical box that contains the symmetric key (K1), which was encrypted with B's public key (K2). This is shown in fig. below. This the output of the process is the one-time symmetric key K1.



6. Finally, B applies the same symmetric-key algorithm as was used by A, and the symmetric key K1 to decrypt the cipher text (C1). This process yields the original plain text (PT), as shown in fig. below.



Digital Signatures

Digital Signature

A digital signature is proof of the authenticity of digital messages, files, executables, and documents etc. in modern communication - by employing strong cryptographic techniques such as encryption and hashing. A unique digital ID (and a signing key) is required to create a digital signature. Digital IDs or certificates are based on asymmetric (or public key) cryptography which use key pairs (combination of public and private keys). Public keys are openly distributed among the communicating entities and the private (signing) keys are kept secret. The most common public key algorithm used is RSA.

The digital signature is analogous to the handwritten signature. It must have the following properties:

- It must verify the author and the date and time of the signature.
- It must authenticate the contents at the time of the signature.
- It must be verifiable by third parties, to resolve disputes.

Thus, the digital signature function includes the authentication function.

Services provided by Digital Signature

- Sender/Source Authentication
- Message Integrity
- Non-Repudiation

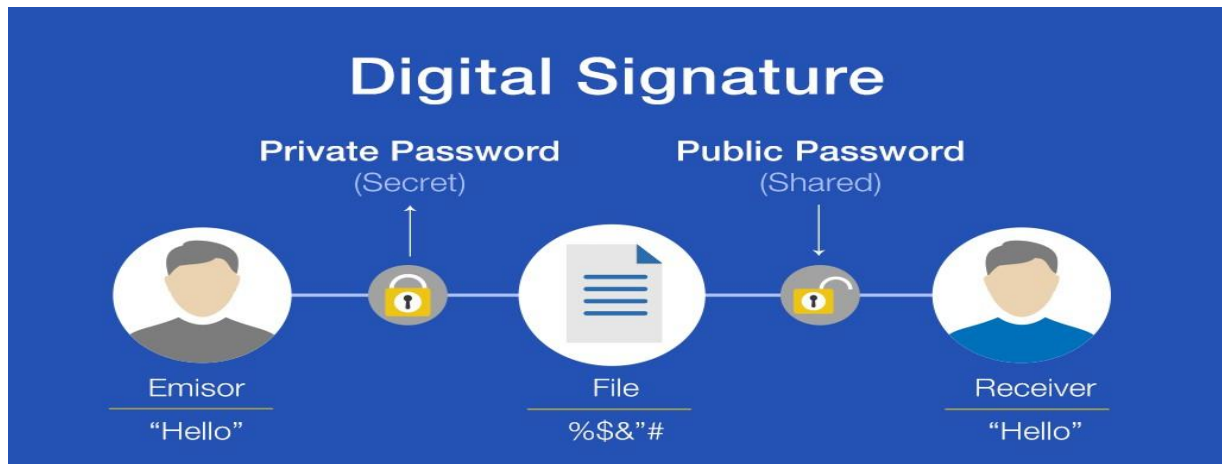
Process of Digital Signing and Verification

Key Generation: A key pair (private and public key) is generated by employing a public key algorithm such as RSA.

Digital Signature Generation: Compute the cryptographic hash of the input file or message by using a secure hashing algorithm such as SHA-256, SHA-384, SHA-512 or Whirlpool etc. The computed hash is encrypted by the private key of sender resulting in the digital signature. The digital signature is sent along with the original file for its verification by the receiver.

Digital Signature Verification: Sender computes the cryptographic hash of the received file or message by using the same hashing algorithm. Sender decrypts the

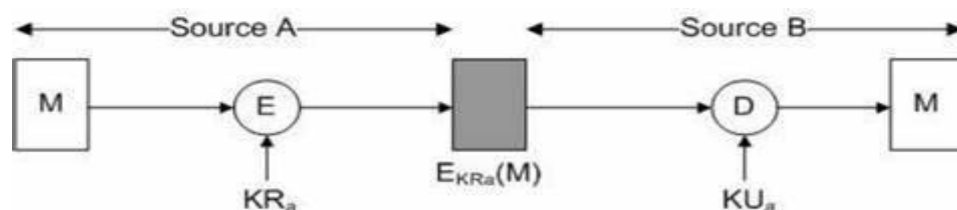
digital signature by using the public key of the sender to get the hash sent by the sender. If the decrypted hash is exactly same as the computed hash of the received file, it means that digital signature is verified and data was not tampered with during transit.



A variety of approaches has been proposed for the digital signature function. These approaches fall into two categories: direct and arbitrated.

Direct Digital Signature: The direct digital signature involves only the communicating parties (source, destination). It is assumed that the destination knows the public key of the source. A digital signature may be formed by encrypting the entire message with the sender's private key (Figure 1) or by encrypting a hash code of the message with the sender's private key.

Figure:



Arbitrated Digital Signature: The problems associated with direct digital signatures can be addressed by using an arbiter. As with direct signature schemes, there is a variety of arbitrated signature schemes.

ARBITRATED DIGITAL SIGNATURE TECHNIQUES

Table 13.1 Arbitrated Digital Signature Techniques

(a) Conventional Encryption, Arbiter Sees Message	
(1) X® A:	$M \parallel E_{K_{xa}}[ID_X \parallel H(M)]$
(2) A® Y:	$E_{K_{ay}}[ID_X \parallel M \parallel E_{K_{xa}}[ID_X \parallel H(M)] \parallel T]$
(b) Conventional Encryption, Arbiter Does Not See Message	
(1) X® A:	$ID_X \parallel E_{K_{xy}}[M] \parallel E_{K_{xa}}[ID_X \parallel H(E_{K_{xy}}[M])]$
(2) A® Y:	$E_{K_{ay}}[ID_X \parallel E_{K_{xy}}[M] \parallel E_{K_{xa}}[ID_X \parallel H(E_{K_{xy}}[M])]] \parallel T$
(c) Public-Key Encryption, Arbiter Does Not See Message	
(1) X® A:	$ID_X \parallel E_{K_{Rx}}[ID_X \parallel E_{K_{Uy}}(E_{K_{Rx}}[M])]$
(2) A® Y:	$E_{K_{Ra}}[ID_X \parallel E_{K_{Uy}}[E_{K_{Rx}}[M]] \parallel T]$

Notation:

X = sender
Y = recipient
A = Arbiter

M = message
T = timestamp

Digital Certificate

A digital certificate is an electronic document used to prove ownership of a public key. The certificate includes the public key, information about its owner's identity and associated permissions. Certificates provide the foundation for a PKI (Public Key Infrastructure). Certificates are electronic representations of users, computers, network devices, or services, issued by a CA (Certificate Authority), that are associated with a public and private key pair. The current standard being used for digital certificates is X.509 defined in RFC 5280. Some main parameters that exist in a certificate are:

- Certificate serial number
- Name of the certificate issuer
- Certificate issuer's signature algorithm identifier
- Validity period
- Public key

UNIT-III

User Authentication Mechanisms

Introduction

Authentication

Verifying the identity of another entity

- Computer authenticating to another computer
- Person authenticating to local computer
- Person authenticating to remote computer

Authentication is the process of verifying the identity of a user who is trying to gain access to a system.

Authorization:

The method that is used to determine the resources that are accessible to an authenticated user is called authorization.

Authentication Requirements:

In the context of communications across a network, the following attacks can be identified:

1. **Disclosure:** Release of message contents to any person or process not possessing the appropriate cryptographic key.
2. **Traffic analysis:** Discovery of the pattern of traffic between parties. In a connection oriented application, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined.
3. **Masquerade:** Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or nonreceipt by someone other than the message recipient.
4. **Content Modification:** Changes to the contents of a message, including insertion, deletion, transposition, or modification.

5. **Sequence modification:** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.

6. **Timing modification:** Delay or replay of messages. In a connection-orientated application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed.

7. **Repudiation:** Denial of receipt of message by destination or denial of transmission of message by source.

Message authentication is a procedure to verify that received messages come from the alleged source and have not been altered. Message authentication may also verify sequencing and timeliness. A **digital signature** is an authentication technique that also includes measures to counter repudiation by either source or destination.

Authentication Functions:

1. **Message Encryption:** The ciphertext of the entire message serves as its authenticator.

2. **Message Authentication Code (MAC):** A public function of the message and a secret key that produces a fixed length value that serves as the authenticator.

3. **Hash Functions:** A public function that maps a message of any length into a fixed length hash value, which serves as the authenticator.

Message Encryption:

□ Message encryption by itself can provide a measure of authentication. □
There is a difference between for symmetric and public-key encryption schemes.

1. **Symmetric Encryption:** □ A message **M** transmitted from source **A** to destination **B** is encrypted using a secret key **K** shared by both **A** and **B**. If no other party knows the key, then confidentiality is provided. No other party can recover the plain text of the message. Hence, authentication is provided. Hence, symmetric encryption provides authentication as well as confidentiality. This is explained in the following diagram.

□

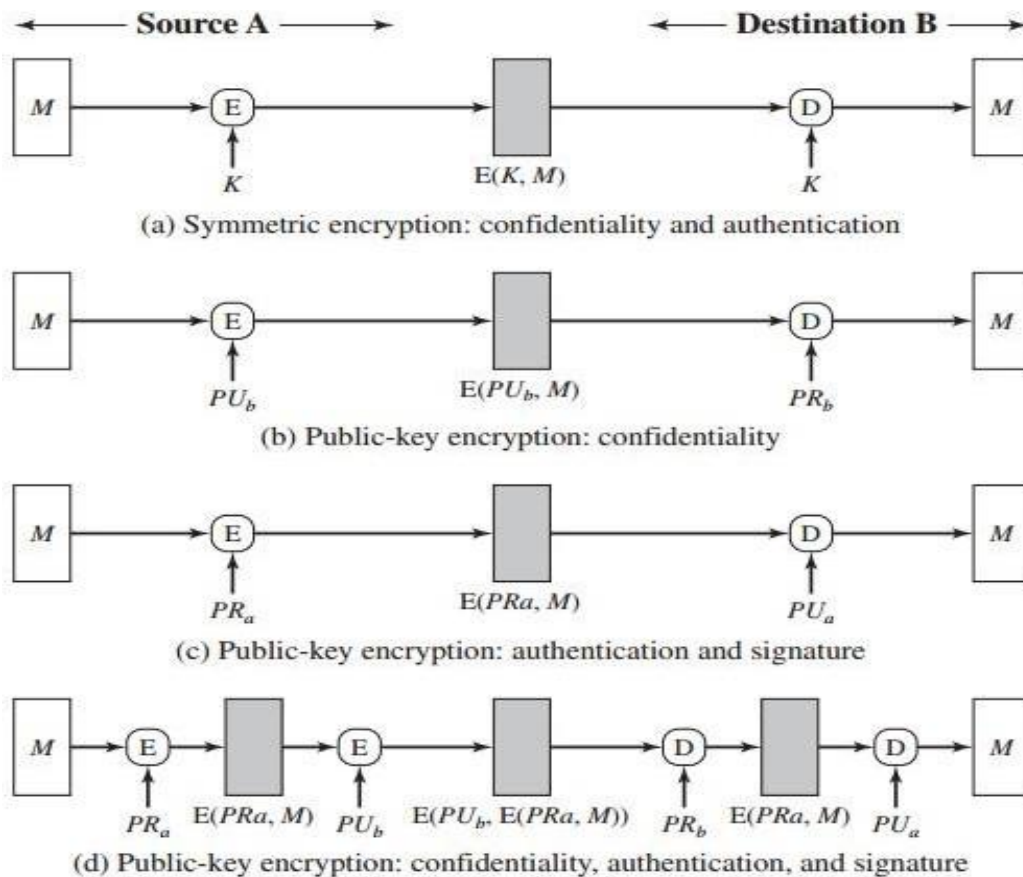


Figure 12.1 Basic Uses of Message Encryption

2. Public-Key Encryption:

This provides no confidence of sender because anyone potentially known public-key. However, if sender signs message using their private-key, then encrypts with recipients public key. Hence it provides both secrecy and authentication.

Message Authentication Code:

An alternative authentication technique involves the use of a secret key to generate a small fixed-size block of data, known as a cryptographic checksum or MAC that is appended to the message. This technique assumes that two communicating parties, say A and B, share a common secret key K.

□ $MAC = C(K, M)$

where M = input message

C = MAC function

K = shared secret key

MAC = message authentication code

This is explained in the following diagram.

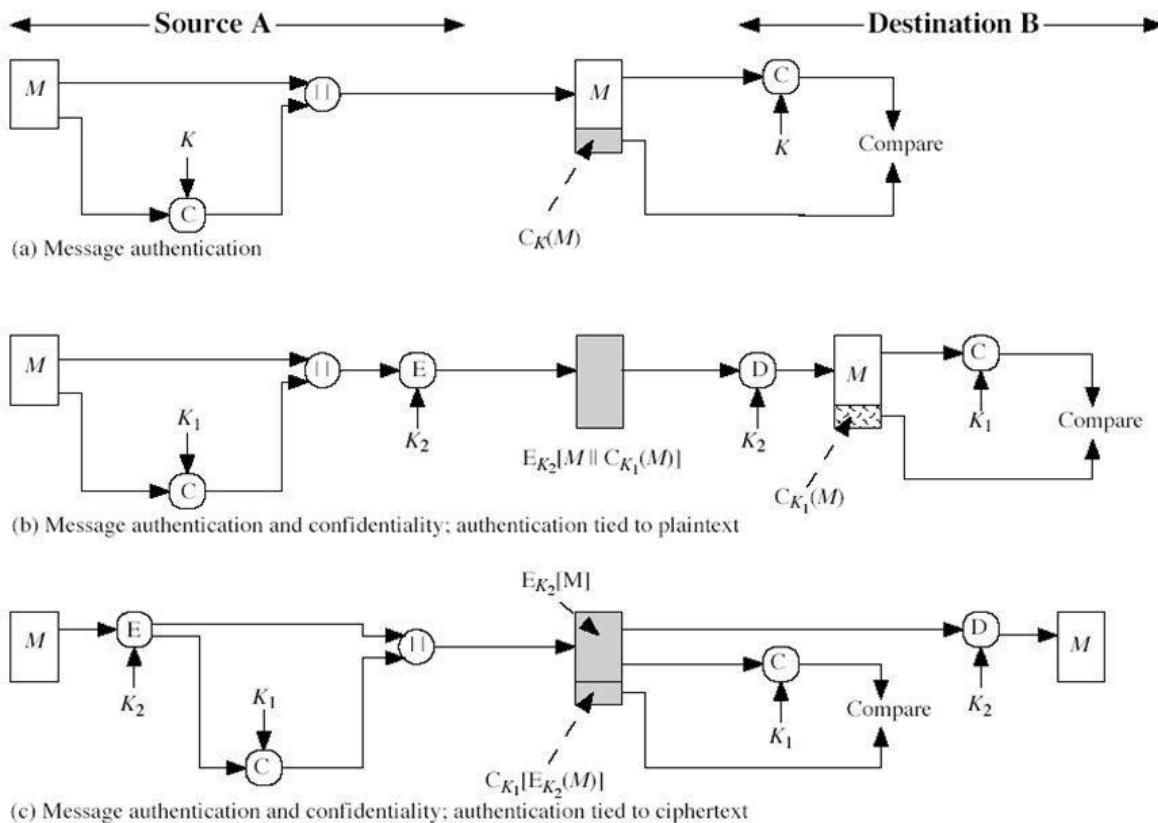
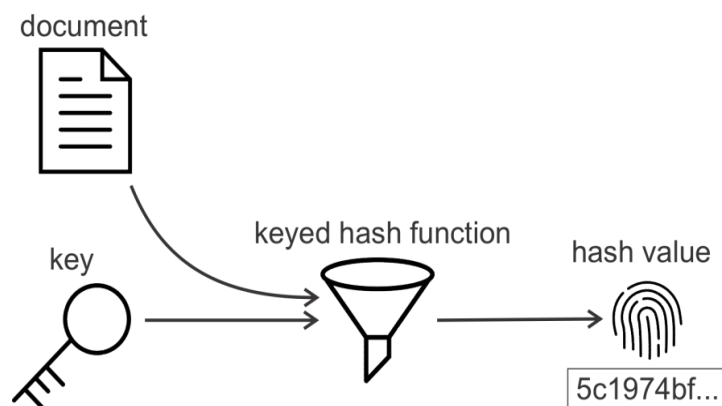
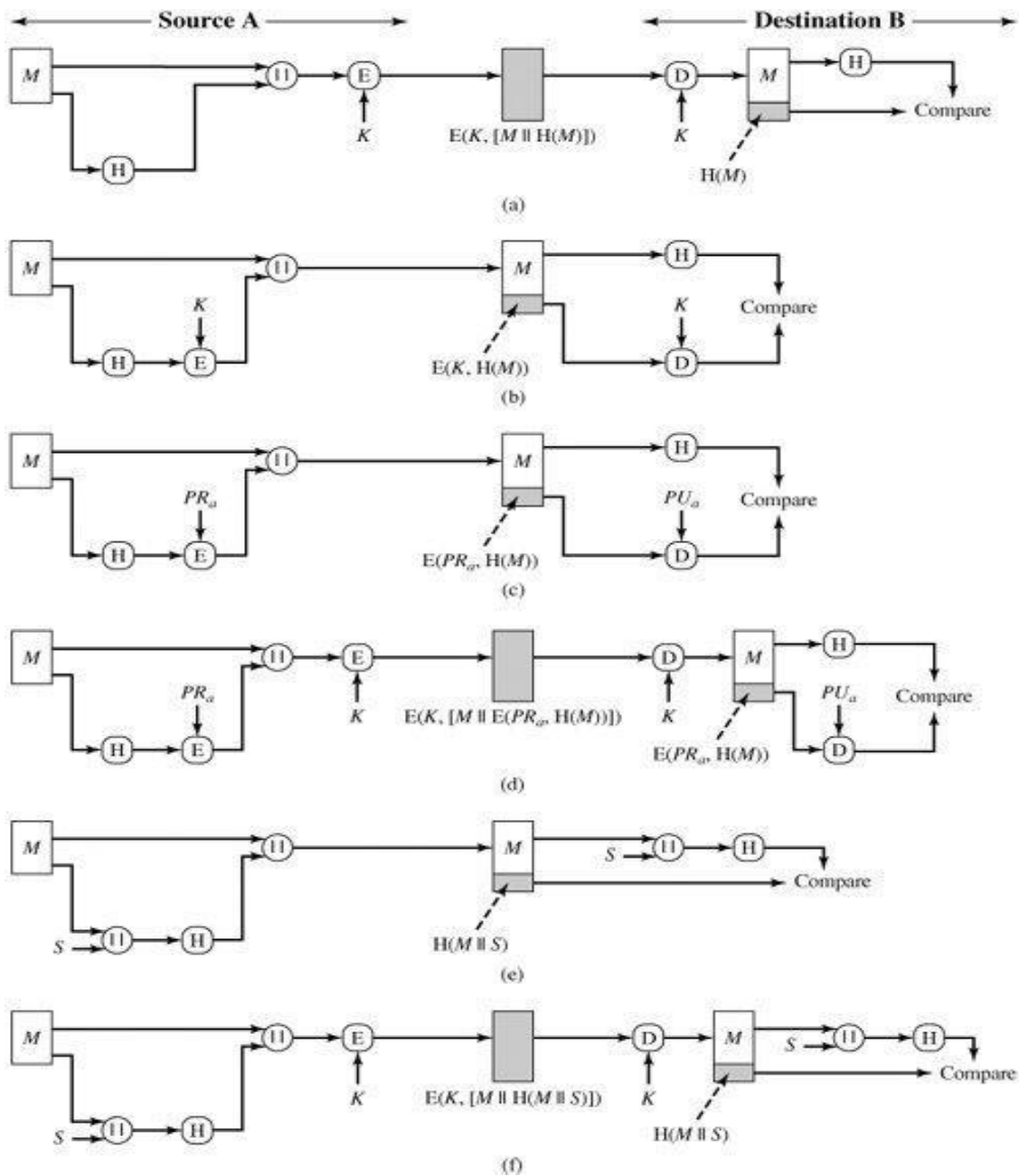


Figure 11.4 Basic Uses of Message Authentication Code (MAC)

Hash Function:

□ The hash function accepts a variable-size message M as input and produces a fixed size output, referred to as a hash code $H(M)$. A hash code does not use a key but is a function only of the input message. The hash code is also referred to as a message digest or hash value. The hash code is a function of all the bits of the message and provides an error-detection capability. The basic uses of hash functions are explained in the following table and diagram.

<p>a) $A \rightarrow B: E(K, [M H(M)])$ □</p> <ul style="list-style-type: none"> * Provides confidentiality <ul style="list-style-type: none"> -- Only A and B share K □ * Provides authentication <ul style="list-style-type: none"> -- H(M) is cryptographically protected 	<p>d) $A \rightarrow B: E(K, [M E(PRa, H(M))])$ □</p> <ul style="list-style-type: none"> * Provides authentication and digital signature * Provides confidentiality <ul style="list-style-type: none"> -- Only A and B share K
<p>b) $A \rightarrow B: M E(K, H(M))$</p> <ul style="list-style-type: none"> □ * Provides authentication <ul style="list-style-type: none"> -- H(M) is cryptographically protected 	<p>e) $A \rightarrow B: M H(M S)$ □</p> <ul style="list-style-type: none"> * Provides authentication <ul style="list-style-type: none"> -- Only A and B share S
<p>c) $A \rightarrow B: M E(PRa, H(M))$ □</p> <ul style="list-style-type: none"> * Provides authentication and digital signature <ul style="list-style-type: none"> -- H(M) is cryptographically protected -- Only A could create $E(PRa, H(M))$ 	<p>f) $A \rightarrow B: E(K, [M H(M S)])$ □</p> <ul style="list-style-type: none"> * Provides authentication <ul style="list-style-type: none"> -- Only A and B share S * Provides confidentiality <ul style="list-style-type: none"> o Only A and B share K



Passwords Authentication

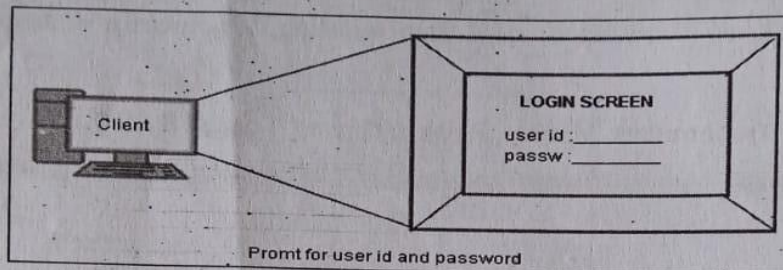
Passwords:

A password is a string of alphabets, numbers and special characters, which is supposed to be known only to the entity (usually a person) that is being authenticated.

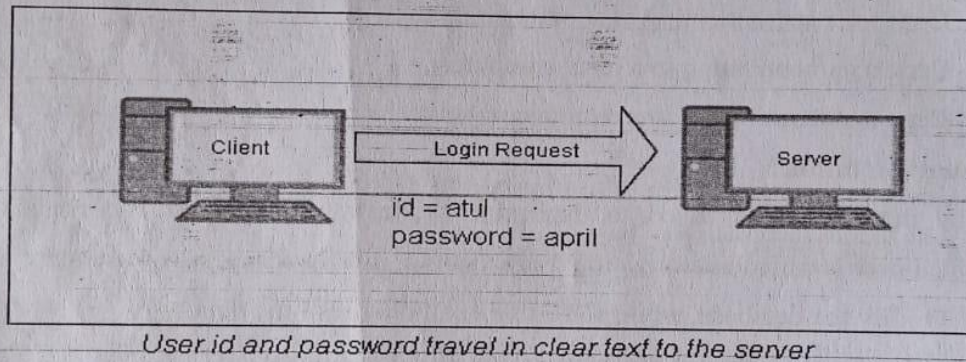
Clear Text Passwords:

This is the simplest password-based authentication mechanism. Usually, every user in the system is assigned a user id and an initial password. The user changes the password periodically for security reasons. The password is stored in clear text in the user database against the user id on the server. The authentication mechanism works as follows:

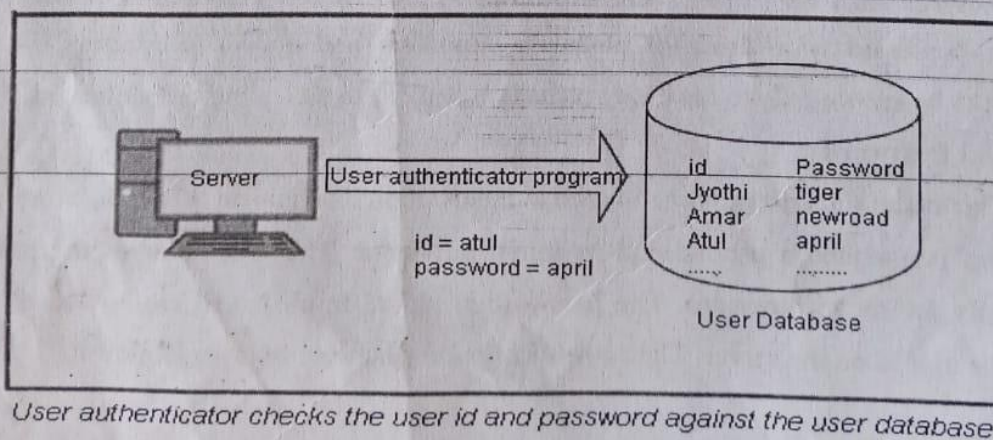
Step 1: Prompt for user id and password: During authentication the application sense a screen to the user, prompting for the user id and password. This is shown in the following figure:



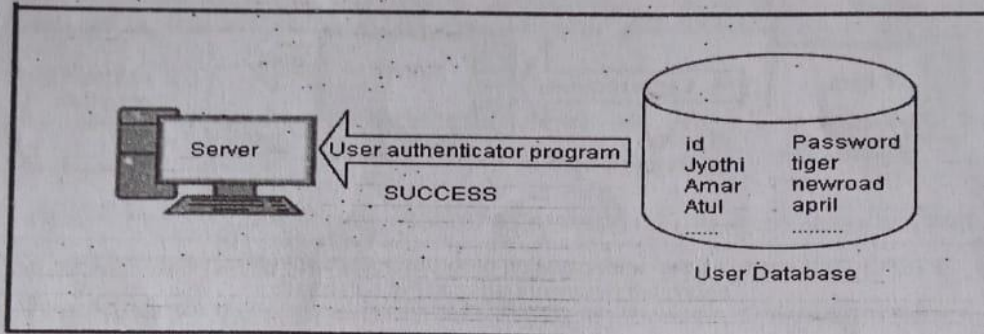
Step 2: User enters user id and password: The user enters her id and password and presses the OK equivalent button. This causes the user id and password to travel in clear text to the server. This is shown in the following figure:



Step 3: User id and password validation: The server consults the user database to see if this particular user id and password combination exists there. Usually, this is the job of a user authenticator program is shown in the following figure. This is a program that takes user id and password, checks it against user database, and returns the result of the authentication (success or failure). This is one of the types of checking.



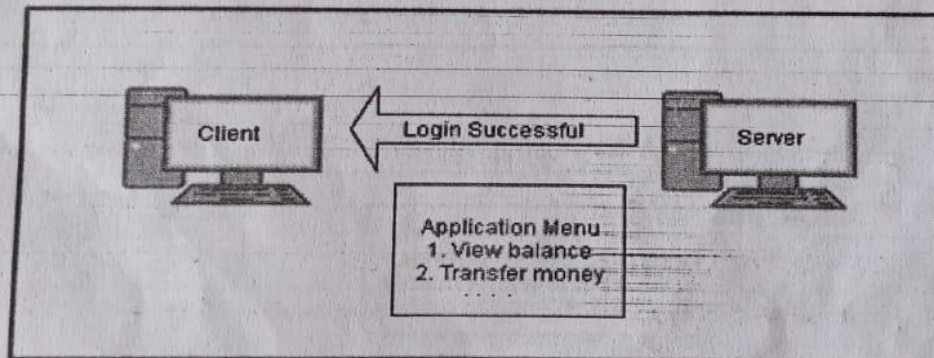
Step 4: Authentication Result: Depending on the success or the failure of the validation of the user id on the password, the user authenticator program returns an appropriate result back to the server which is shown in the following diagram.



User authenticator program returns a success or failure message to the server

Here, we assume that the user was authenticated successfully.

Step 5: Inform user accordingly: Depending on the outcome (success/failure), the server sends back an appropriate screen to the user. If the user authentication was successful, the server typically sends a menu of options of the user, which lists the actions the user is allowed to perform. If the result of the user authentication was a failure, the server sends an error screen to the user which is shown in the following figure.



Server returns a success or failure result back to the user

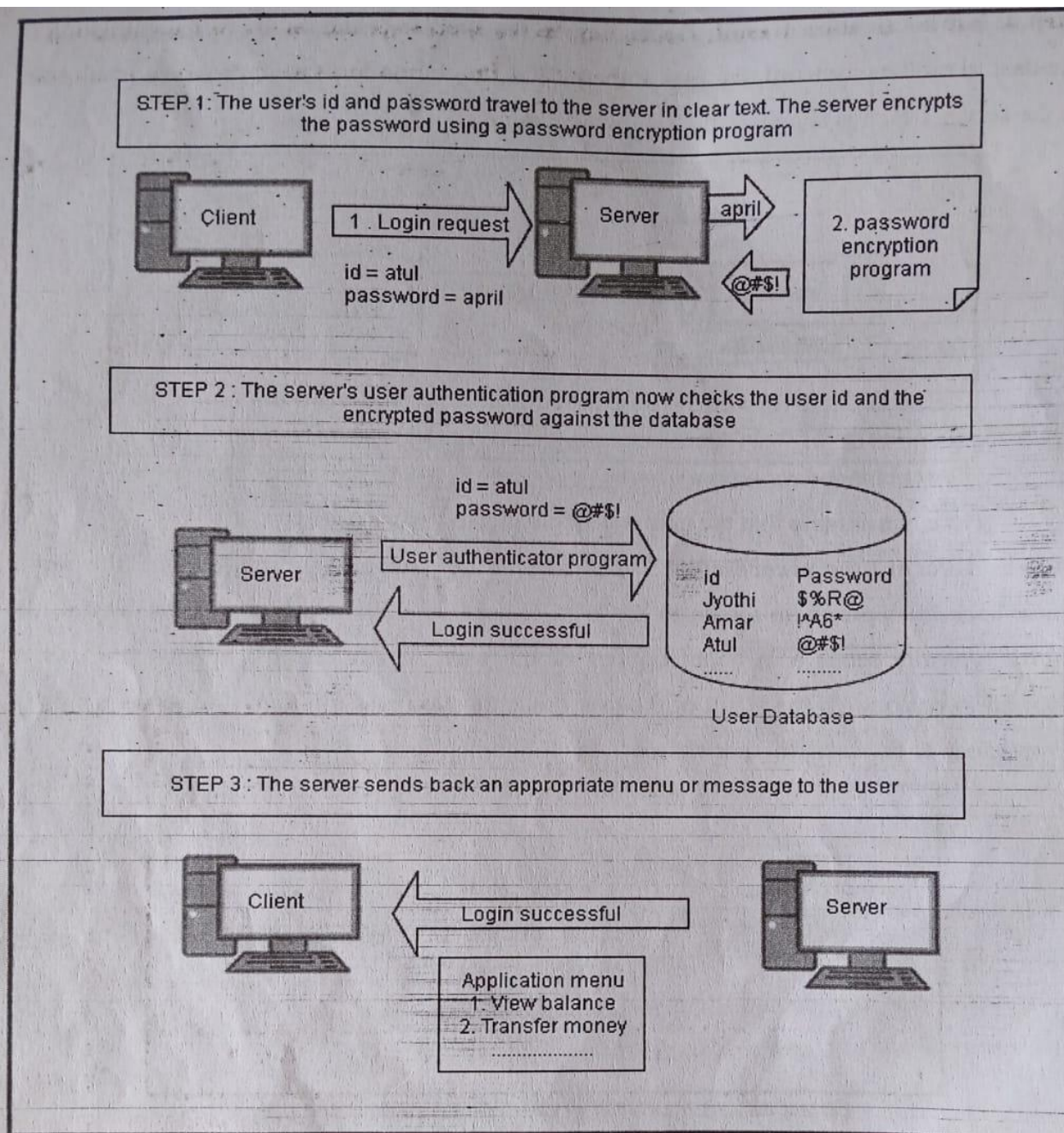
Here we assume that the user was authenticated successfully.

Problems with the scheme:

There are two major problems in this approach.

Problem 1: The database consists user ids and passwords in clear text

If an attacker succeeds to access the database, he can get whole list of user ids and corresponding passwords and also he can understand the passwords because they are in the clear text. The solution to this problem is the passwords should be encrypted. This is shown in the following:



Encryption password before they are stored and verified

Problem 2: Password travels in clear text from the user's computer to the server

Though the server stored passwords in the encryption format, the attacker attacks while the password is traveling from user's computer to the server for authentication and he can understand the password easily because the password is in the clear text format.

Something derived from Passwords:

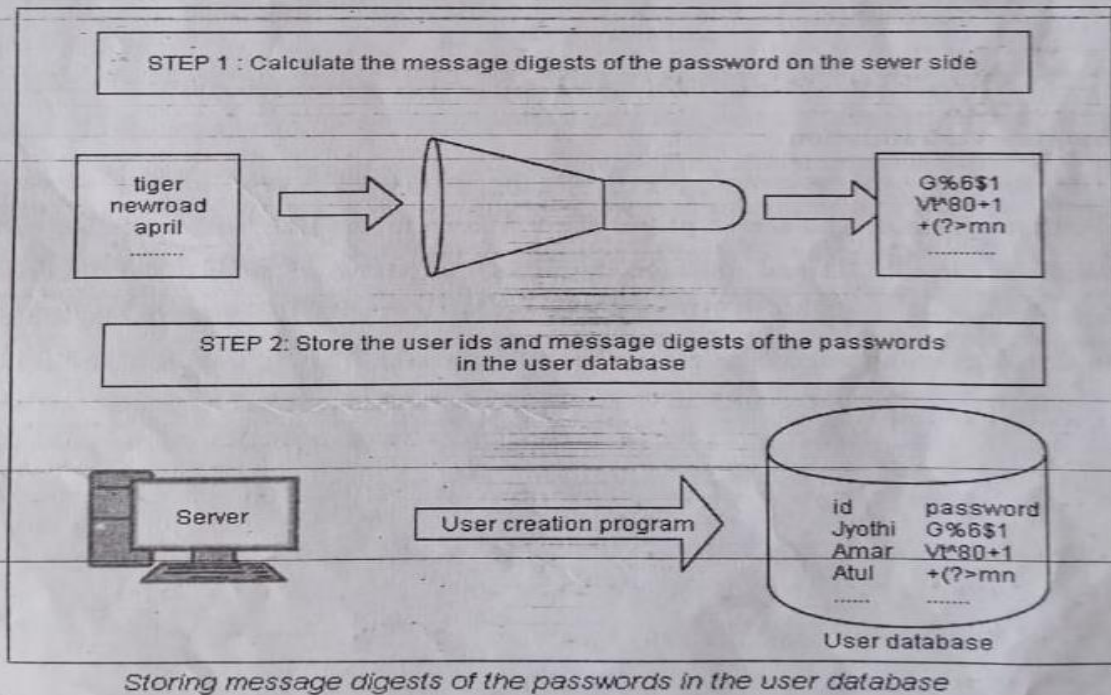
Instead of storing password as it is, the password is passed as input to the encryption algorithm and the output password is in the encrypted format which is stored in the server. When the user wants to be authenticated, the user enters password and the user's computer

Message Digests of Passwords:

This is a simple technology to avoid the storage and transmission of clear text passwords. The following steps explain the working nature of this technique.

Step 1: Storing message digests as derived passwords in the user database

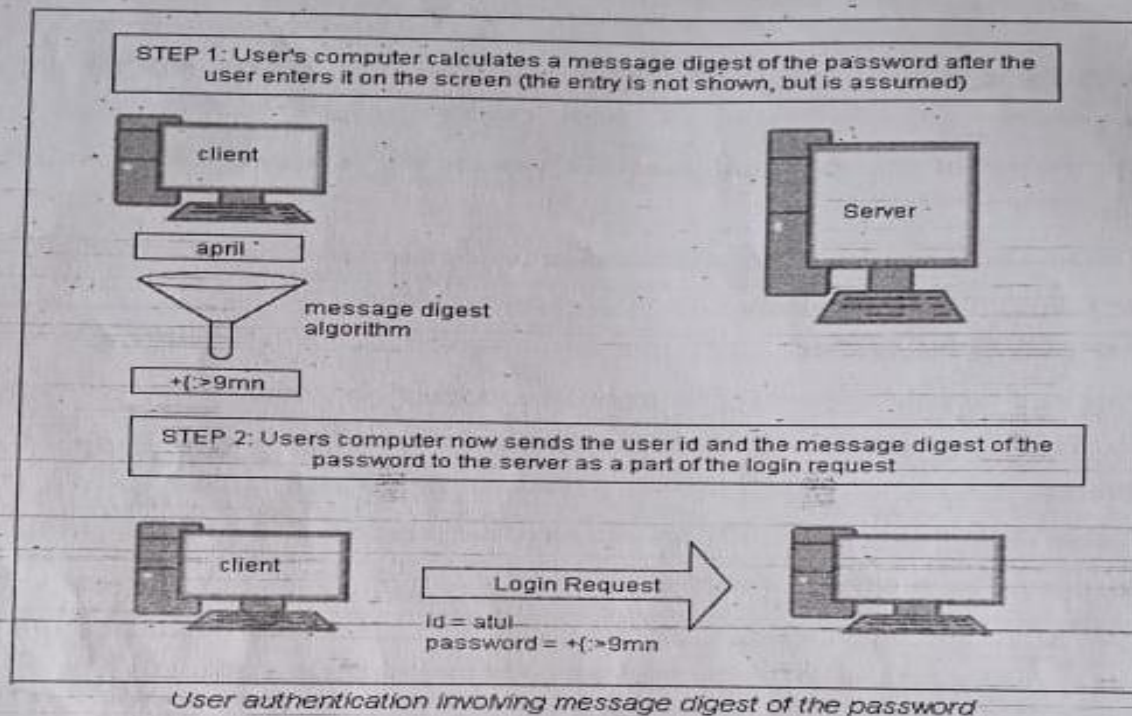
Rather than storing passwords, we can store the message digests of the passwords in the database is shown in the following figure:



Step 2: User authentication

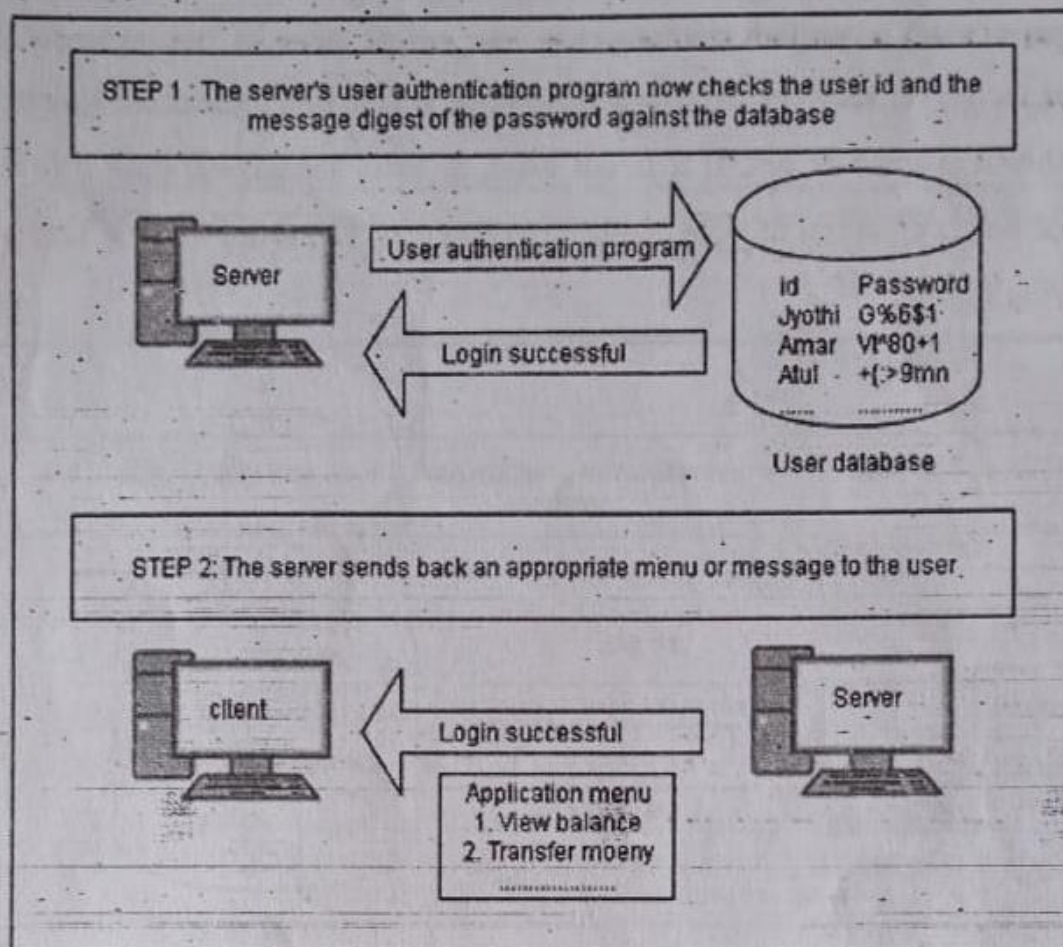
When a user needs to be authenticated, the user enters the id and password as usual. Now the user's computer computes the message digest of the password and sends user id and

message digest of the password to the server for authentication. This is shown in the following figure:



Step3: Server-side validation

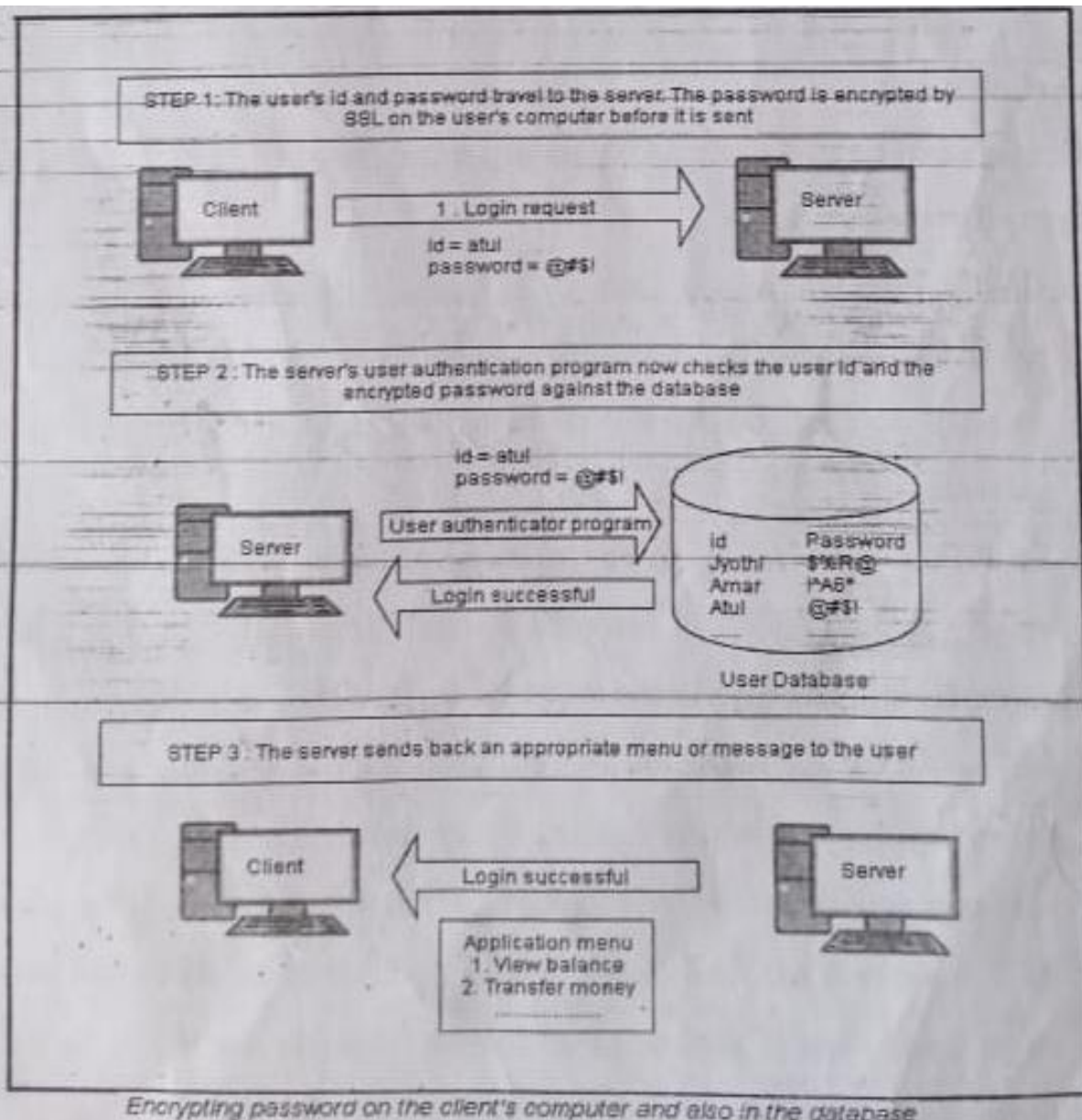
The user id and message digest of the password travel to the server over communication channel. The server passes these values to the user authentication program which validates the user id and message digest of the password against the database and returns an appropriate response to the back to the server. The server used the result of this operation to return an appropriate message back to the user. This is shown in the following figure:



User authentication program validates the user id and the message digest of the password

Password Encryption:

The clear text password is encrypted on the user's computer and then sends it to the server for authentication. In the case of Internet applications, the client is a Web browser, which is not having encryption capability. Consequently, we must resort to technologies such as Secure Socket Layer (SSL). The SSL creates a secure connection between client and server. The SSL would perform the required encryption operations. This is shown in the following figure:



Encrypting password on the client's computer and also in the database

The Problems with Passwords:

The passwords are not truly random:

1. With 52 upper- and lower-case letters, 10 digits and 32 punctuation symbols, there are $94^8 = 6$ quadrillion possible 8-character passwords
2. Humans like to use dictionary words, human and pet names 1 million common passwords

The problems with passwords are:

1. **External Disclosure:** Password becomes known to an unauthorized person by a means outside normal network or system operation includes storing passwords in unprotected files or posting it in an unsecured place.
2. **Password Guessing:** Results from very short passwords, not changing passwords often, allowing passwords which are common words or proper names, and using default passwords.
3. **Live Eavesdropping:** Results from allowing passwords to transmit in an unprotected manner.
4. **Verifier Comprise:** Occurs when the verifier's password file is designed via an internal attack.
5. **Replay:** Recording and later replaying of a legitimate authentication exchange.

Types of Attacks:

There are two types of attacks. They are a) On-line Password Attack b) Off-line or Dictionary Password Attack.

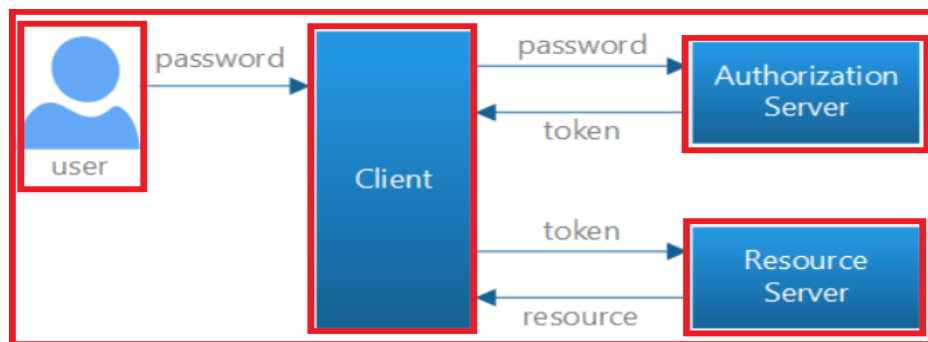
1. **On-line password Attack:** One way of guessing passwords is simply to type passwords at the system that is going to verify the password is known as *on-line password attacking*. But the system can make it impossible to guess too many passwords.
Ex: ATM card eat our card, if we type three incorrect passwords. If the attacker is trying with several different passwords to break the original password, then the system dispatch a message, human to investigate.
2. **Off-line Attack:** An intruder can get a quantity X that is derived from a password in known way. Now the intruder uses an arbitrary amount of compute power to guess passwords and convert them into known way. In this manner the intruder can produce the X. A source of good passwords kept in a little dictionary and the intruder can use to get the X value. Therefore, an off-line password guessing is also known as *Dictionary Attack*.

Authentication Tokens

An authentication token is an extremely useful alternative to a password. An authentication token is a small device that generates a new random value every time it is used. This random value becomes the basis for authentication. The small devices are typically of the size of small key chains, calculators or credit cards.

Components of Authentication Token

- Processor
- LCD for displaying output (Liquid Crystal Display)
- Battery
- A small keypad for entering information
- A real-time clock



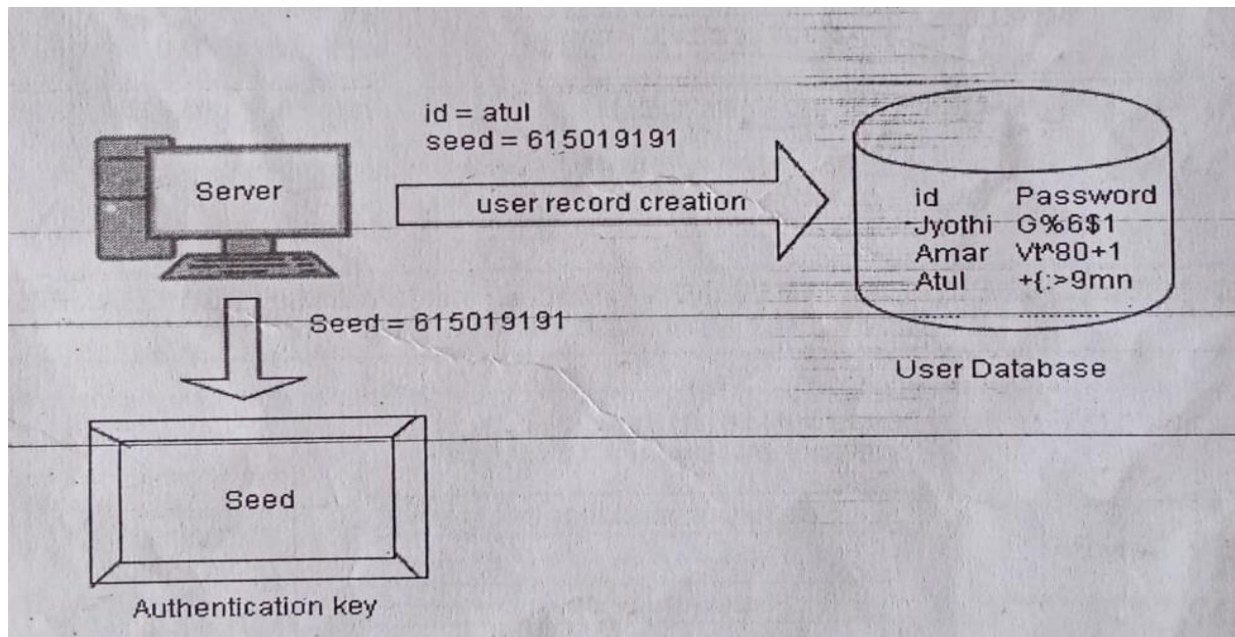
1. The user enters their username and password.
2. The server verifies that the login information is correct and generates a secure, signed token for that user at that particular time.
3. The token is sent back to the user's browser and stored there.
4. When the user needs to access something new on the server, the system decodes and verifies the attached token. A match allows the user to proceed.
5. Once the user logs out of the server, the token is destroyed.

Uses of TOKENS

1. Tokens are stateless.
2. Tokens can be generated from anywhere.

3. Fine-grained access control.

Every authentication token is pre-programmed with a unique number called **random seed or seed**. This seed ensures that the output generated by the authentication token (the device) is unique. An authentication token is an example of 2-factor authentication because the token itself is protected with some PIN.



1. Creation of Token

When an authentication token is created, the corresponding random seed is generated for the token by the authentication server. This seed is automatically used by the authentication token due to which the value of the seed is not known by the user. This seed is pre-programmed inside the token, as well as its entry is made against that user's record in the user database.

2. Use of Token

An authentication token automatically generates pseudo-random numbers, called one-time passwords or one-time passcode (these codes/passwords can be used only once). Once they are used, they cannot be reused. This one time password is basically a 4 digit PIN. Below are some important points to use this one time password.

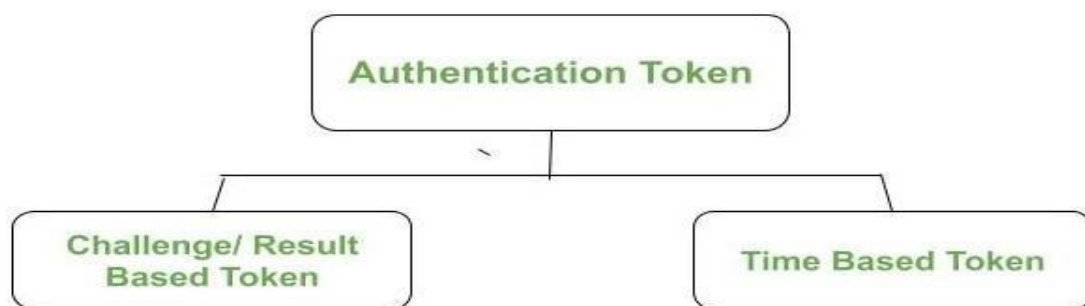
1. The user being authenticated will enter his/her id and one-time password which goes to the server.
2. The server receives the seed corresponding to the user id from the user database, using a *seed-retrieval program*.

3. The server gives the seed and one-time passcode to a *Password Validation Program*.
4. This program checks if the one-time password and seed are related to each other.

3. Server responds

The server finally responds back with a suitable message based on the output (success/failure) of the previous step.

Types of Authentication Token:



Challenge/Response Token

1. User sends a login request by providing only his user id and not the one-time password.
2. Server checks whether the user id is valid. If it is not valid, it responds with an error message otherwise if it is valid, then the server creates a random challenge. Then sends the random challenge to the user.
3. User receives the random challenge. Open the authentication token using the PIN and keys in the random challenge using the small keypad.
4. The seed of the token encrypts the random challenge which is then entered by the user in the password section of the login request.
5. Server verifies the encrypted random challenge received by the user which can be done in two ways-
 - a) The server can decrypt the encrypted random challenge received from the user with the seed value for the user, which is available to the server via the user database. If this decryption matches the original random challenge available on the server, the authentication is successful.

b) The server can encrypt its own version of the random challenge, which was sent earlier to the user, with the seed for the user. If this encryption matches with the encrypted random challenge received from the user, the authentication is successful.

Time-Based Token

In a time-based token, the server need not send any random challenge to the user. The token need not have a keypad for entry. In fact, it uses time in place of a random challenge. The tokens automatically generate a password every 60 seconds and display the latest password on the LCD output for the user.

For generating the password, the time-based tokens use seed and current system time.

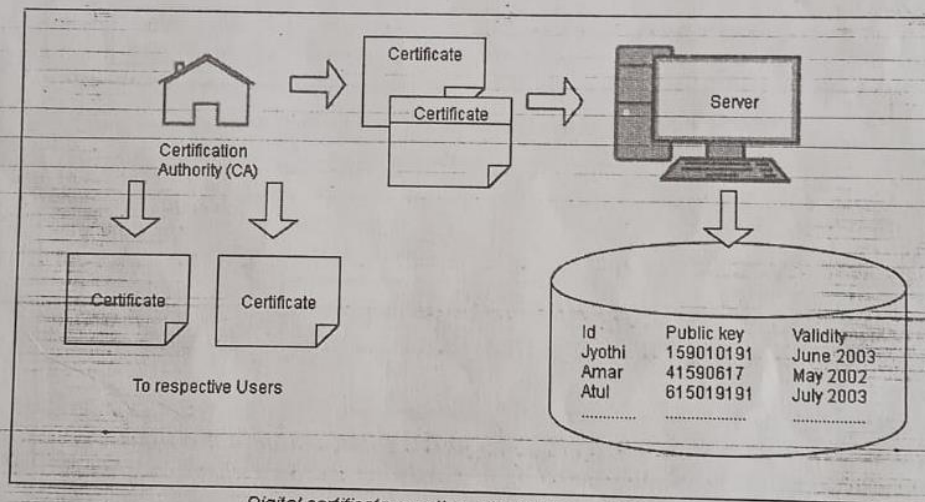
- When a user wants to log in he/she enters the password displayed on the LCD of the token and uses it to login along with their user id.
- The server receives the password and performs an independent cryptographic function on the user's seed value and the current system time to generate its version of the password. If the two values match, it considers the user as a valid one.
- Finally, the server sends an appropriate message back to the user based on the result of the previous step.

Certificate based authentication

Certificate-Based Authentication:

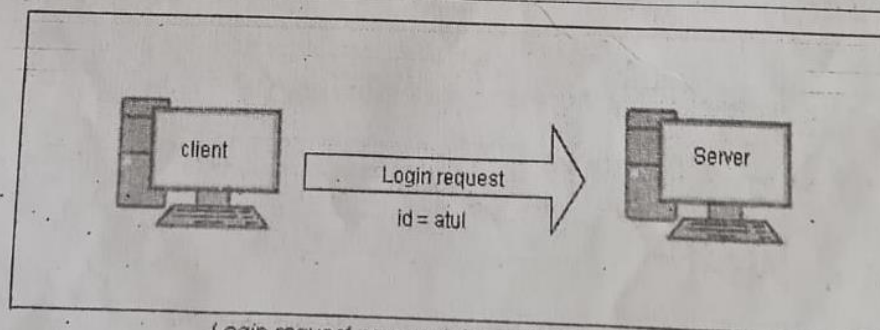
This is a stronger authentication mechanism as compared to a password based authentication because here the user is expected to have certificate and not know password. At the time of login, the user requested to send his/her certificate to the server over the network. The server verifies the user's the validity of the certificate. The following steps explain this mechanism.

Step 1: Creation, storage and distribution of digital certificates: The digital certificates are created and issued to the users and the copy of the certificates is stored in the server in binary format. This is shown in the following figure:



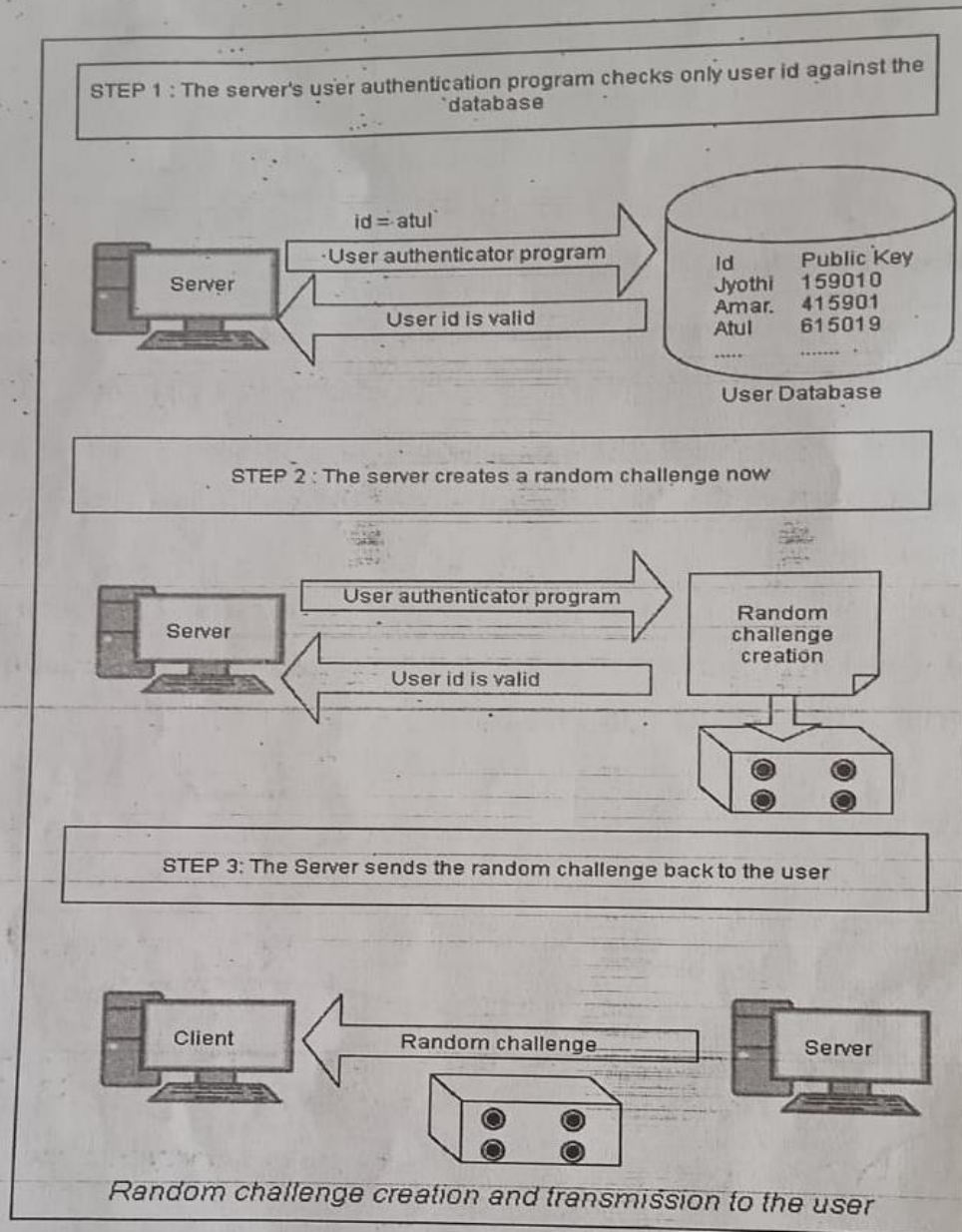
Digital certificate creation, distribution and storage

Step 2: Login request: The user send only user id to the server and is shown in the following figure:

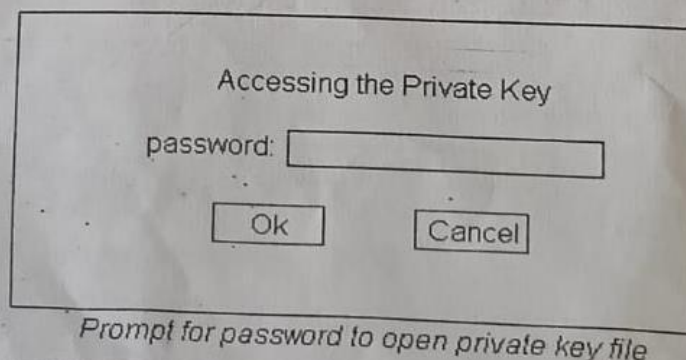


Login request now contains only the user id

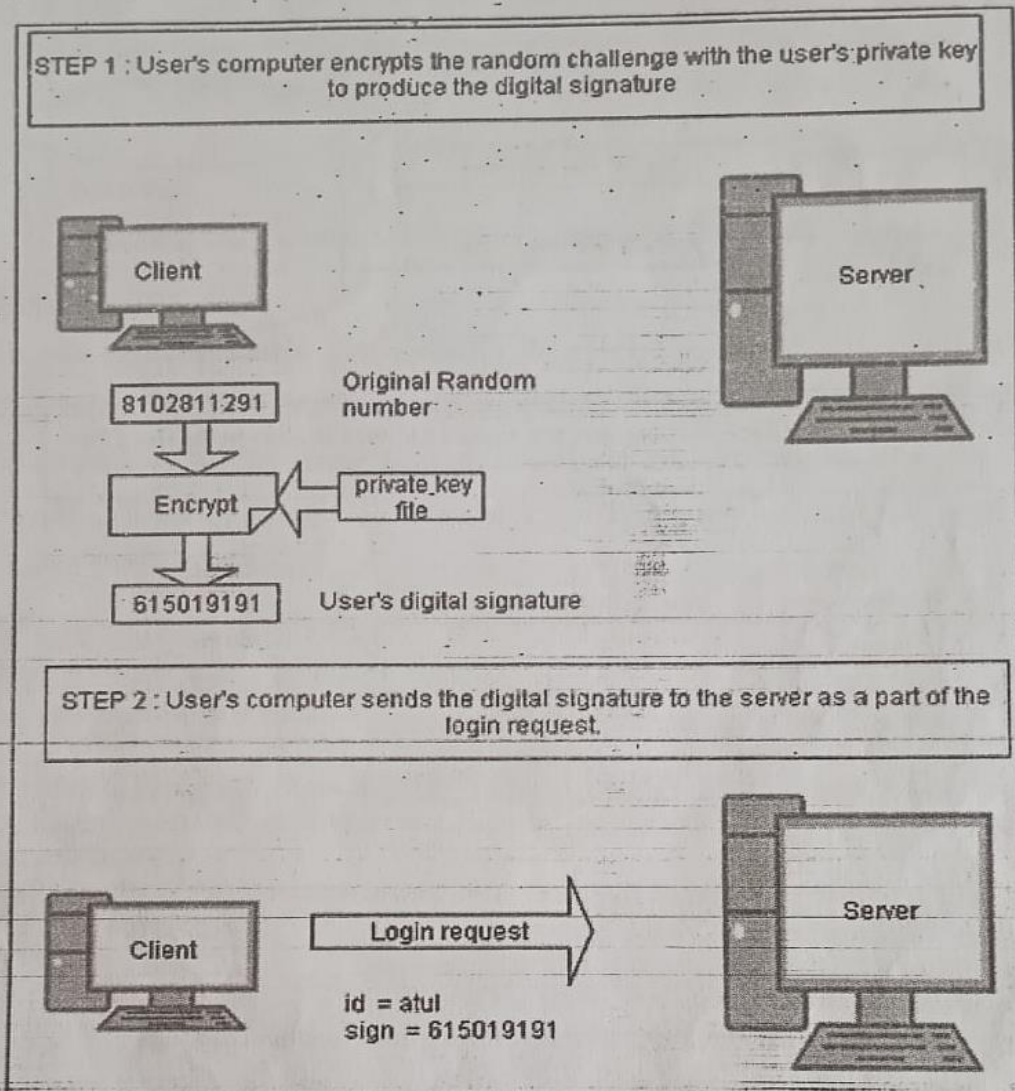
Step 3: Server creates a random challenge: The server validates the user id and if it is correct, it sends a random number as challenge to the user and is explained in the following figure:



Step 4: User signs the user challenge: The user signs the random number using his/her private key. The user needs to access private key which is stored on disk of his/her computer. This is shown in the following figure:



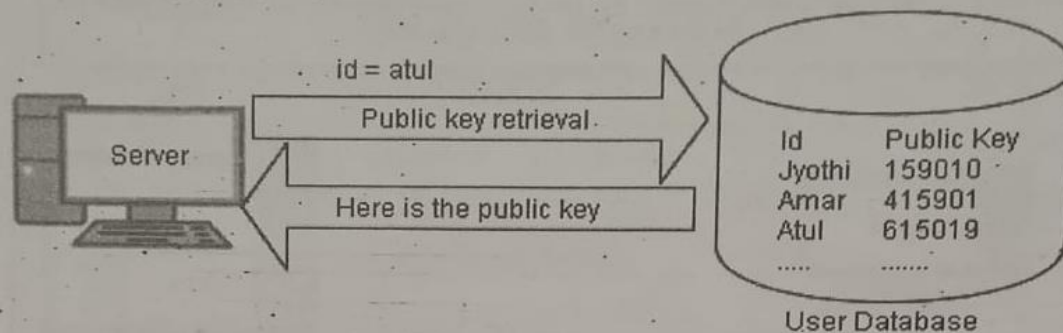
After entering correct password, the user's private key file is opened and the user uses private key to encrypt the random number. This is shown in the following diagram:



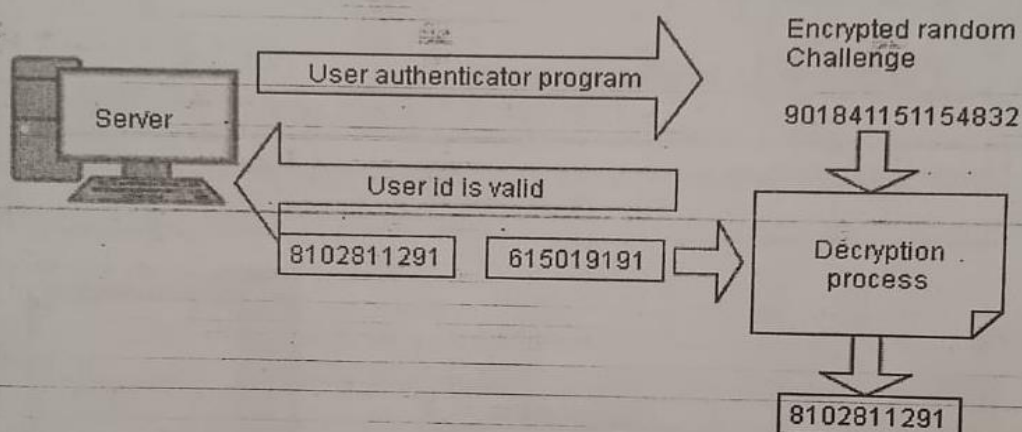
User's computer signs the random challenge and sends it back to the server

The server uses public key of the user to decrypt the encrypted random number received from the user. This is shown in the following figure:

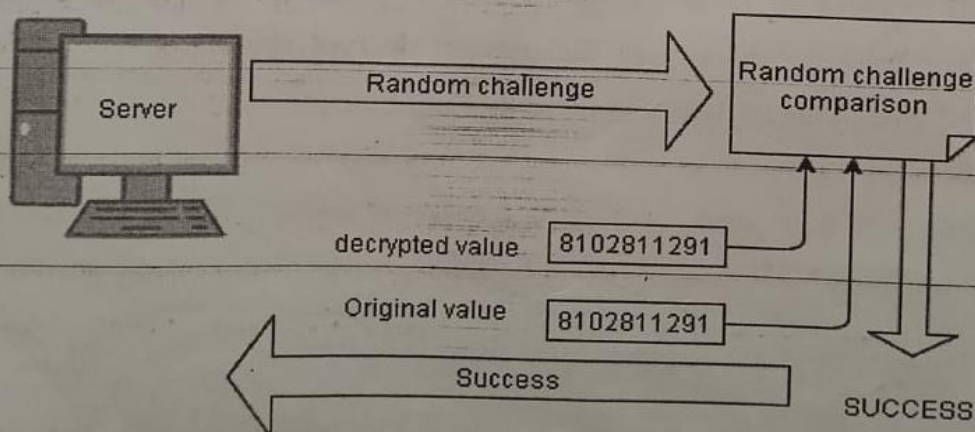
STEP 1: The server's user authentication program obtain the public key for the user from the user database



STEP 2: The server decrypts (design) the signed random challenge received from the user, using the user's public key

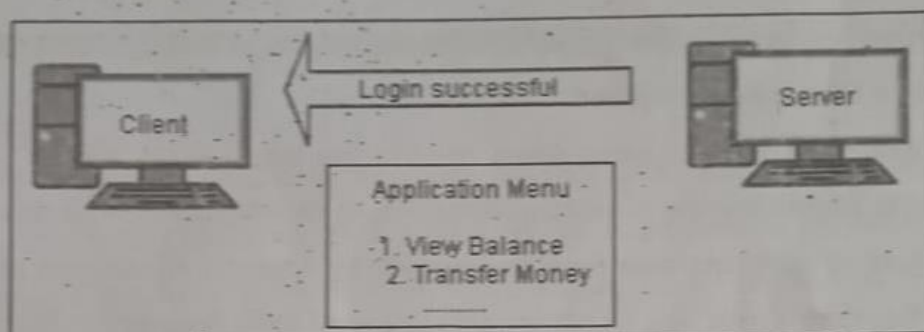


STEP 3: The Server compares the two encrypted random challenges



Server compares the two random challenges

Step 5: Server returns an appropriate message back to the user: Finally the server sends an appropriate message (success/failure) to the user. This is shown in the following figure:



Server sends an appropriate message back to the user

Use of Smart Cards:

The use of smart cards can actually be related to certificate-based authentication because they allow the generation of private-public key pairs within the cards. They also support the storage of digital certificates within the card. The private key always is stored within the card in a secure, tampered free fashion. The public key and the certificate can be exported outside. The smart cards are capable of performing cryptographic functions such as encryption & decryption, message digest creation and signing within the card. The following table explains the problems and their solutions related to smart card technology:

<i>Problem/Issue</i>	<i>Emerging solution</i>
Smart card readers are not yet a part of a desktop computer, unlike a hard disk drive or a floppy disk drive	The new versions of computers and mobile devices are expected to come with smart card readers <i>out of the box</i> .
Non-availability of smart card reader driver software	Microsoft has made the PC/SC smart card framework an integral part of the Windows 2000 operating system. Most smart card reader manufacturers ship the PC/SC compliant reader drivers, making the process of adding a reader hardware to the computer a plug-and-play operation.
Non availability of smart card aware cryptographic services software Cost of smart cards and card readers is high	Smart-card aware software such as Microsoft Crypto API (MS-CAPI) comes free with Internet Explorer. This is reducing now. Smart cards are available for about \$5, and the card readers for about \$20.

Biometric authentication:

These mechanisms are receiving a lot of public attention. A biometric device is perhaps the ultimate attempt in trying to prove who you are. Biometrics allows a person to be identified and authenticated based on a set of recognizable and verifiable data, which are unique and specific to them.

Biometric authentication is the process of comparing data for the person's characteristics to that person's biometric "template" in order to determine resemblance. The reference model is first store in a database or a secure portable element like a smart card. The

data stored is then compared to the person's biometric data to be authenticated. Here it is the person's identity which is being verified.

In this mode, the question being asked is: "Are you indeed Mr or Mrs X?"

Biometric identification consists of determining the identity of a person. The aim is to capture an item of biometric data from this person, for example by taking a photo of their face, by recording their voice, or by capturing an image of their fingerprint. This data is then compared to the biometric data of several other persons kept in a database.

In this mode, the question being asked is a simple one: "Who are you?"

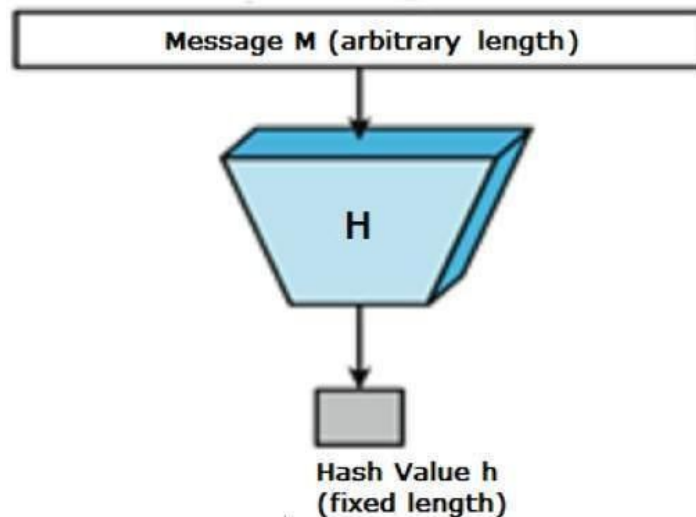
Types of biometric authentication technologies:

1. **Retina scans** produce an image of the blood vessel pattern in the light-sensitive surface lining the individual's inner eye.
2. **Iris recognition** is used to identify individuals based on unique patterns within the ring-shaped region surrounding the pupil of the eye.
3. **Finger scanning**, the digital version of the ink-and-paper fingerprinting process, and works with details in the pattern of raised areas and branches in a human finger image.
4. **Finger vein ID** is based on the unique vascular pattern in an individual's finger.
5. **Facial recognition** systems work with numeric codes called faceprints, which identify 80 nodal points on a human face.
6. **Voice identification** systems rely on characteristics created by the shape of the speaker's mouth and throat, rather than more variable conditions.

Hash Functions

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

1. Fixed Length Output (Hash Value)

- Hash function converts data of arbitrary length to a fixed length.
- 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.

2. 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.

Popular Hash Functions

- Message Digest (MD)
- Secure Hash Function (SHA)
- RIPEMD (RACE Integrity Primitives Evaluation Message Digest)

Applications of Hash Functions

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

- Password Storage
- Data Integrity Check

SHA-1 Algorithm

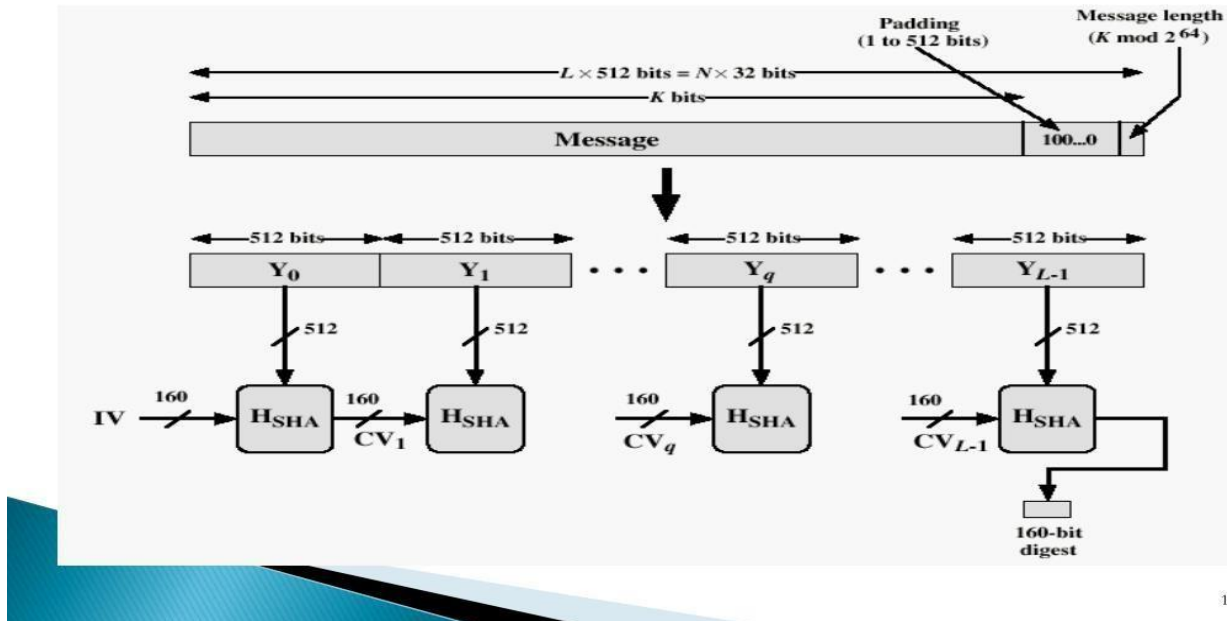
The Secure Hash Algorithm 1 (SHA-1) is a cryptographic computer security algorithm. It was created by the US National Security Agency in 1995, after the SHA-0 algorithm in 1993, and it is part of the Digital Signature Algorithm or the Digital Signature Standard (DSS). SHA is based on the MD4 algorithm and its design closely models MD4.

SHA-1 Logic:

The algorithm takes as input a message with a maximum length of less than 2^{64} bits and produces as output a 160 bit message digest. The input is processed in 512 bit blocks.

The overall processing of a message follows the structure shown in the following figure:

Message Digest Generation Using SHA-1



The processing consists of the following steps:

Step 1: Append padding bits: The message is padded so that its length is congruent to 448 modulo 512. Padding is always added, even if the message is already of the desired length.

Step 2: Append length: A block of 64 bits is appended to the message. This block is treated as an unsigned 64-bit integer and contains the length of the original message.

Step 3: Initialize MD buffer: A 160 bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as five 32-bit registers (A,B,C,D,E). These registers are initialized to the following 32-bit integers (hexadecimal values):

$$A = 67452301$$

$$B = \text{EFCDA}89$$

$$C = 98\text{BADCFE}$$

$$D = 10325476$$

$$E = \text{C3D2E1F0}$$

Step 4: Process message in 512-bit (16 word) blocks: The heart of the algorithm is a module that consists of four rounds of processing of 20 steps each. The logic is illustrated in the following figure. The four rounds have a similar structure, but each uses a different primitive logical function, which refer to as f_1 , f_2 , f_3 and f_4 .

Each round takes as input the current 512 bit block being processed (Y_q) and the 160-bit buffer value ABCDE and updates the contents of the buffer. Each round also makes use of an additive constant K_t , where $0 < t < 79$.

The output of the fourth round (eightieth step) is added to the input to the first round (CV_q) to produce CV_{q+1} . The addition is done independently for each of the five words in the buffer with each of the corresponding words in CV_q , using addition modulo 2^{32} .

Step 5: Output: After all L 512-bit blocks have been processed, the output from the L^{th} stage is the 160-bit message digest.

We can summarize the behavior of SHA-1 as follows:

$$CV_0 = IV$$

$$CV_{q+1} = \text{SUM}_{32}(CV_q, \text{ABCDE}_q)$$

$$MD = CV_L$$

where

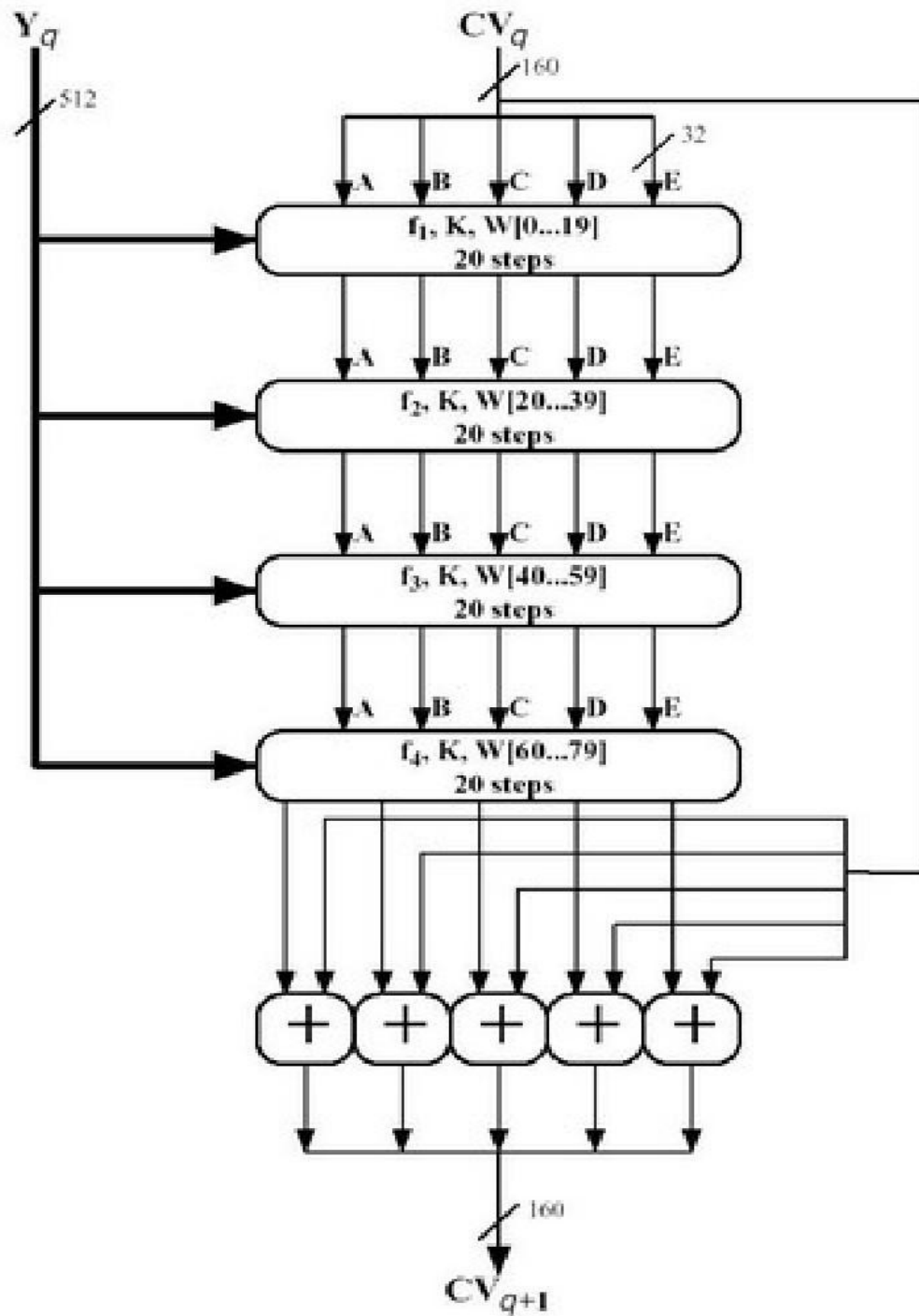
IV = Initial value of the ABCDE buffer

ABCDE_q = The output of the last round of the processing of the q th message Block

L = The number of blocks in the message

SUM_{32} = Addition modulo 2^{32} performed separately on each word of the pair of inputs.

MD = Final message digest value.



SHA-1 Compression Function:

The logic in each of the 80 steps of the processing of one 512-bit block. Each round is of the form (following figure)

$$A, B, C, D, E \leftarrow (E + f(t, B < C < D) + S^5(A) + W_t + K_t), A, S^{30}(B), C, D$$

Where A, B, C, D, E = the five words of the buffer

t = step number; $0 < t < 79$

$f(t, B, C, D)$ = primitive logical function for step t

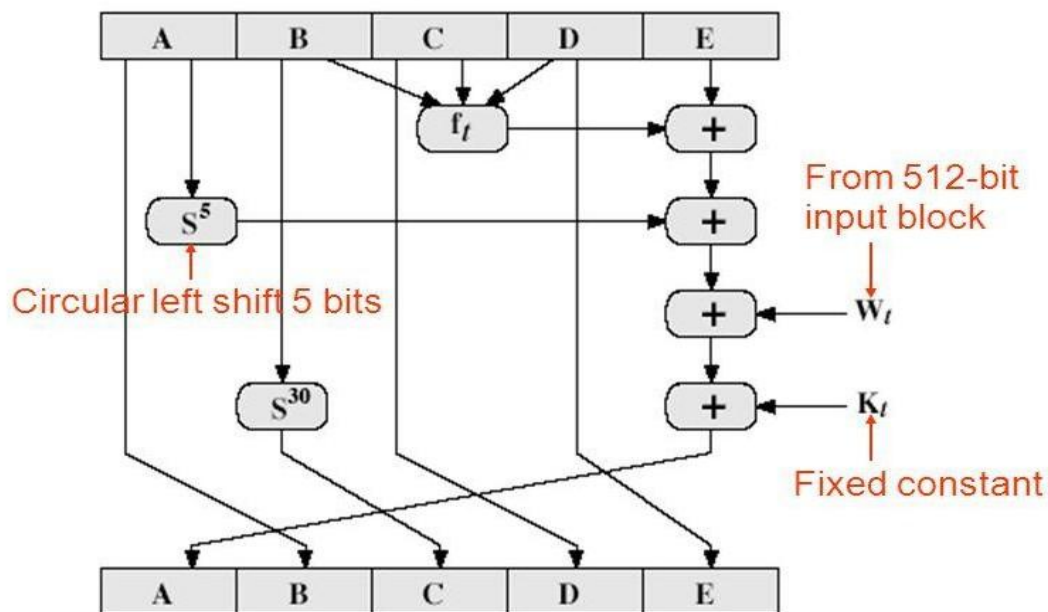
S^k = circular left shift of the 32-bit argument by k bits

W_t = a 32-bit word derived from the current 512-bit input block

K_t = an additive constant

$+$ = addition modulo 2^{32}

SHA1 Compression Function



Zhijun Li

S1034040/Autumn08/HIT

49

Each primitive function takes three 32-bit words as input and produces a 32-bit word output. Each function performs a set of bitwise logical operations.

System Security

Intruders and Types of Intruders

One of the two most publicized threats to security is the intruder (the other is viruses), often referred to as a hacker or cracker. In an important early study of intrusion, Anderson [ANDE80] identified three classes of intruders:

- **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 misuses 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 .

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. □

Password Guessing

The following techniques are used for learning passwords.

1. Try default passwords used with standard accounts that are shipped with the system. Many administrators do not bother to change these defaults.
2. Exhaustively try all short passwords.

3. Try words in the system's online dictionary or a list of likely passwords.
4. 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.
5. Try user's phone number, social security numbers and room numbers.
6. Try all legitimate license plate numbers.
7. Use a torjan horse to bypass restriction on access.
8. 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.
- ☐ **Prevention** – challenging security goal and an uphill battle at all times.

INTRUSION DETECTION:

The intrusion detection 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.

Approaches to intrusion detection:

1. 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.

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

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

2. 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.

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

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

Audit Records

A fundamental tool for intrusion detection is the audit record. Basically, two plans are used:

1. **Native audit records:** Virtually all multiuser operating systems include accounting software that collects information on user activity. The advantage of using this information is that no additional collection software is needed. The disadvantage is that the native audit records may not contain the needed information or may not contain it in a convenient form. □□

2. **Detection-specific audit records:** A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system. One advantage of such an approach is that it could be made vendor independent and ported to a variety of systems. The disadvantage is the extra overhead involved in having, in effect, two accounting packages running on a machine. □

Each audit record contains the following fields:

- **Subject:** Initiators of actions. A subject is typically a terminal user but might also be a process acting on behalf of users or groups of users. □
- **Object:** Receptors of actions. Examples include files, programs, messages, records, terminals, printers, and user- or program-created structures.

Statistical Anomaly Detection:

Statistical anomaly detection techniques fall into two broad categories: threshold detection and profile-based systems.

1. **Threshold detection involves** counting the number of occurrences of a specific event type over an interval of time. If the count surpasses what is considered a reasonable number that one might expect to occur, then intrusion is assumed.

Threshold analysis, by itself, is a crude and ineffective detector of even moderately sophisticated attacks. Both the threshold and the time interval must be determined.

2. **Profile-based anomaly** detection focuses on characterizing the past behavior of individual users or related groups of users and then detecting significant

deviations. A profile may consist of a set of parameters, so that deviation on just a single parameter may not be sufficient in itself to signal an alert.

Rule-Based Intrusion Detection

Rule-based techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious. There are two types of approaches for rule-based intrusion detection. They are

1. Rule-based anomaly detection

1.1 analyze historical audit records to identify usage patterns and to generate automatically rules that describe those patterns.

1.2 Then observe current behavior and match against rules to see if conforms

1.3 Like statistical anomaly detection does not require prior knowledge of security flaws

2. Rule-based penetration identification

2.1 uses expert systems technology

2.2 With rules identifying known penetration, weakness patterns, or suspicious behavior.

2.3 Rules usually machine and O/S specific

2.4 Rules are generated by experts who interview & codify knowledge of security admins.

2.5 Quality depends on how well this is done

2.6 Compare audit records or states against rules

The Base-Rate Fallacy

1. Practically an intrusion detection system needs to detect a large percentage of intrusions with few false alarms

1.1. If too few intrusions detected -> false security

1.2. If too many false alarms -> ignore / waste time

2. This is very hard to do

3. Existing systems seem not to have a good record.

Distributed Intrusion Detection

Generally, intrusion detection systems focused on single-system. But in general we have networked systems (LAN). So a more effective defense has these working together to detect intrusions. The main issues are:

- a. Dealing with varying audit record formats
- b. Integrity & confidentiality of networked data
- c. Centralized or decentralized architecture

The following distributed intrusion detection system diagram is developed at the University of California at Davis.

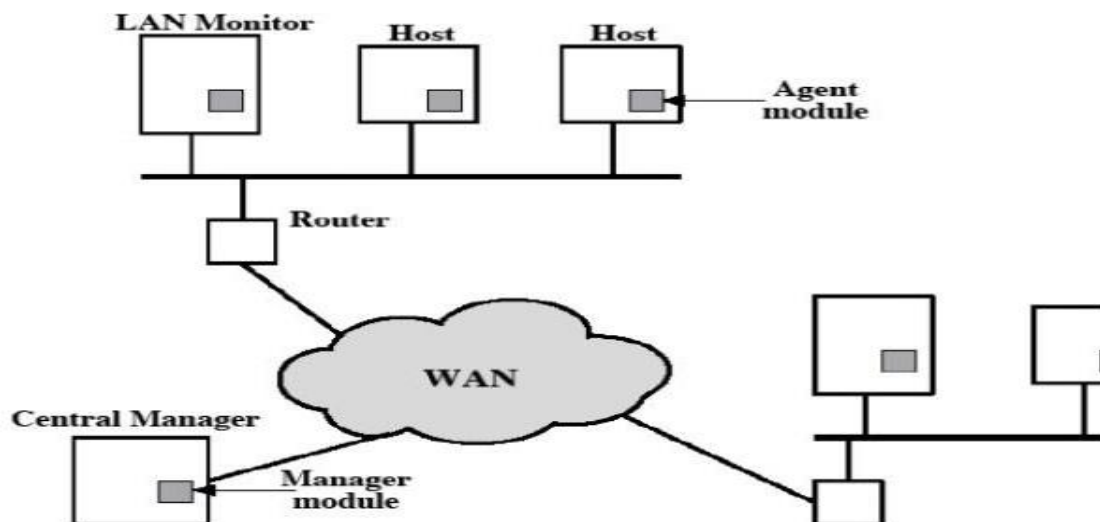
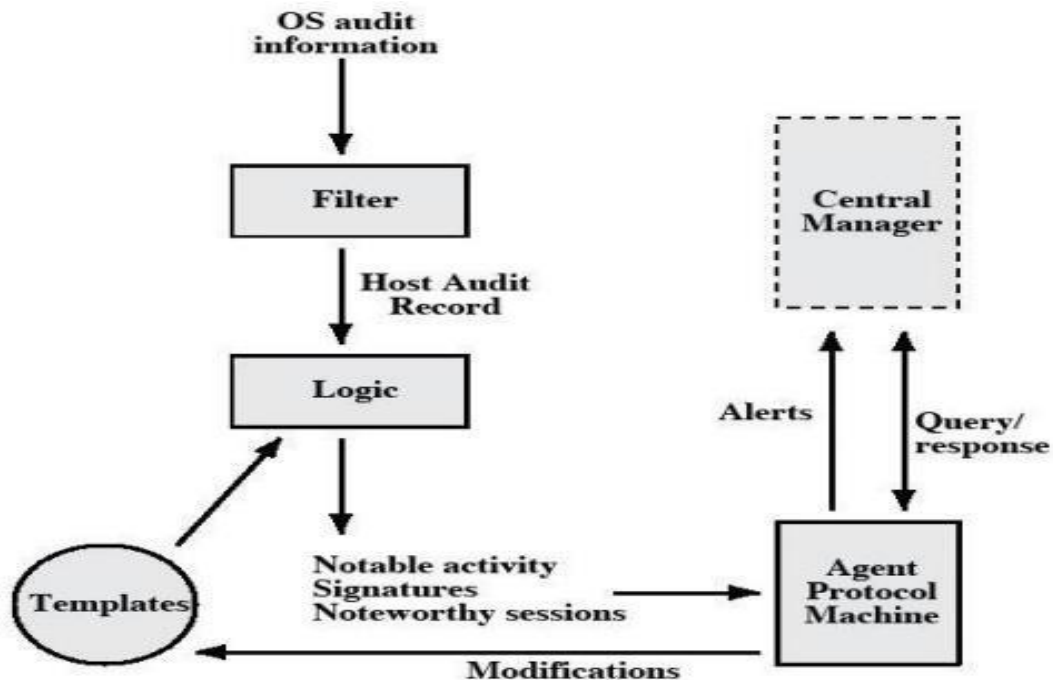


Fig. 5.2.3.1 Architecture of Distributed intrusion detection

The above diagram shows the overall architecture, which consists of three main components:

1. **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. □
2. **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.
3. **Central manager module:** Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion. □

The scheme is designed to be independent of any operating system or system auditing implementation. The agent captures each audit record produced by the native audit collection system. The following diagram shows the general approach of agent architecture.



- The agent captures each audit record produced by the native audit collection system.
- A filter is applied that retains only those records that are of security interest.
- These records are then reformatted into a standardized format referred to as the host audit record (HAR).
- Next, a template-driven logic module analyzes the records for suspicious activity.
- At the lowest level, the agent scans for notable events that are of interest independent of any past events.
- At the next higher level, the agent looks for sequences of events, such as known attack patterns (signatures).
- Finally, the agent looks for anomalous behavior of an individual user based on a historical profile of that user, such as number of programs executed, number of files accessed, and the like.
- The central manager includes an expert system that can draw inferences from received data.

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.

PASSWORD MANAGEMENT

1. Password Protection

The front line of defense against intruders is the password system. The password serves to authenticate the ID of the individual logging on to the system. The ID provides security in the following ways:

- The ID determines whether the user is authorized to gain access to a system. □
- The ID determines the privileges accorded to the user. □
- The ID is used in ,what is referred to as discretionary access control. For example, by listing the IDs of the other users, a user may grant permission to them to read files owned by that user. □

2. The Vulnerability of Passwords

To understand the nature of the threat to password-based systems, let us consider a scheme that is widely used on UNIX, the following procedure is employed.

- Each user selects a password of up to eight printable characters in length. •
- This is converted into a 56-bit value (using 7-bit ASCII) that serves as the key input to an encryption routine. •
- The encryption routine, known as crypt(3), is based on DES. The DES algorithm is modified using a 12-bit "salt" value.

- Typically, this value is related to the time at which the password is assigned to the user. ▪
- The modified DES algorithm is exercised with a data input consisting of a 64-bit block of zeros.
- The output of the algorithm then serves as input for a second encryption. ▪
- This process is repeated for a total of 25 encryptions. ▪
- The resulting 64-bit output is then translated into an 11-character sequence.
- The hashed password is then stored, together with a plaintext copy of the salt, in the password file for the corresponding user ID. ▪
- This method has been shown to be secure against a variety of cryptanalytic attacks

•3. Access Control

If the encrypted password portion of the file is accessible only by a privileged user, then the opponent cannot read it without already knowing the password of a privileged user.

Password Selection Strategies

Four basic techniques are in use:

- User education □
- Computer-generated passwords □
- Reactive password checking □
- Proactive password checking □

1. Educating users in Password selection: It is required to educate the users to design a string password.

2. Using Computer Generated Passwords: UNIX operating system provides some predefined passwords which is mixture of alphabets and numerals. Even if the password is pronounceable. It is difficult to remember by the user.

3. Reactive password checking: A reactive password checking strategy is one in which the system periodically runs its own password cracker to find guessable passwords.

4. Proactive password checking: The □Proactive password checker verifies is that password feasible or not I.e., whether the password is a guessable password or not.

Viruses and Related Threats

Malicious Software:

Malicious Software is software that is intentionally included or inserted in a system for a harmful purpose.

Malicious software can be divided into two categories: those that need a host program, and those that are independent.

Malware: is a combination of 2 terms- Malicious and Software. So Malware basically means malicious software that can be an intrusive program code or anything that is designed to perform malicious operations on system. Malware can be divided in 2 categories:

1. Infection Methods
2. Malware Actions

The following diagram provides classification of software threats or malicious programs.

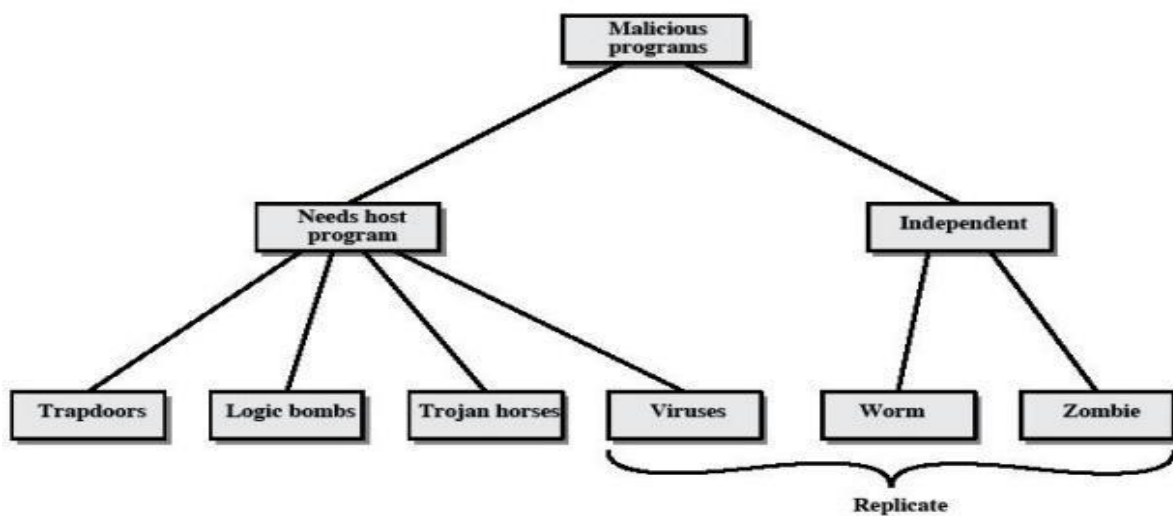


Figure 19.1 Taxonomy of Malicious Programs

- 1. Trapdoor:** An undocumented entry point intentionally written into a program, often for debugging purposes, which can be exploited as a security flaw.
- 2. Trojan Horse:** Instructions hidden inside an otherwise useful program that do undesirable things.
- 3. Logic Bomb:** Malicious instructions that trigger on some event in the future, such as a particular time occurring.

4. Virus: A set of instructions that, when executed, inserts copies of itself into other programs.

5. Worm: A program that propagates copies of itself to other computers.

6. Bacterium: A free-standing program that replicates itself, causing harm by consuming resources.

7. Zombie: Malicious instructions installed on a system that can be remotely triggered to carry out some attack with less traceability because the attack comes from another victim. Often the attacker installs large number of zombies to generate large bursts of network traffic.

The Nature of Viruses

A virus is a piece of software that can "infect" other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs. Once a virus is executing, it can perform any function, such as erasing files and programs.

During its lifetime, a typical virus goes through the following four phases:

- 1. Dormant phase:** The virus is idle. The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage. □
- **2. Propagation phase:** The virus places an identical copy of itself into other programs or into certain system areas on the disk. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase. □
- **3. Triggering phase:** The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself. □
- 4. Execution phase:** The function is performed. The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files. □

Virus Structure

A virus can be prepended or postpended to an executable program, or it can be embedded in some other fashion. The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program.

A very general description of virus structure is shown as the following. In this case, the virus code, V, is prepended to infected programs, and it is assumed that the entry point to the program, when invoked, is the first line of the program.

```
program V :=  
{ goto main;  
  1234567;  
  
  subroutine infect-executable :=  
    { loop:  
      file := get-random-executable-file;  
      if (first-line-of-file = 1234567)  
        then goto loop  
        else prepend V to file; }  
  
  subroutine do-damage :=  
    { whatever damage is to be done }  
  
  subroutine trigger-pulled :=  
    { return true if some condition holds }  
  
  main:  main-program :=  
    { infect-executable;  
      if trigger-pulled then do-damage;  
      goto next; }  
  next:  
  
}
```

Figure 21.1 A Simple Virus

When this program is invoked, control passes to its virus, which performs the following steps:

1. For each uninfected file P₂ that is found, the virus first compresses that file to produce P'₂, which is shorter than the original program by the size of the virus.
2. A copy of the virus is prepended to the compressed program.
3. The compressed version of the original infected program, P'₁, is uncompressed.
4. The uncompressed original program is executed.

The key lines in this virus are numbered. We assume that program P₁ is infected with the virus CV.

```

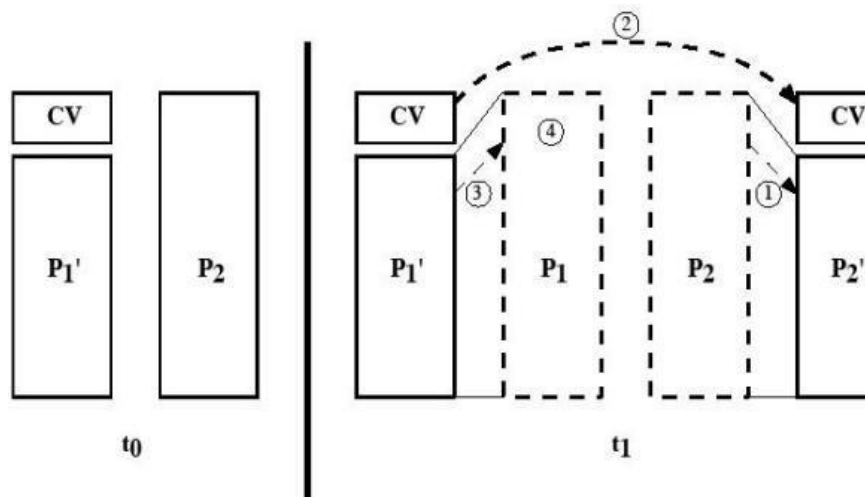
program CV :=
{goto main;
 01234567;

subroutine infect-executable :=
  {loop:
    file := get-random-executable-file;
    if (first-line-of-file = 01234567) then goto loop;
  (1) compress file;
  (2) prepend CV to file;
  }

main: main-program :=
  {if ask-permission then infect-executable;
  (3) uncompress rest-of-file;
  (4) run uncompressed file;}
  }

```

Figure 21.2 Logic for a Compression Virus



In this example, the virus does nothing other than propagate. As in the previous example, the virus may include a logic bomb.

Initial Infection

Once a virus has gained entry to a system by infecting a single program, it is in a position to infect some or all other executable files on that system when the

infected program executes. Thus, viral infection can be completely prevented by preventing the virus from gaining entry in the first place.

Types of Viruses

The viruses are classified into six types. They are

1. **Parasitic virus:** The traditional and still most common form of virus. A parasitic virus attaches itself to executable files and replicates, when the infected program is executed, by finding other executable files to infect. □
2. **Memory-resident virus:** Lodges in main memory as part of a resident system program. From that point on, the virus infects every program that executes. □
3. **Boot sector virus:** Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus. □
4. **Stealth virus:** A form of virus explicitly designed to hide itself from detection by antivirus software. □
5. **Polymorphic virus:** A virus that mutates with every infection, making detection by the "signature" of the virus impossible. □
6. **Metamorphic virus:** As with a polymorphic virus, a metamorphic virus mutates with every infection. The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection. Metamorphic viruses may change their behavior as well as their appearance. □

Macro Viruses

In the mid-1990s, macro viruses became by far the most prevalent type of virus. Macro viruses are particularly threatening for a number of reasons:

1. A macro virus is platform independent. Virtually all of the macro viruses infect Microsoft Word documents. Any hardware platform and operating system that supports Word can be infected.
2. Macro viruses infect documents, not executable portions of code. Most of the information introduced onto a computer system is in the form of a document rather than a program.
3. Macro viruses are easily spread. A very common method is by electronic mail.

Macro viruses take advantage of a feature found in Word and other office applications such as Microsoft Excel, namely the macro. In essence, a macro is an executable program embedded in a word processing document or other type of file.

Antivirus Approaches

The ideal solution to the threat of viruses is prevention: The next best approach is to be able to do the following:

- **Detection:** Once the infection has occurred, determine that it has occurred and locate the virus. □
- **Identification:** Once detection has been achieved, identify the specific virus that has infected a program. □
- **Removal:** Once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state. Remove the virus from all infected systems so that the disease cannot spread further. □

There are four generations of antivirus software:

- First generation: simple scanners □
- Second generation: heuristic scanners
- Third generation: activity traps □
- Fourth generation: full-featured protection □

1. A first-generation scanner requires a virus signature to identify a virus.. Such signature-specific scanners are limited to the detection of known viruses. Another type of first-generation scanner maintains a record of the length of programs and looks for changes in length.

2. A second-generation scanner does not rely on a specific signature. Rather, the scanner uses heuristic rules to search for probable virus infection. One class of such scanners looks for fragments of code that are often associated with viruses.

3. Third-generation programs are memory-resident programs that identify a virus by its actions rather than its structure in an infected program.

4. Fourth-generation products are packages consisting of a variety of antivirus techniques used in conjunction. These include scanning and activity trap components. In addition, such a package includes access control capability, which limits the ability of viruses to penetrate a system and then limits the ability of a virus to update files in order to pass on the infection.

Advanced Antivirus Techniques

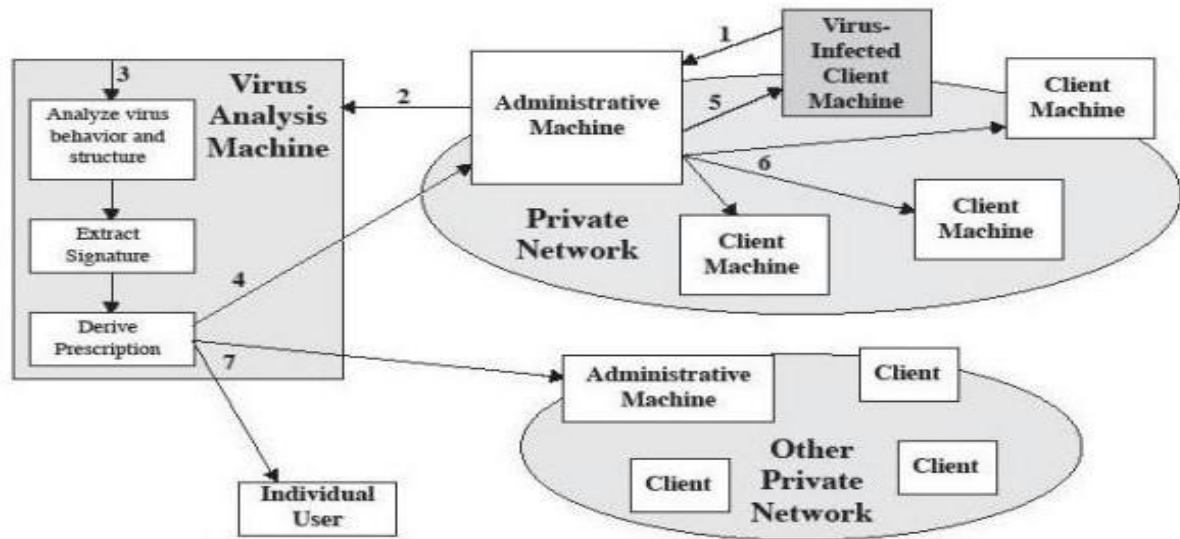
The most important advanced antivirus techniques are

1. Generic Decryption: Generic decryption (GD) technology enables the antivirus program to easily detect even the most complex polymorphic viruses, while maintaining fast scanning speeds. This contains the following elements:

- a) **CPU emulator:** A software-based virtual computer. Instructions in an executable file are interpreted by the emulator rather than executed on the underlying processor. The emulator includes software versions of all registers and other processor hardware, so that the underlying processor is unaffected by programs interpreted on the emulator. □□
- b) **Virus signature scanner:** A module that scans the target code looking for known virus signatures. □
- c) **Emulation control module:** Controls the execution of the target code. □

2. Digital Immune System: The digital immune system is a comprehensive approach to virus protection developed by IBM]. The motivation for this development has been the rising threat of Internet-based virus propagation. Two major trends in Internet technology have had an increasing impact on the rate of virus propagation in recent years:

- a) **Integrated mail systems:** Systems such as Lotus Notes and Microsoft Outlook make it very simple to send anything to anyone and to work with objects that are received.
- b) **Mobile-program systems:** Capabilities such as Java and ActiveX allow programs to move on their own from one system to another.



The above figure illustrates the typical steps in digital immune system operation.

1. A monitoring program on each PC uses a variety of heuristics based on system behavior, suspicious changes to programs, or family signature to infer that a virus may be present. The monitoring program forwards a copy of any program thought to be infected to an administrative machine within the organization.
2. The administrative machine encrypts the sample and sends it to a central virus analysis machine.
3. This machine creates an environment in which the infected program can be safely run for analysis. Techniques used for this purpose include emulation, or the creation of a protected environment within which the suspect program can be executed and monitored. The virus analysis machine then produces a prescription for identifying and removing the virus.
4. The resulting prescription is sent back to the administrative machine.
5. The administrative machine forwards the prescription to the infected client.
6. The prescription is also forwarded to other clients in the organization.
7. Subscribers around the world receive regular antivirus updates that protect them from the new virus.

Trusted systems

One way to enhance the ability of a system to defend against intruders and malicious programs is to implement trusted system technology.

Sometimes we need to protect our data on the basis of levels, I.e., entire data is accessible for MD (Managing Director), some other data is accessible for Managers and some other data is accessible by Accounts Department. This is generally happens in the defense where we have to classify the data into confidential, secret, top secret data.

Trusted system provides such type of Multilevel Security. The Multilevel Security is defined in the following way. In the Multilevel Security, we have subjects, objects, access rights and two important principles. 1) No Read Up 2) No Write Down.

Subject: An entity capable of accessing objects. Generally, the concept of subject equates with that of process.

Object: Anything to which access is controlled. Examples include files, portion of files, programs, and segments of memory.

Access right: The way in which the object is accessed by a subject. Examples are read, write and execute.

Subject Object	File1	File2	DB1	DB2	Seg. A	Seg. B
User1	R	W	RW	RWE		
User2	W		R	E		
Process1	R	W	E	E	R	W
Process2	RW	WE	ER	R	W	E

**The concept of
Trusted
Systems**

There are the two principles that are used in the Multilevel Security.

No read up: A subject can only read an object of less or equal security level. This is referred to as simple security property.

No write down: A subject can only write into an object of greater or equal security level.

The trusted system based on reference monitor concept will provide Multilevel Security. This is shown in the following diagram.

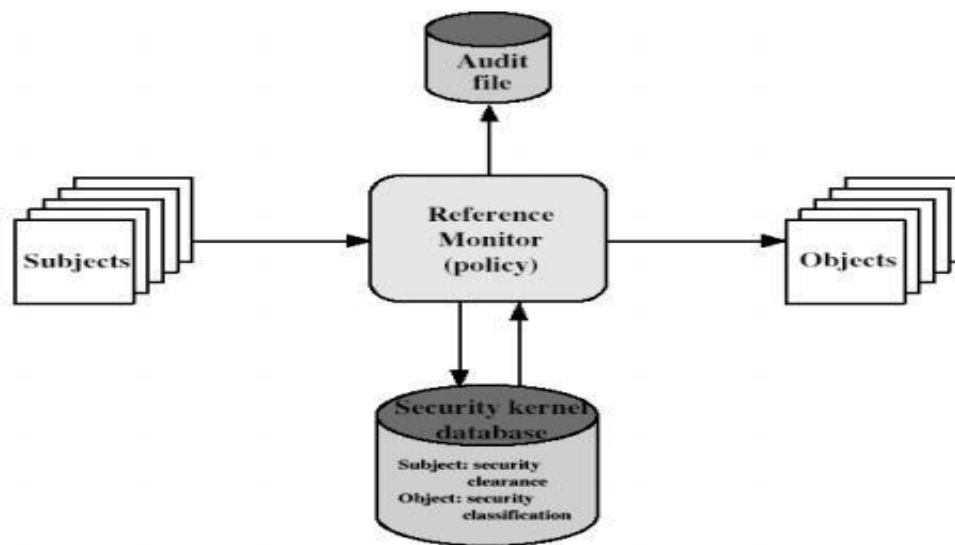


Fig:Reference Monitor Concept

Reference Monitor concept

Here, the reference monitor controls the access of objects by the subjects according to the security parameters that are defined in the security kernel database. Here, all the access, no read up and no write down principles are programmed in the security kernel database. is a controlling element in the hardware and operating system of a computer. The following are the properties of Reference Monitor:

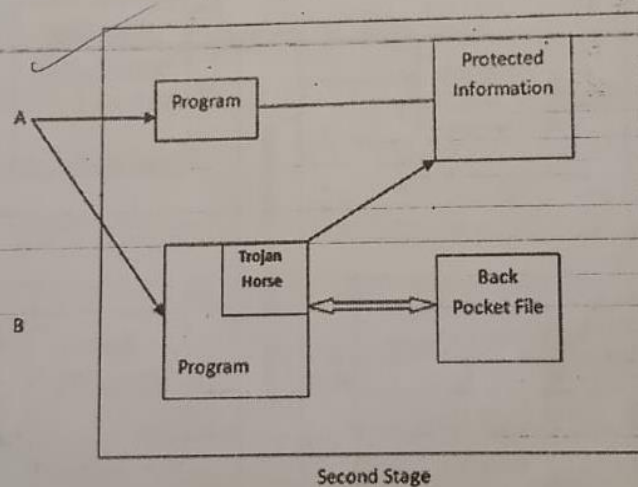
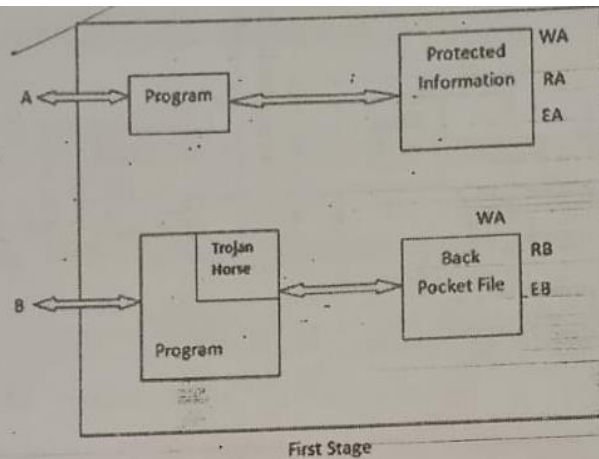
1. **Complete mediation:** The security rules are enforced (applied) to each and every request.
2. **Isolation:** The reference monitor and database are protected from unauthorised modification.
3. **Verifiability:** The reference monitor's correctness must be provable.

As the reference monitor is capable of providing verifiability, we call it as a trusted system. The final element in the reference monitor concept Audit File.

Audit File: The reference monitor keeps all the transactions that are granted or that are stored in the audit file. It also keeps the information about illegal operations performed by the users in the audit file. All the security policies that are having some drawbacks are also written in the audit file. Based on the audit file the administrators may update the kernel database.

Trojan Horse and Defense

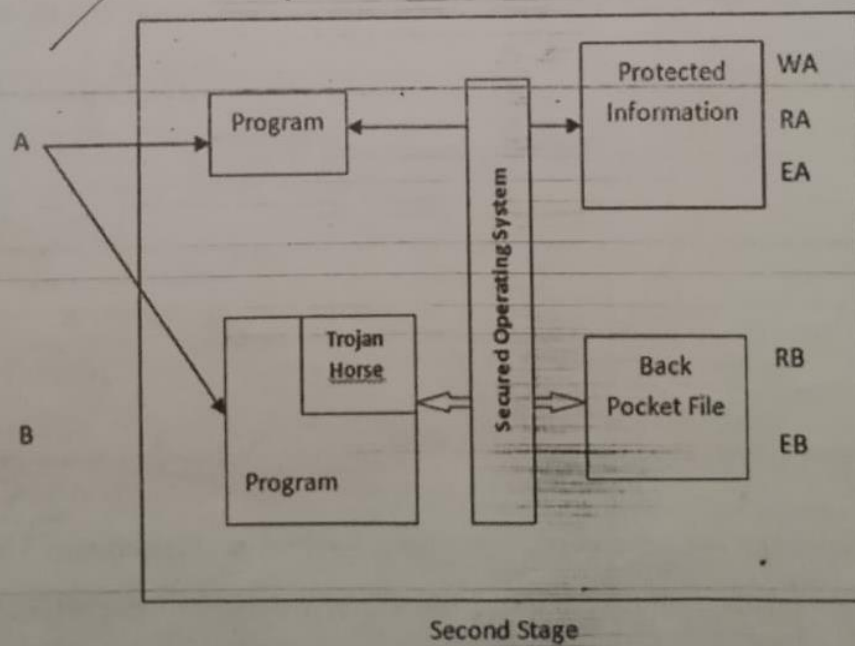
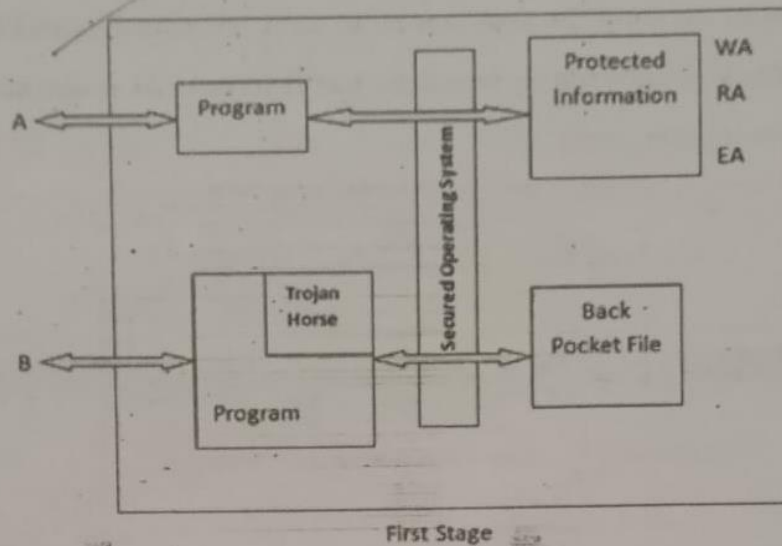
Trojan horse attack begins when an illegal user wants to access the information from a legal users login before him. After a legal user 'A' gains access to system, 'B' comes to him and asks to run a file or program in the floppy. The program internally contains another file called Trojan horse. When the program is executed by the legal user 'A', internally Trojan horse is also executed. Its job is to create a file called Back Pocket file. On this file, 'Write' right is given to the user who logs in I.e., 'A'. The Trojan Horse program copies all the protected information from the user A's login into back pocket file. After completion of copying the program ends, user 'B' comes with his floppy. Now the floppy contains the back pocket file in which he has the required information. This attack is explained in the following diagram:



Let us consider the use of secured Operating System in this scheme. Here, we have to define security levels for the subjects and objects. Here, we use two important security levels: They are a) Public b) Sensitive.

Sensitive is higher than Public. Whenever user 'A' logs in his program, his data (protected) are given Sensitive security level. All other programs that he will execute, which are not owned by him are given Public security level. So, user B's program gets public security level. Whenever user 'A' executes B's program, it executes, the Trojan Horse program is also executed. Its job is read information, from the user A which is assigned Sensitive Security level. We have the principle "No Read up", in the security Operating System. So, the Trojan Horse program which is having public security level is not able to

read user A's protected data which is assigned sensitive security level. As a result, Back Pack file is created and nothing written into it. This is shown in the following diagram:



UNIT-IV

Internet Security Protocols

Basic Concepts

Internet security refers to securing communication over the internet. It includes specific security protocols such as:

Internet Security Protocol (IPSec)

Secure Socket Layer (SSL)

1) Internet Security Protocol (IPSec)

It consists of a set of protocols designed by Internet Engineering Task Force (IETF). It provides security at network level and helps to create authenticated and confidential packets for IP layer.

2) Secure Socket Layer (SSL)

It is a security protocol developed by Netscape Communications Corporation. It provides security at transport layer. It addresses the following security issues: Privacy, Integrity, Authentication

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets.

Web page is a document available on World Wide Web. Web pages are stored on web server and can be viewed using a web browser. A web page can contain huge information including text, graphics, audio, video and hyperlinks. These hyperlinks are the link to other web pages. Collection of linked web pages on a web server is known as website.

Web pages are classified into three types. They are

1. Static Web Pages: These are also known as flat or stationary web pages. They are loaded on the client's browser as exactly they are stored on the web server. Such web pages contain only static information.

2. Dynamic Web Pages: Dynamic web page shows different information at different point of time. It is possible to change a portion of a web page without loading the entire web page. It has been made possible using **Ajax** (Asynchronous JavaScript and XML) technology.

Dynamic web pages are classified into two types. They are:

- a. Server-Side Dynamic Web Page
- b. Client-Side Dynamic Web Page

3. **Active Web Pages:** As active web page is a page where the browser performs the logic instead of the server. When a client sends an HTTP (Hypertext Transfer Protocol) request for an active web page, the web server sends back an HTTP response that contains an HTML page as usual.

Protocol, Internet Protocol and Packet:

In information technology, a **protocol** is the special set of rules that end points in a telecommunication connection use when they communicate. Protocols specify interaction between the communicating entities.

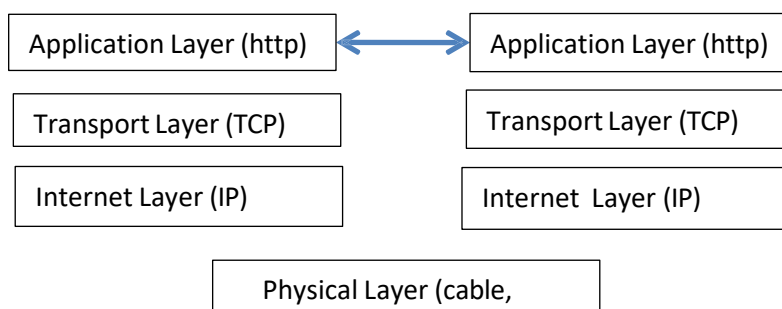
The **Internet Protocol (IP)** is the method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one IP address that uniquely identifies it from all other computers on the Internet.

A Packet is the unit of data that is routed between an origin and a destination on the Internet or any other packet-switched network.

TCP/IP:

The Transmission Control Protocol/Internet Protocol(TCP/IP), is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP can also be used as a communications protocol in a private network.

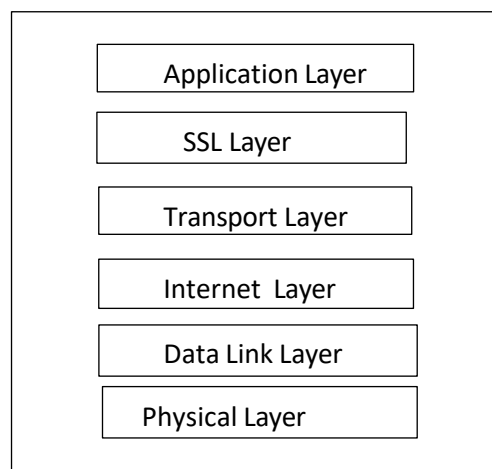
The whole Internet runs on these two protocols IP and TCP. These two protocols provide all of the requirements to transmit and receive data across complex WANs (Wide Area Network). Networks are made of layers and each layer providing a specific functionality. The following figure describes layers of a network protocol:



1. The Physical Layer is made from the actual hardware (cables, network interface cards etc,...) and the drivers which are required to run that hardware.
2. The Application Layer represents the application which we are running. In our case the application is the WEB and the application layer is HTTP (Hyper Text Transfer Protocol).
3. The Transport Layer establishes a reliable communication stream between a pair of systems across a network by putting sequence numbers in packets, holding packets at the destination until they can be delivered in order and re-transmitting lost packets.
4. The Internet Layer or IP layer is a group of internetworking methods, protocols, and specifications in the Internet protocol suite that are used to transport datagrams (packets) from the originating host across network boundaries.
5. Internet Layer protocols used IP-based packets.

SSL

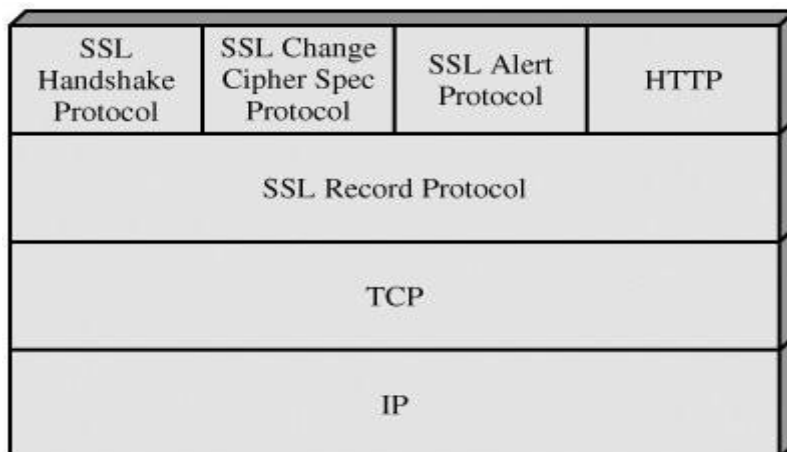
Secure Sockets Layer (SSL) protocol is an Internet Protocol for secure exchange of information between a web browser and a web server. It provides two basic security services: authentication and confidentiality. Logically, it provides a secure pipe between the web browser and the web server. This is developed by Netscape Corporation developed in 1994 and SSL become the world's most popular web security mechanism. All the major web browser support SSL. SSL comes in three versions: 2, 3 and 3.1. The most popular version is 3 and is released in 1995. SSL is located in between application and transport layers. This is shown in the following figure:



Position of SSL in TCP/IP

SSL is designed to make use of TCP to provide reliable service. It is not a single protocol. It consists two layers of protocol as shown in the following figure:

SSL Protocol Stack



The SSL Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction. There are 3 higher level SSL protocols. They are

1. SSL Alert Protocol,
2. SSL Change Cipher Spec Protocol
3. SSL Handshake Protocol.

There are 2 important SSL concepts:

- a. SSL Connection b. SSL Session

- **Connection:** A connection is a transport that provides transport service. In SSL, the connections are end-to-end and are associated with session.
- **Session:** A session is created by handshaking protocol. It contains a set of cryptographic security parameters which are shared by the connection.

Between any pair of parties (Systems), there may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but generally we use only one connection between parties & that connection is fully secured. A session is defined by the following parameters:

- a) Session identifier b) Peer certificate c) Compression method
- c) Cipher specification e) Master secret value

A connection state is defined by the following parameters:

1. **Client & Server**
2. **Server write MAC secret**
3. **Client write MAC secret**
4. **Server write key** - The conventional encryption key for data encrypted by the server and decrypted by the client.
5. **Client write key** - The conventional encryption key for data encrypted by the client and decrypted by the server.
6. **Initialization vectors** - When a block cipher in CBC mode is used, an initialization vector (IV) is used.

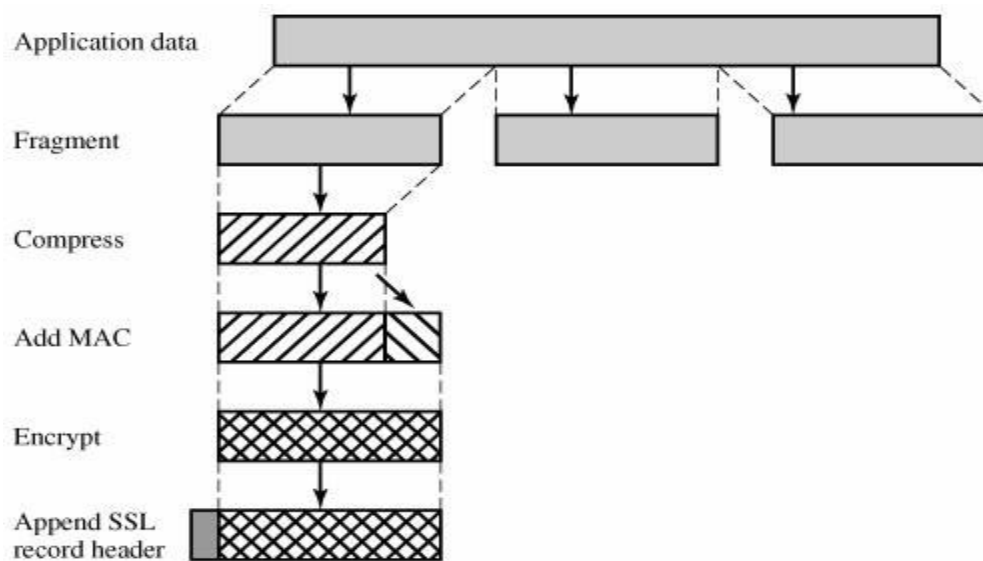
7. **Sequence numbers** - Each party maintains separate sequence numbers for sending messages and received messages.

SSL Record Protocol

The SSL Record Protocol provides two services for SSL connections:

- **Confidentiality:** The Handshake Protocol defines a shared secret key that is used for encryption of SSL payloads. It provides confidentiality.
- **Message Integrity:** The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC) which provides integrity. The following [Figure](#) indicates the overall operation of the SSL Record Protocol.

This Record Protocol takes message to be transmitted, fragments it into several blocks, optionally compresses each block, apply a MAC, append MAC at the end of each block, encrypt each block. Finally append SSL record header to each block and transmit to destination which is shown in the following figure:



At the receiving end SSL record protocol performs remove header part, decrypt, verifies MAC values, and optionally decompresses to get a fragment. Such fragments are reassembled to get the message and it is delivered to the end user.

In the SSL record protocol, MAC is calculated for the compressed fragment or uncompressed fragment. To calculate this, we require a key shared between two parties. The MAC value is calculated using the following formulae:

```

hash(MAC_write_secret || pad_2 ||
hash(MAC_write_secret || pad_1 || seq_num ||
SSLCompressed.type ||
SSLCompressed.length || SSLCompressed.fragment))

```

where

|| = concatenation

MAC_write_secret = shared secret key

hash = cryptographic hash algorithm; either MD5 or SHA-1

pad_1 = the byte 0x36 (0011 0110) repeated 48 times (384 bits) for MD5 and
40 times (320 bits) for SHA-1

pad_2 = the byte 0x5C (0101 1100) repeated 48 times for MD5 and 40 times for
SHA-1

seq_num = the sequence number for this message

SSLCompressed.type = the higher-level protocol used to process this fragment

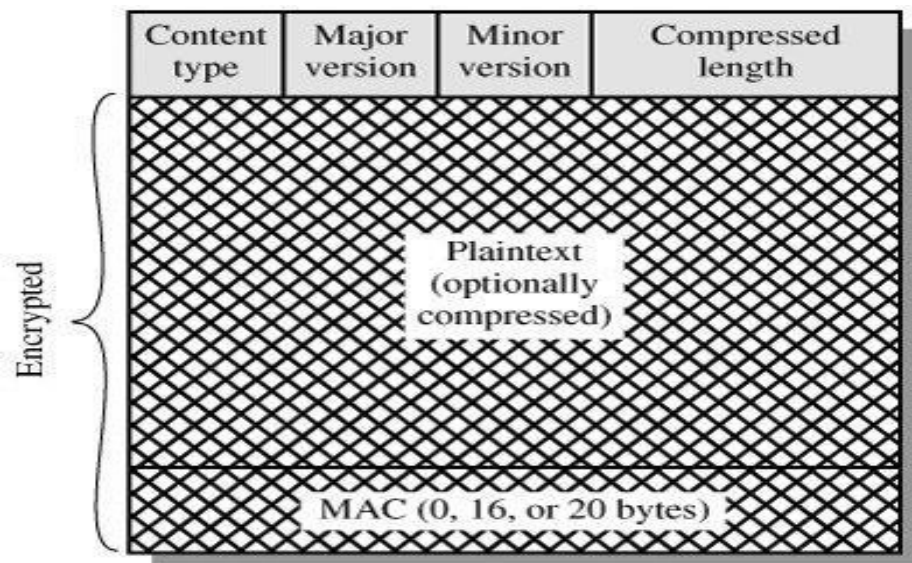
SSLCompressed.length = the length of the compressed fragment

SSLCompressed.fragment = the compressed fragment.

The final step of SSL Record Protocol processing is to prepend a header, consisting of the following fields:

- a) **Content Type (8 bits):** The higher layer protocol used to process the enclosed fragment.
- b) **Major Version (8 bits):** Indicates major version of SSL protocol used & its value is 3.
- c) **Minor Version (8 bits):** Indicates minor version SSL protocol used & its the value is 0.
- d) **Compressed Length (16 bits):** Indicates length of the compressed fragment after padding MAC.

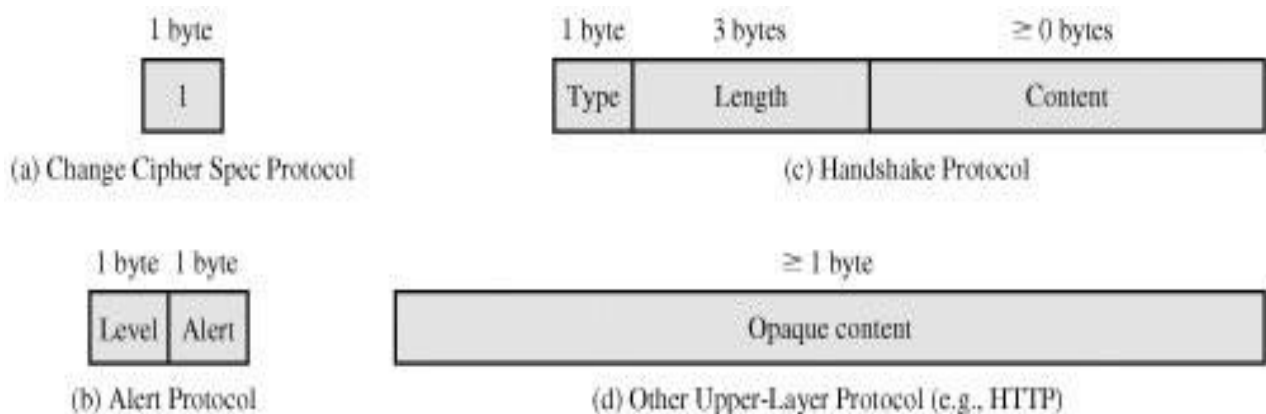
The following figure shows the SSL record format;



Change Cipher Spec Protocol

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

Figure: SSL Record Protocol Payload



Alert Protocol

The Alert Protocol is used to convey SSL-related alerts to the peer entity. As with other applications that use SSL, alert messages are compressed and encrypted, as specified by the current state. Each message in this protocol consists of two bytes (Figure (b)). The first byte takes the value warning(1) or fatal(2) to convey the severity of the message. If the level is fatal, SSL immediately terminates the connection.

Other connections on the same session may continue, but no new connections on this session may be established. The second byte contains a code that indicates the specific alert. First, we list those alerts that are always fatal (definitions from the SSL specification):

- **unexpected_message:** An inappropriate message was received.
- **bad_record_mac:** An incorrect MAC was received.
- **decompression_failure:** The decompression function received improper input
- **handshake_failure:** Sender was unable to negotiate an acceptable set of security parameters.
- **illegal_parameter:** A field in a handshake message was out of range or inconsistent with other fields.

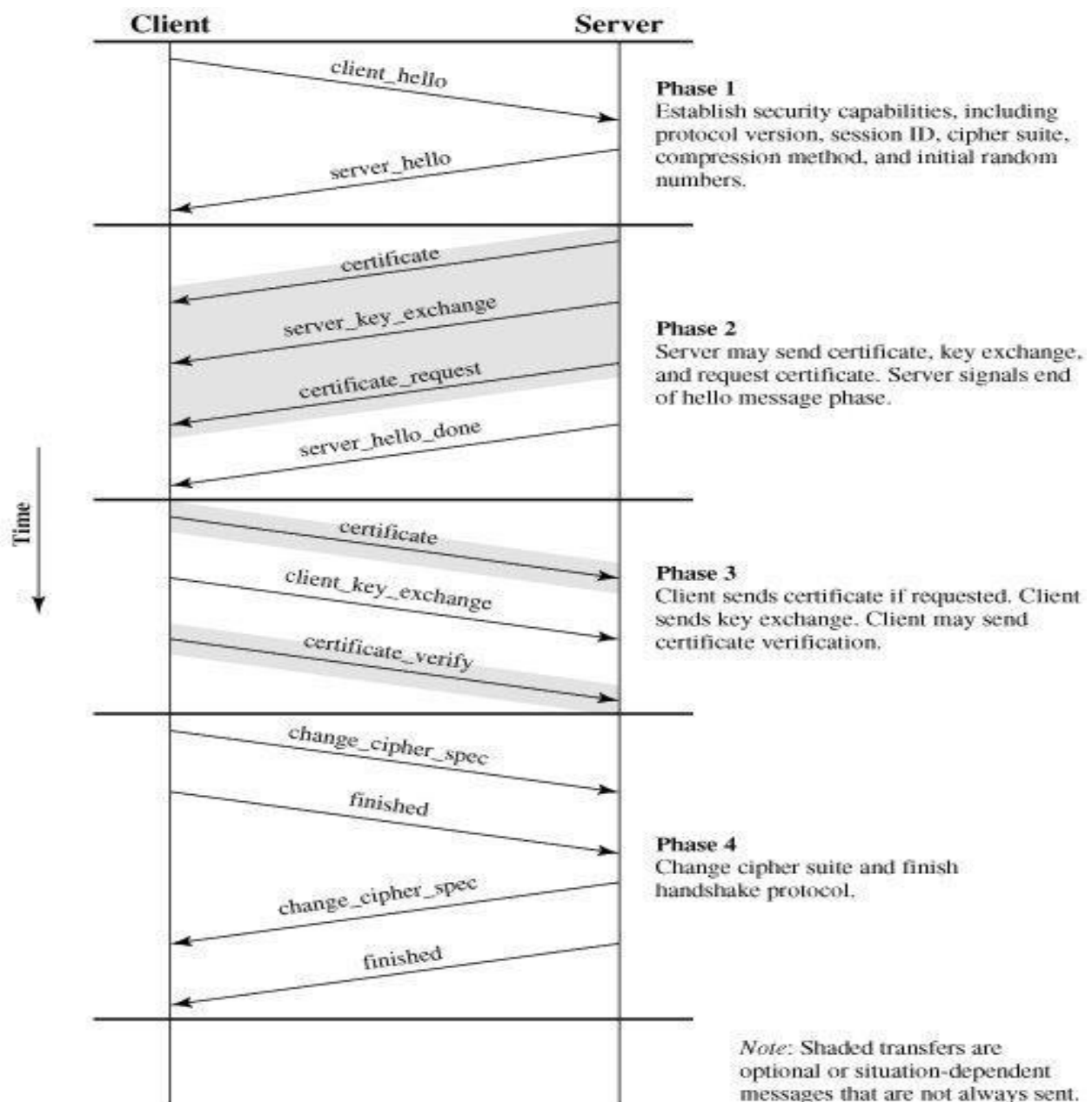
Handshake Protocol

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

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities

server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

Figure: Handshake Protocol Action



Transport Layer Security

TLS is an Internet standard version of SSL. The current version of TLS is very similar to the SSL v3 with slight differences:

Version Number: The TLS Record Format is the same as that of the SSL Record Format. But the major version is 3 and minor version is 1 for TLS.

Message Authentication Code: The MAC value for TLS is also calculated in different. Here we use HMAC algorithm.

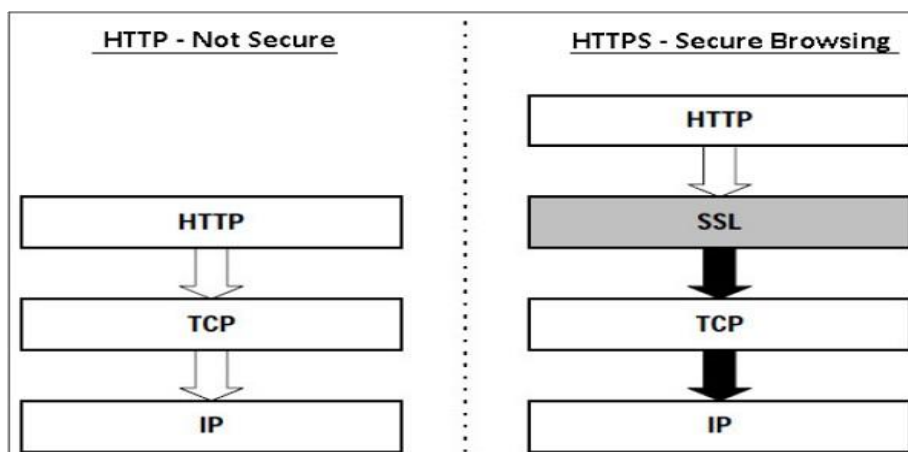
SHTTP

Secure Hypertext Transfer Protocol (S-HTTP) is an obsolete alternative to the HTTPS protocol for encrypting web communications carried over HTTP. It was developed by Eric Rescorla and Allan M. Schiffman, and published in 1999 as RFC 2660.

The SHTTP is a set of security of Mechanisms defined for protecting the internet traffic. This includes data entry forms and Internet based transactions. Each sHTTP file is either encrypted, contains a digital certificate, or both. A major difference is that S-HTTP allows the client to send a certificate to authenticate the user whereas using SSL, only the server can be authenticated.

HTTPS Defined

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

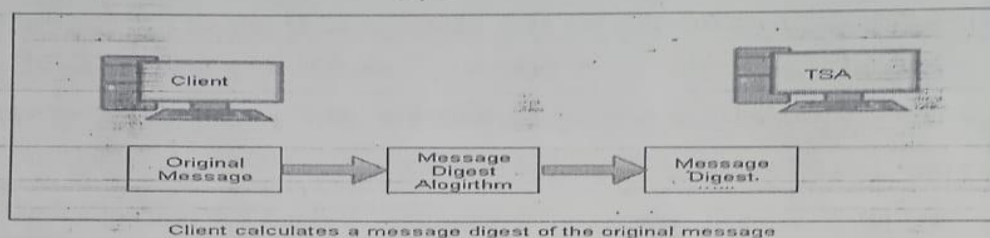


TSP

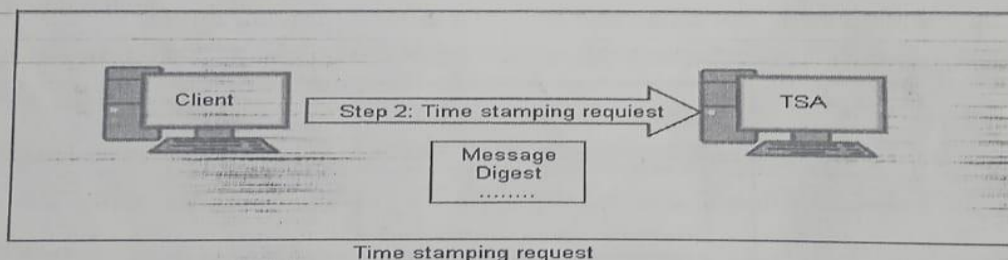
Time Stamping Protocol (TSP):

This provides a proof that a certain piece of data existed at a particular time. This Public Key Infrastructure (PKI) service is provided by an authority called as Time Stamping Authority (TSA). TSP is currently under the development of the PKIX working group. The TSP is a simple request response protocol, similar to HTTP. This work as described below, step-by-step:

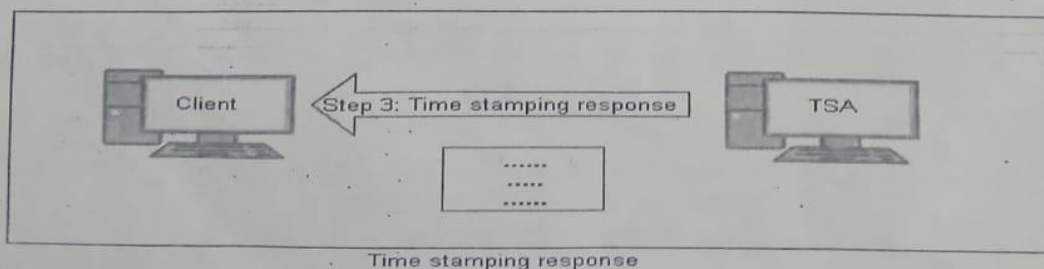
Step 1: Message digest calculation: Firstly, the entry (client) requiring a timestamp calculates a message digest of the original message, which needs a timestamp from the TSA. The client should use a standard message digest algorithm, such as MD5 or SHA-1 for this purpose. This is shown in the following figure:



Step 2: Timestamping request: Now, the client sends the message digest calculated in the step 1 to the Time Stamp Authority (TSA) for getting a Timestamped and is known as Time Stamping Request. This is shown in the following figure:



Step 3: Time stamping response: In response to the client's request, The TSA might decide to grant or reject the timestamp. If it decides to accept the request and process it, it signs the client's request together with the timestamp by the TSA private key. Regardless, it returns a Time Stamping Response back to the client. This is shown in the following figure:



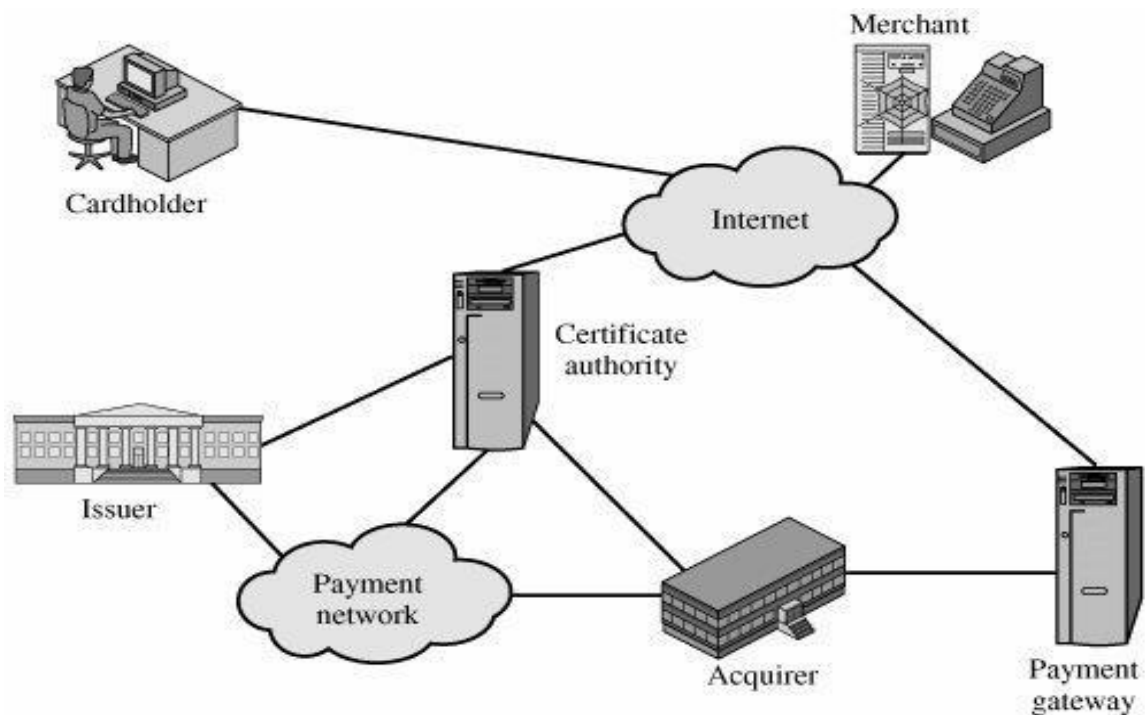
SET

Secure Electronic Transaction or SET is an open encryption and security specification designed to protect credit card transactions on the Internet. The current version, SETv1, emerged from a call for security standards by MasterCard and Visa in February 1996. A wide range of companies were involved in developing the initial specification, including IBM, Microsoft, Netscape, RSA, Terisa, and Verisign. SET provides three services:

1. Provides a secure communications channel among all parties involved in a transaction
2. Provides trust by the use of X.509v3 digital certificates
3. Ensures privacy because the information is only available to parties in a transaction when and where necessary

SET Participants (Components)

Figure Secure Electronic Commerce Components



The above Figure indicates the participants in the SET system, which include the following:

1. **Cardholder:** In the electronic environment, consumers and corporate purchasers interact with merchants from personal computers over the Internet. A cardholder is an authorized holder of a payment card (e.g., MasterCard, Visa) that has been issued by an issuer.

2. **Merchant:** A merchant is a person or organization that has goods or services to sell to the cardholder. Typically, these goods and services are offered via a Web site or by electronic mail. A merchant that accepts payment cards must have a relationship with an acquirer.

3. **Issuer:** This is a financial institution, such as a bank, that provides the cardholder with the payment card. Typically, accounts are applied for and opened by mail or in person. Ultimately, it is the issuer that is responsible for the payment of the debt of the cardholder.

4. **Acquirer:** This is a financial institution that establishes an account with a merchant and processes payment card authorizations and payments. Merchants will usually accept more than one credit card brand but do not want to deal with multiple bankcard associations or with multiple individual issuers. The acquirer provides authorization to the merchant that a given card account is active and that the proposed purchase does not exceed the credit limit. The acquirer also provides electronic transfer of payments to the merchant's account. Subsequently, the acquirer is reimbursed by the issuer over some sort of payment network for electronic funds transfer.

5. **Payment gateway:** This is a function operated by the acquirer or a designated third party that processes merchant payment messages. The payment gateway interfaces between SET and the existing bankcard payment networks for authorization and payment functions. The merchant exchanges SET messages with the payment gateway over the Internet, while the payment gateway has some direct or network connection to the acquirer's financial processing system.

6. **Certification authority (CA):** This is an entity that is trusted to issue X.509v3 public-key certificates for cardholders, merchants, and payment gateways. The success of SET will depend on the existence of a CA infrastructure available for this purpose. As was discussed in previous chapters, a hierarchy of CAs is used, so that participants need not be directly certified by a root authority.

SET Transaction:

We now briefly describe the sequence of events that are required for a transaction.

1. **The customer opens an account.** The customer obtains a credit card account, such as MasterCard or Visa, with a bank that supports electronic payment and SET.
2. **The customer receives a certificate.** After suitable verification of identity, the customer receives an X.509v3 digital certificate, which is signed by the bank. The certificate verifies the customer's RSA public key and its expiration date. It also establishes a relationship, guaranteed by the bank, between the customer's key pair and his or her credit card.
3. **Merchants have their own certificates.** A merchant who accepts a certain brand of card must be in possession of two certificates for two public keys owned by the merchant: one for signing messages, and one for key exchange. The merchant also needs a copy of the payment gateway's public-key certificate.
4. **The customer places an order.** This is a process that may involve the customer first browsing through the merchant's Web site to select items and determine the price. The customer then sends a list of the items to be purchased to the merchant, who returns an order form containing the list of items, their price, a total price, and an order number.
5. **The merchant is verified.** In addition to the order form, the merchant sends a copy of its certificate, so that the customer can verify that he or she is dealing with a valid store.
6. **The order and payment are sent.** The customer sends both order and payment information to the merchant, along with the customer's certificate. The order confirms the purchase of the items in the order form. The payment contains credit card details. The payment information is encrypted in such a way that it cannot be read by the merchant. The customer's certificate enables the merchant to verify the customer.
7. **The merchant requests payment authorization.** The merchant sends the payment information to the payment gateway, requesting authorization that the customer's available credit is sufficient for this purchase.

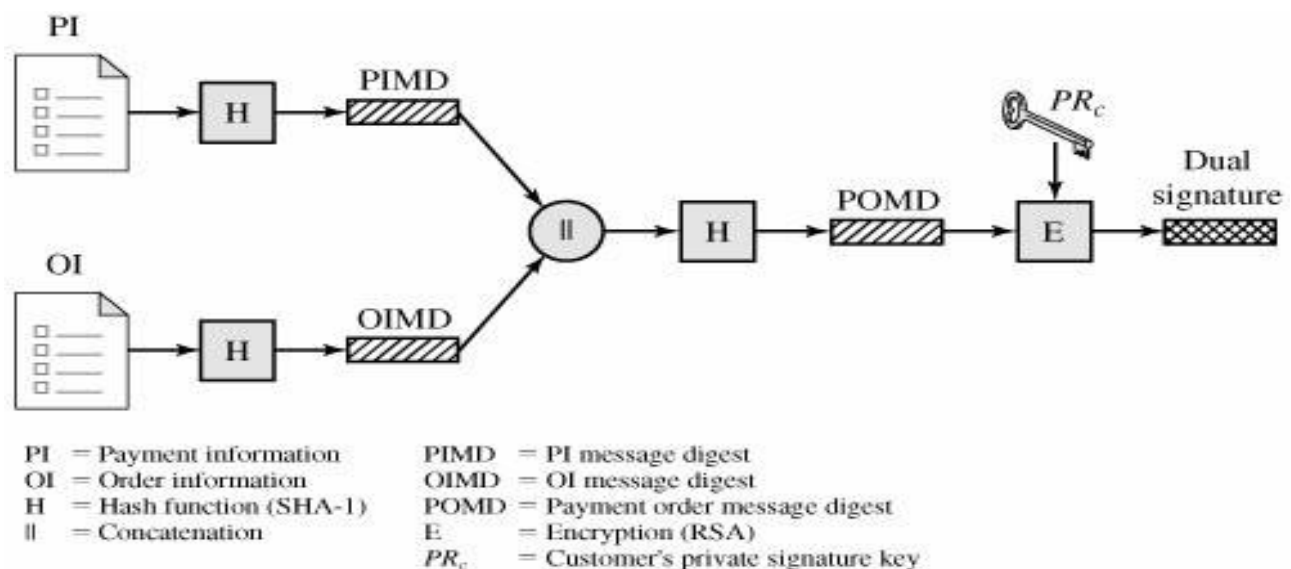
8. **The merchant confirms the order.** The merchant sends confirmation of the order to the customer.
9. **The merchant provides the goods or service.** The merchant ships the goods or provides the service to the customer.
10. **The merchant requests payment.** This request is sent to the payment gateway, which handles all of the payment processing

Dual Signature

The purpose of the dual signature is to link two messages that are intended for

two different recipients. In this case, the customer wants to send the order information (OI) to the merchant and the payment information (PI) to the bank. The merchant does not need to know the customer's credit card number, and the bank does not need to know the details of the customer's order.

Construction of Dual Signature

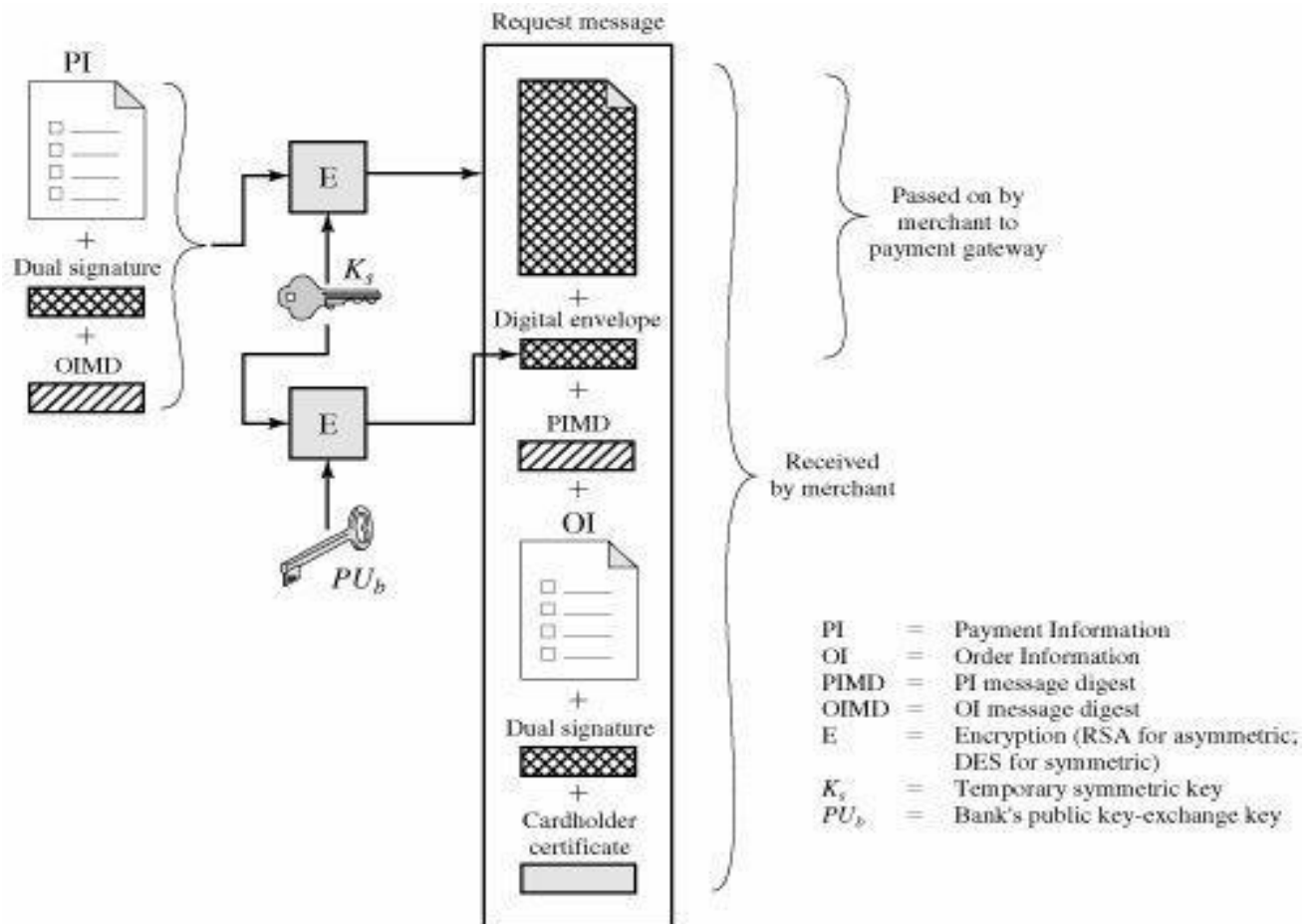


Purchase Request - Customer:

Before the Purchase Request exchange begins, the cardholder has completed browsing, selecting, and ordering. The end of this preliminary phase occurs when the merchant sends a completed order form to the customer. All of the preceding occurs without the use of SET.

The purchase request exchange consists of four messages: Initiate Request, Initiate Response, Purchase Request, and Purchase Response. This is explained in the following figure:

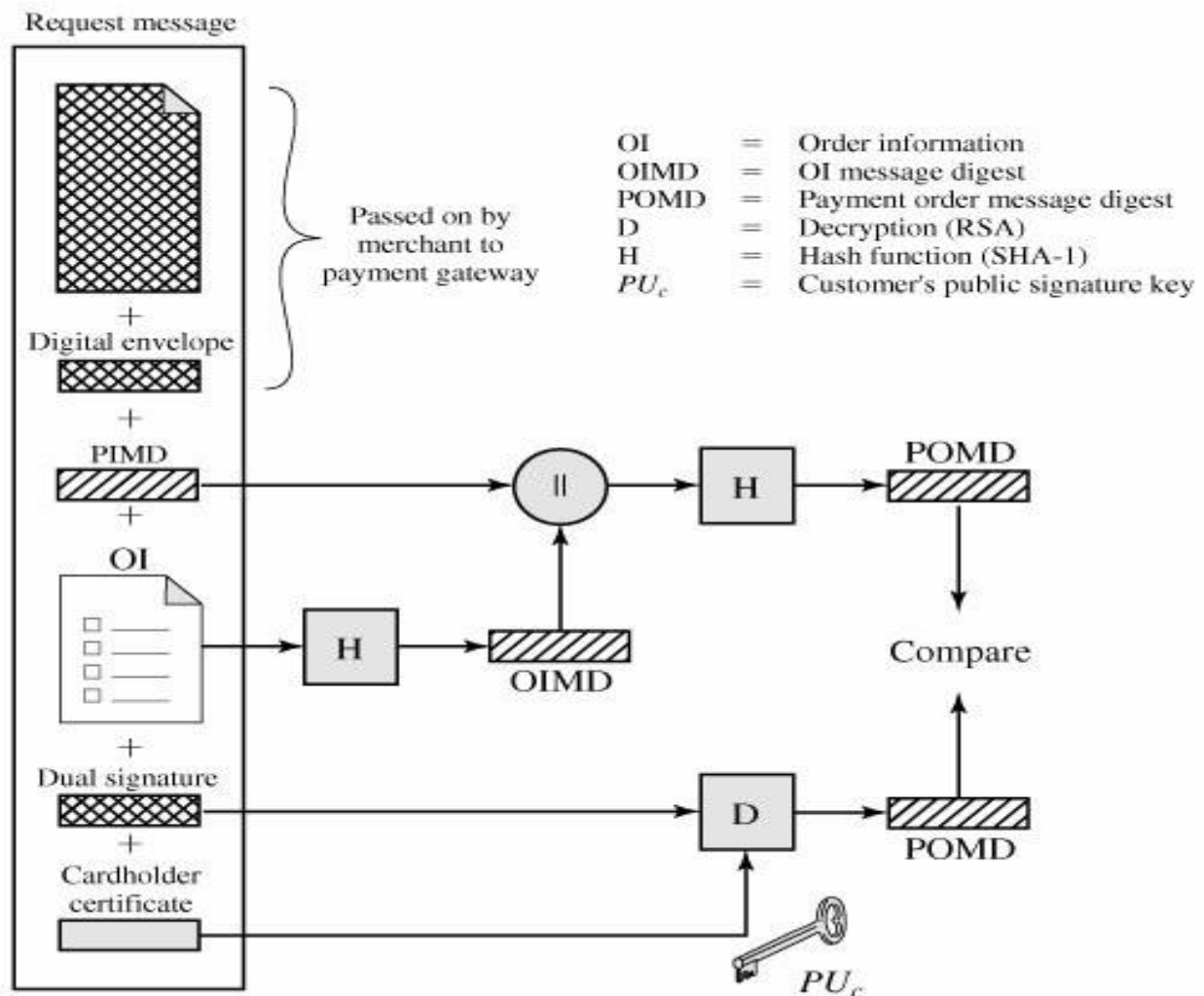
Cardholder sends Purchase Request



Purchase Request - Merchant:

1. Verifies the cardholder certificates by means of its CA signatures.
2. Verifies the dual signature using the customer's public signature key. This ensures that the order has not been tampered with in transit and that it was signed using the cardholder's private signature key.
3. Processes the order and forwards the payment information to the payment gateway for authorization (described later).
4. Sends a purchase response to the cardholder

Merchant Verifies Customer Purchase Request



Payment Gateway Authorization

1. Verifies all certificates
2. Decrypts the digital envelope of the authorization block to obtain the symmetric key and then decrypts the authorization block
3. Verifies the merchant's signature on the authorization block
4. Decrypts the digital envelope of the payment block to obtain the symmetric key and then decrypts the payment block
5. Verifies the dual signature on the payment block

6. Verifies that the transaction ID received from the merchant matches that in the PI received (indirectly) from the customer
7. Requests and receives an authorization from the issuer
8. sends authorization response back to merchant

Payment Capture:

1. merchant sends payment gateway a payment capture request
2. Gateway checks request
3. Then causes funds to be transferred to merchants account
4. Notifies merchant using capture response

SSL versus SET

SET Vs. SSL	
<u>Secure Electronic Transaction (SET)</u>	<u>Secure Socket Layer (SSL)</u>
Complex	Simple
SET is tailored to the credit card payment to the merchants.	SSL is a protocol for general-purpose secure message exchanges (encryption).
SET protocol hides the customer's credit card information from merchants, and also hides the order information to banks, to protect privacy. This scheme is called <i>dual signature</i> .	SSL protocol may use a certificate, but there is no payment gateway. So, the merchants need to receive both the ordering information and credit card information, because the capturing process should be initiated by the merchants.

SSL vs. SET

SSL

- Server authentication
 - Merchant certificate as legitimate business
- Possible for client authentication
 - Not tied to payment method
- Privacy
 - Encrypted message to merchant includes account number
- Integrity
 - Message authenticity check

SET

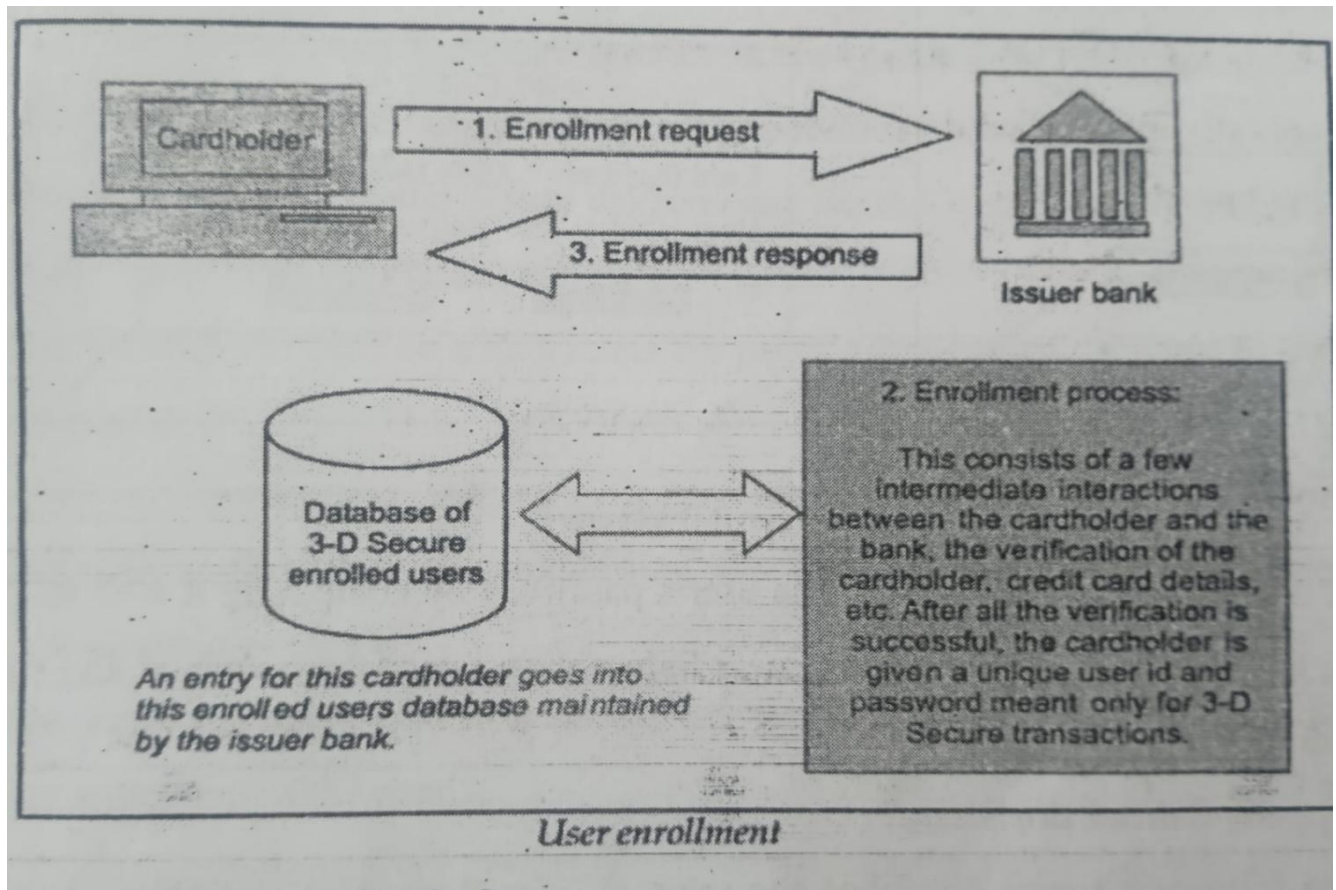
- Server authentication
 - Merchant certificate tied to accept payment brands
- Customer authentication
 - Digital certificate tied to certain payment method
- Privacy
 - Encrypted message does not pass account number to merchant
- Integrity
 - Hash/message envelope

9

3D Secure Protocol

3-D Secure is an XML (Extensible Markup Language) based protocol designed to be an additional security layer for online credit and debit card transactions. It was originally developed by Arcot Systems (now CA Technologies) and first deployed by Visa with the intention of improving the security of Internet payments, and is offered to customers under the Verified by Visa.

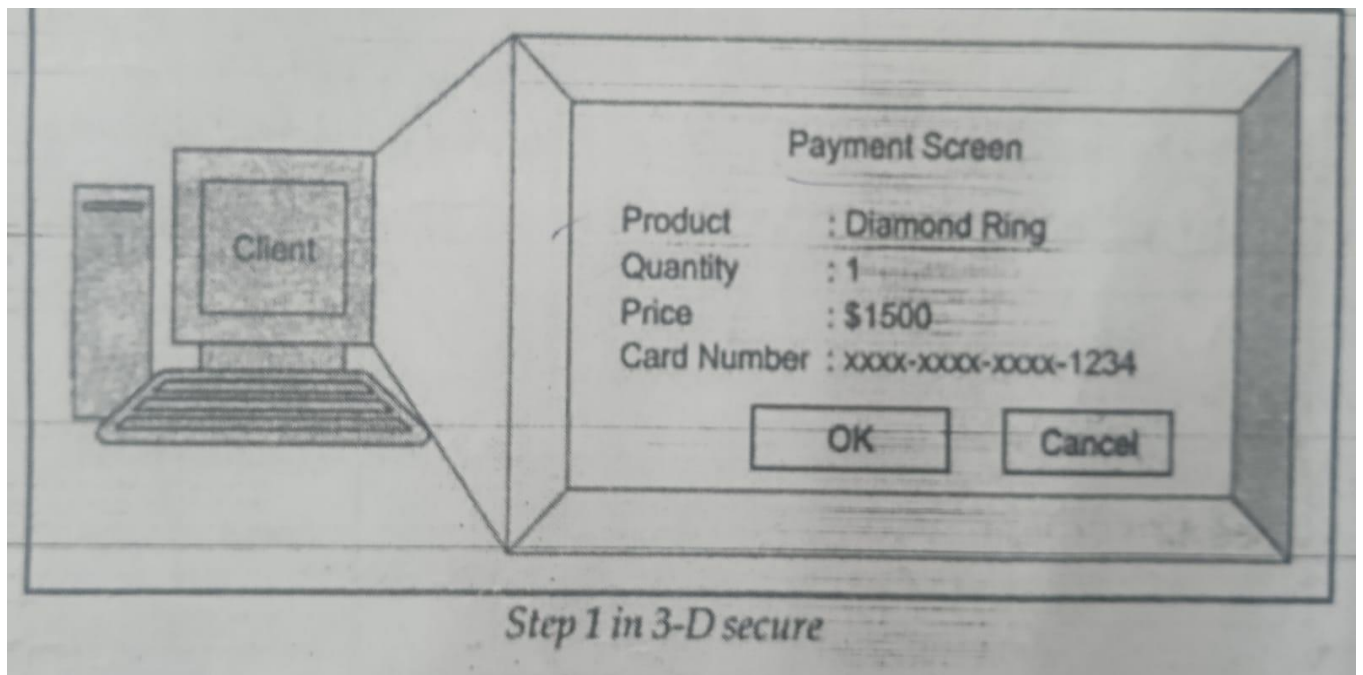
The main difference between SET and 3-D Secure is that any cardholder who wishes to participate in a payment transaction involving the use of the 3-D secure protocol has to enroll on the issuer bank's Enrollment Server. That is the cardholder makes a card payment, he/she must enroll with the issuer bank's Enrollment Server. This process is shown in the following figure:



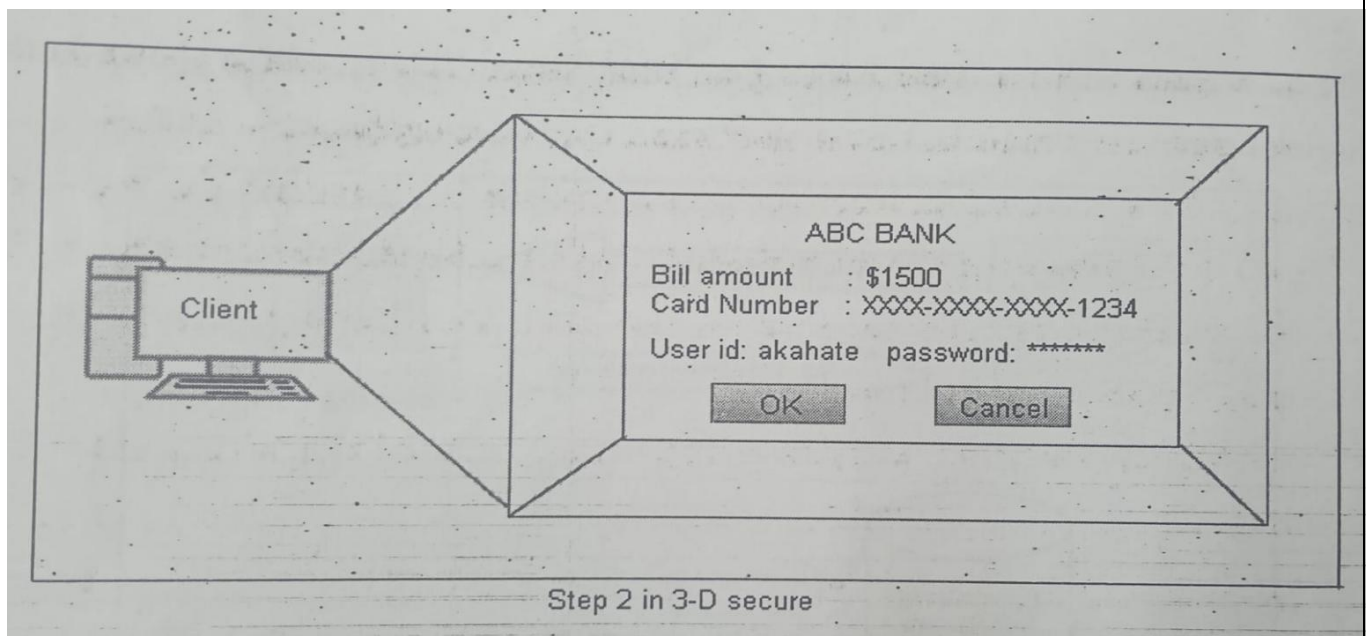
How 3-D Secure Protocol Works?

The following steps explain 3-D secure protocol works:

Step 1: The user shops using the shopping cart on the merchant site and decides to pay amount. The user enters the credit card details and clicks on OK button. This is shown in the following figure:



Step 2: The user will be redirected to issuer's bank site and enters the password given by the bank. This is shown in the following figure:



At this stage, the bank verifies the user's password by comparing it with its database entry. The bank sends an appropriate success/failure message to the merchant and shows the corresponding screen to the user.

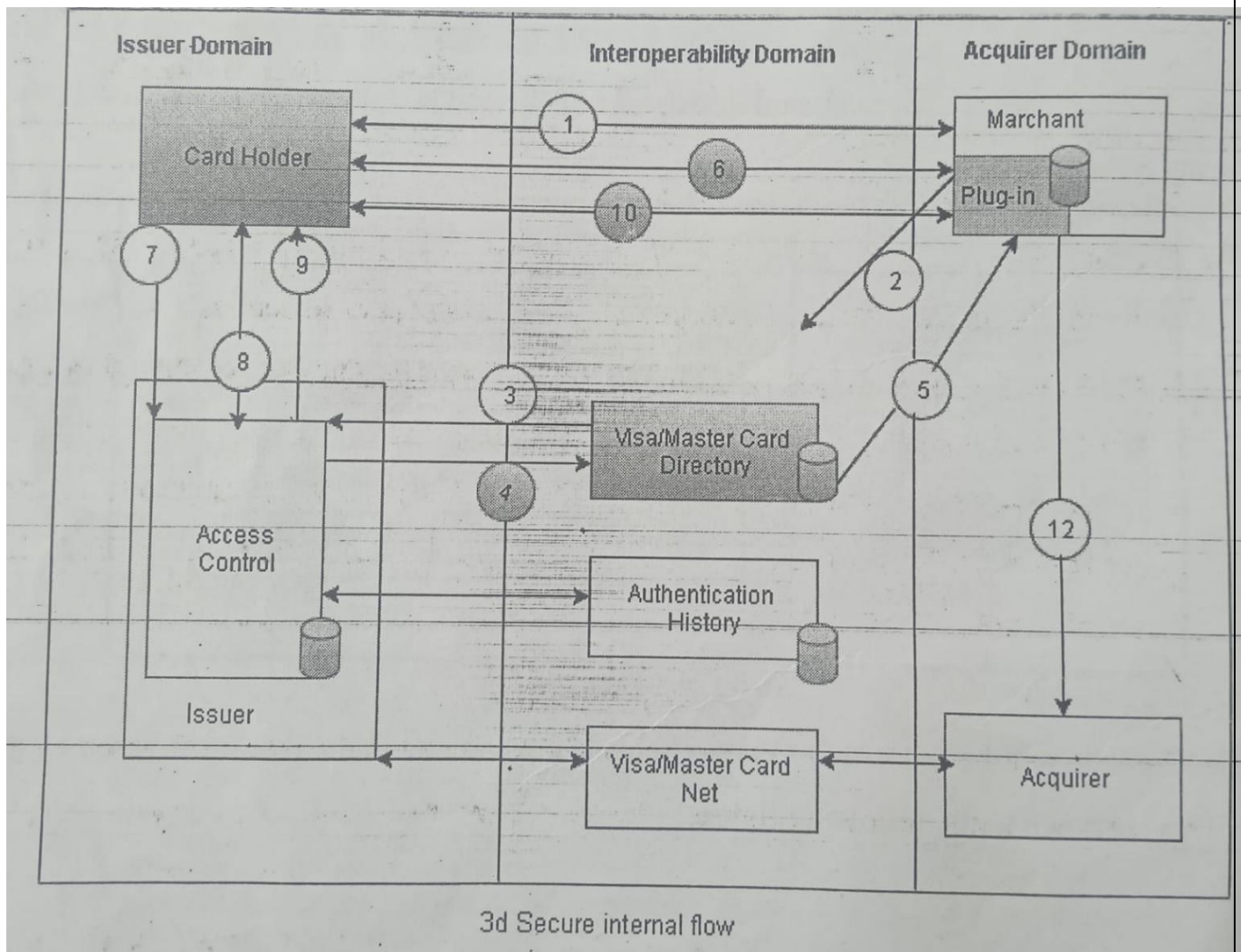
What happens behind the scene?

The basic concept of the protocol is to tie the financial authorization process with online authentication. This additional security authentication is based on a “three-domain” model (hence the 3-D in the name itself).

The three domains are:

1. Issuer domain: the bank which issued the card being used.
2. Acquirer domain: the bank and the merchant to which the money is being paid.
3. Interoperability domain: the infrastructure provided by the card scheme, credit, debit, prepaid or other types of a payment card, to support the 3-D Secure protocol. It includes the Internet, merchant plug-in, access control server, and other software providers.

The following figure describes the internal operations of 3-D Secure. The process uses SSL for confidentiality and server authentication.



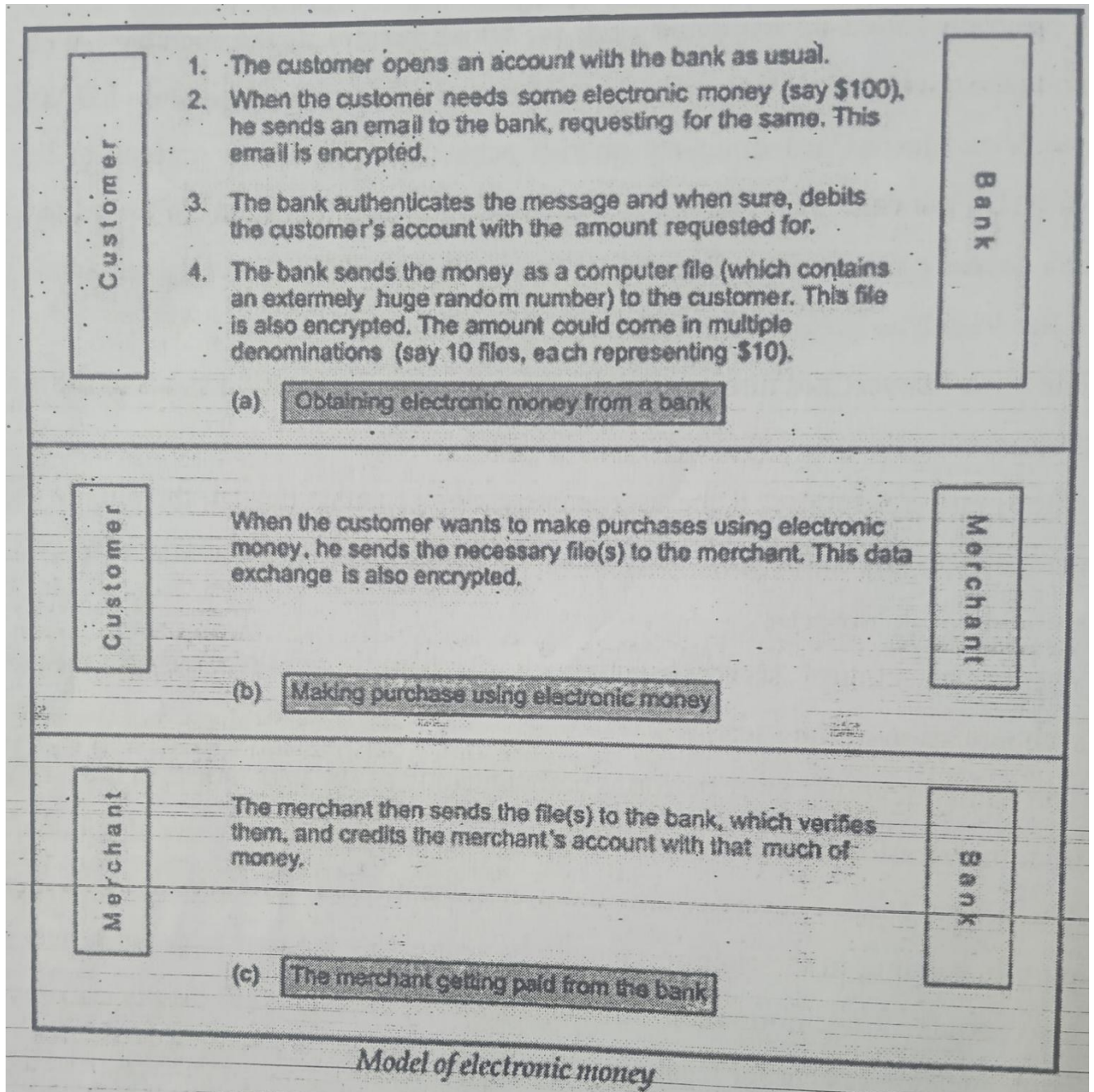
The flow can be described as follows:

1. The customer finalizes on the payment on merchant side (the merchant has all the data of this customer).
2. A program called as merchant plug in, which resides at the merchant Web server, sends the user information to the Visa/Master Card directory.
3. The Visa/MasterCard directory queries access control server running at the issuer bank (i.e. the customer's bank), to check the authentication status of the customer.
4. The access control server forms the response for the Visa directory and sends it back to the Visa/MasterCard directory.
5. The Visa/MasterCard directory sends the payer's authentication status to the merchant plug in.

6. After getting response, if the user is currently not authenticated, the plug in redirects the user to the bank site, requesting the bank or the issuer site to perform the authentication process.
7. The access control server receives the request for authentication of the user.
8. The authentication server performs authentication of the user based on the mechanism of authentication chosen by the user (e.g. password, dynamic password, mobile, etc.).
9. The access control server returns the user authentication information to the merchant plug in running in the acquirer domain by redirecting the user to the merchant site. It also sends the information to the repository where the history of the user authentication is kept for legal purpose.
10. The plug in receives the response of the access control server through the user's browser. This contains the digital signature of the access control server.
11. The plug in validates the digital signature of the response and the response from the access control server.
12. If the authentication was successful and the digital signature of the access control server is validated, the merchant sends the authentication information to its bank (I.e.the acquirer bank).

Electronic money

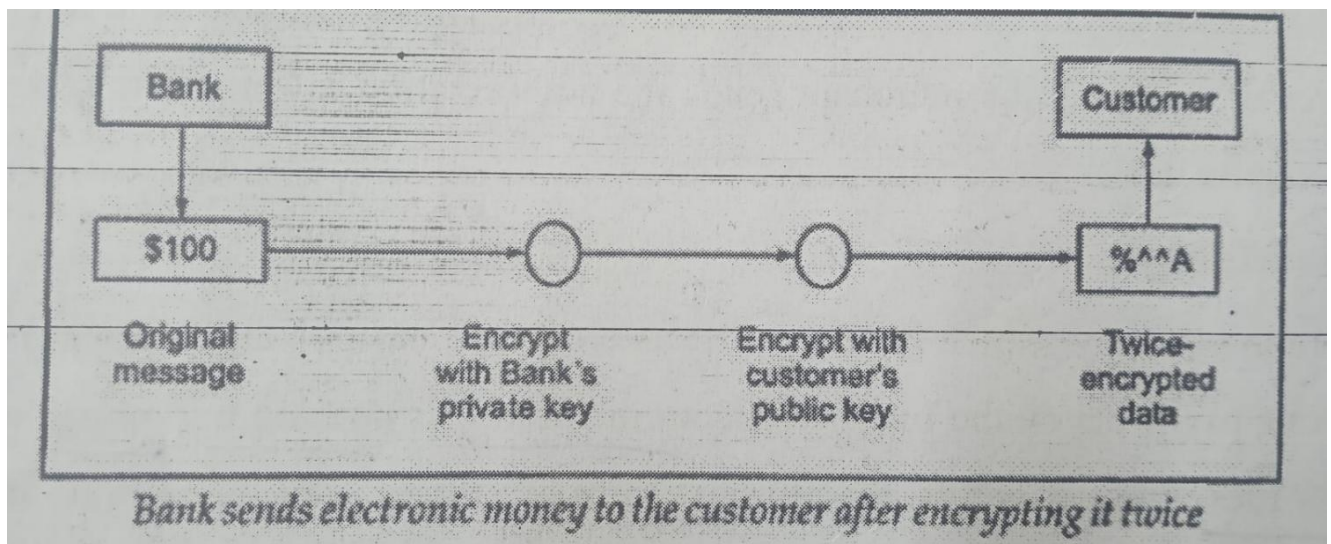
Electronic money, which is also called as electronic cash or digital cash, is one more way of making payments on the Internet. Electronic money is nothing but money represented by computer files. In other words, the physical form of money is converted into binary form of computer data. The following figure shows the conceptual steps involved in electronic money processing:



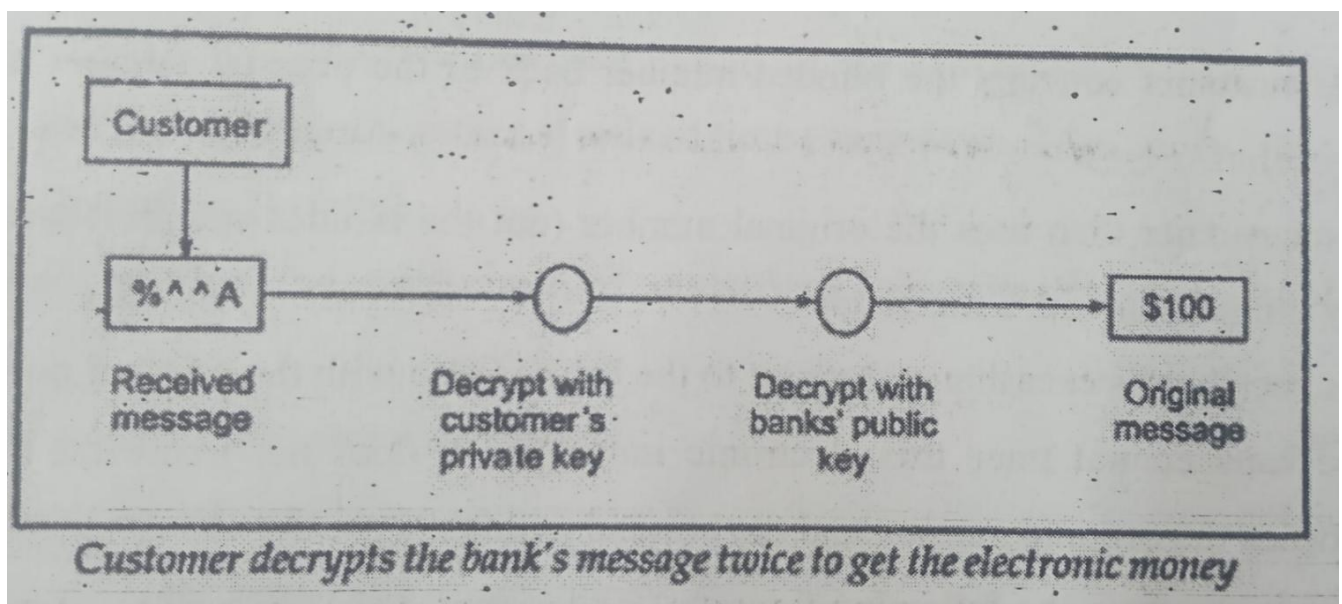
Security Mechanisms in Electronic Money:

The following steps provide security mechanisms in Electronic Money.

Step 1: Bank sends the electronic money to the customer. This is shown in the following figure:



Step 2: The customer receives the money and decrypts it. This is shown in the following figure:



Types of electronic money

E-Money virtual money stored in the banking computer systems. It doesn't have a physical form like regular paper currency. Nowadays E-money is trending because of its flexible and safe features rather than physical cash. Let's take a look at the common types of E-Money.

1. E-money Payment Networks

E-money is electronic money that can be transferred through debit cards, credit cards, computer systems and smartphones. It gives customers the possibility to make bank to bank transactions and buy goods and services online.

2. Hard Electronic Currency

Non-reversible transactions are dealt through hard electronic currency, such as those drawn through a bank. The best part about this type of e-money is its cost efficiency of operations and limited paperwork.

3. Soft Electronic currency

Reversible transactions are dealt through soft electronic currency. It includes products like UPIs and credit cards. The benefit of reversible transactions include that users can take back a transaction or cancel the transaction within a defined period of 72 hours.

4. E-money Delivery Systems

E-money can be conveniently stored on your phone, computer, a USB card(in code) or a smart money card. Credit and debit cards are a form of digital money. You can make use of any of the payment methods to transfer money easily.

5. Identified and Unidentified or Anonymous E-money

Identified E-money is a form of e-money that allows the user who withdraws the money to be tracked such as credit/debit card transactions. Banks can easily track your payments. Unidentified money is usage of physical cash which can be withdrawn from the bank and used anywhere.

6. Online and Offline E-transactions

For online e-transaction one does require a proper internet connection as it conducts transactions from bank to the third parties. But an offline e-transaction does not require internet connection as it does not involve a bank and the e-money is stored on a card, chip or other media and can be used by anyone.

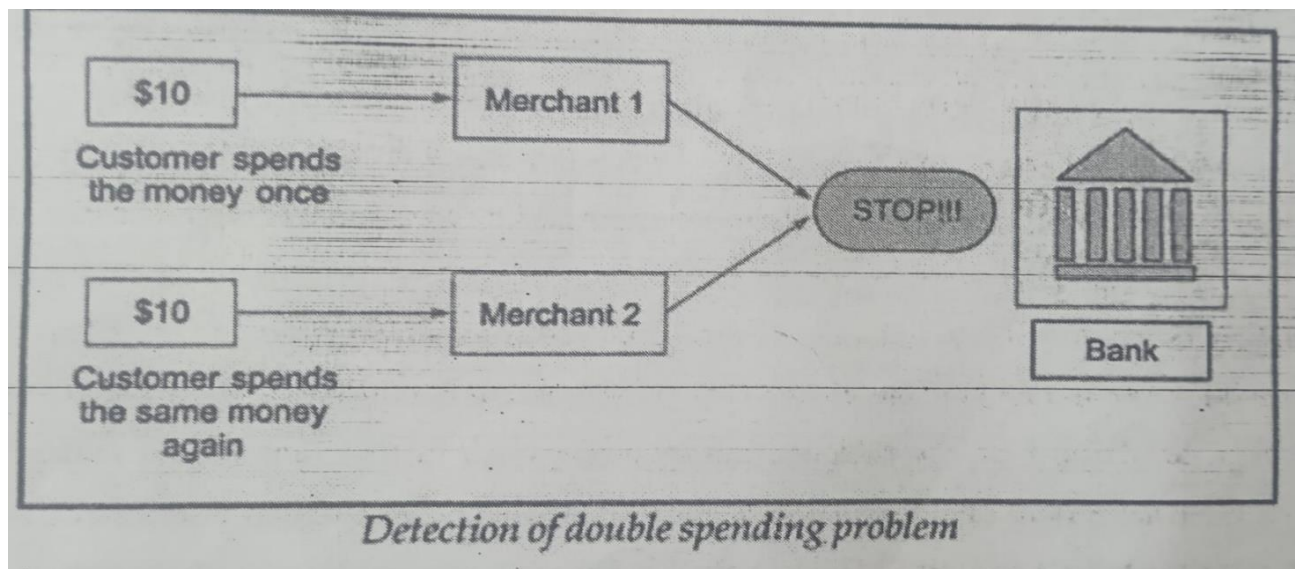
The Double Spending Problem:

If we combine the two ways of electronic money, we have four possibilities:

1. Identified online electronic money
2. Identified offline electronic money
3. Anonymous online electronic money
4. Anonymous offline electronic money

Of the four, the last type can create the double spending problem. A customer could arrange for anonymous electronic money by using the blinded money concept. Later on, he could spend it offline more than once in quick succession with to different merchants. Since the bank is not involved in any of the two online transactions, the fact that same price of money is being spent cannot be prevent. Moreover, when it is realized that the same piece of money is spent more than once, the bank cannot determine which customer spent it more than once, because of the blinding factor. Consequently, anonymous offline electronic money is of little practical use.

Double spending problem can happen in case of identified offline electronic money as well. However, upon detection, the customer under question can be easily tracked from the serial numbers of the electronic money. This is shown in the following figure:

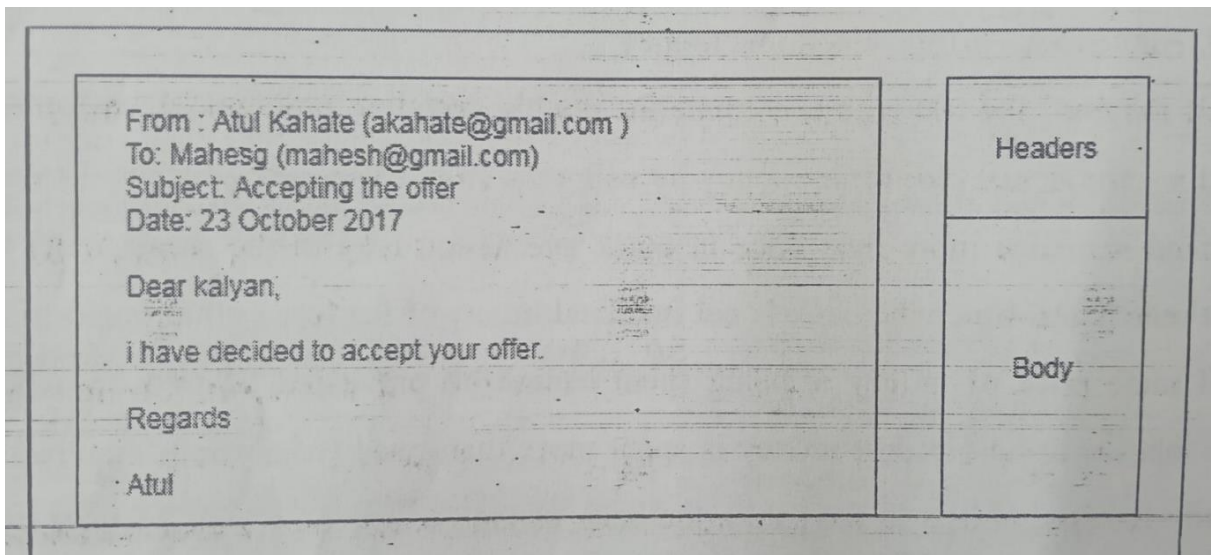


However, this detection is not possible in case of anonymous offline electronic money. Double speaking problem is not possible either of the online transactions because the bank is a part of transaction between the customer and the merchant.

Email Security

Electronic mail (email) is perhaps the most widely used application on the Internet. Email users can send and receive data easily on the Internet. Consequently, the security of email messages has become an extremely important issue. RFC 822 (Request for Comments) defines a format for test email messages. An email consists two parts: Headers and Body (contents). A header line consists of keyword, followed by a colon, followed by the keyword's arguments. Examples of header keywords are From, To, Subject and Date. The following figure distinguishes between its headers and contents.

Email Header and Body



Email Security Protocols:

The following main three types of security protocols are providing security to the emails

1. Privacy Enhanced Mail (PEM)
2. Pretty Good Privacy (PGP)
3. Secure and Multipurpose Internet Mail Extension (S/MIME)

PGP

PGP (Pretty Good Privacy), is high security cryptographic software applications, which allows people to exchange messages or file with privacy (confidentiality), authentication and integrity. PGP can also be used to encrypt and apply digital signature for e-mail, PGP was developed by Zimmermann in the 1980s and first version was released on Internet in 1991. Because of legal issues for usage of RSA, it was purchased by Via-Crypt and RSA licensed company in 1993 and released again in 1994.

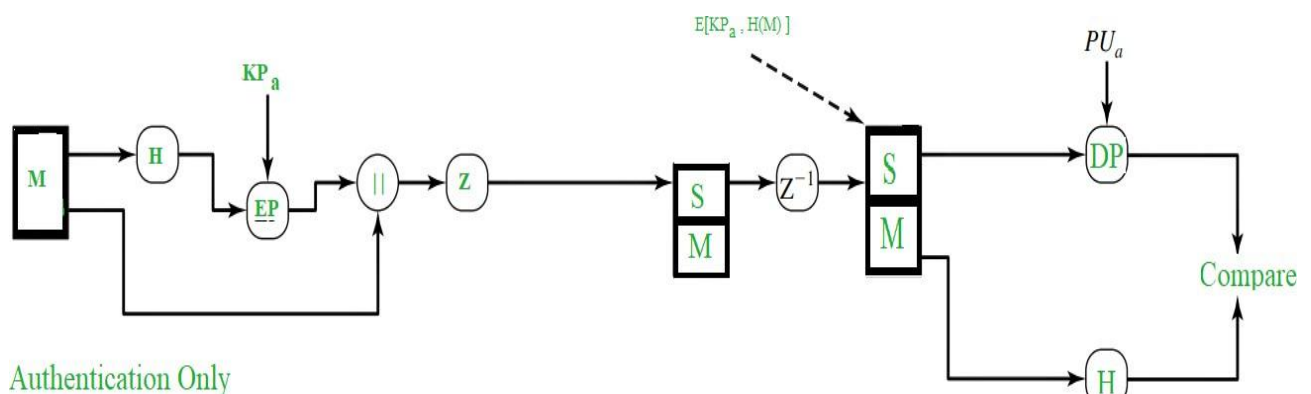
There are a number of reasons that make the PGP to use widely Some of them are

1. PGP is freeware, but also commercial version available.
2. Operating system independent and run on Windows/Unix/Macintosh etc.
3. Used popular and stranded algorithms like RSA, DSS, IDEA
4. Wide range of applications.
5. One of the major reasons is it was not controlled by any governmental or standard organizations.

The following are the services offered by PGP:

1. Authentication (Signature/Verification)
2. Confidentiality (Encryption/Decryption)
3. Compression (ZIP)
4. Email Compatibility
5. Segmentation and Reassembly

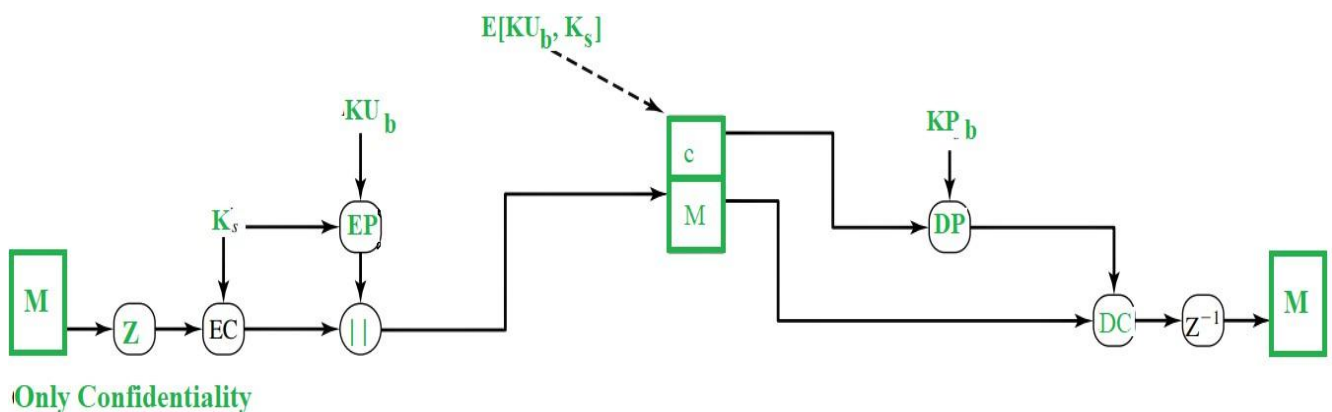
1. Authentication: The following diagram shows the authentication provision by using digital structure :



The above set scheme is called a digital signature scheme Steps involved in the above agreement explained as follows:

1. At source machine messages created
2. Generation of 160-bit hash code of message by using hash SHA-1.
3. RSA algorithm is used for encrypting hash code; sender's private key is used for encryption so that authentication is provided. This hash code is used to the message.
4. Receiver decrypts the message by using public key of sender and recovers the hash code.
5. Now receiver generates hash code for the message and compares it with the decrypted hash code. If both are same, then the message is authenticated.

2. Confidentiality: PGP provides basic service confidentiality. The following diagram shows how confidentiality is provided :

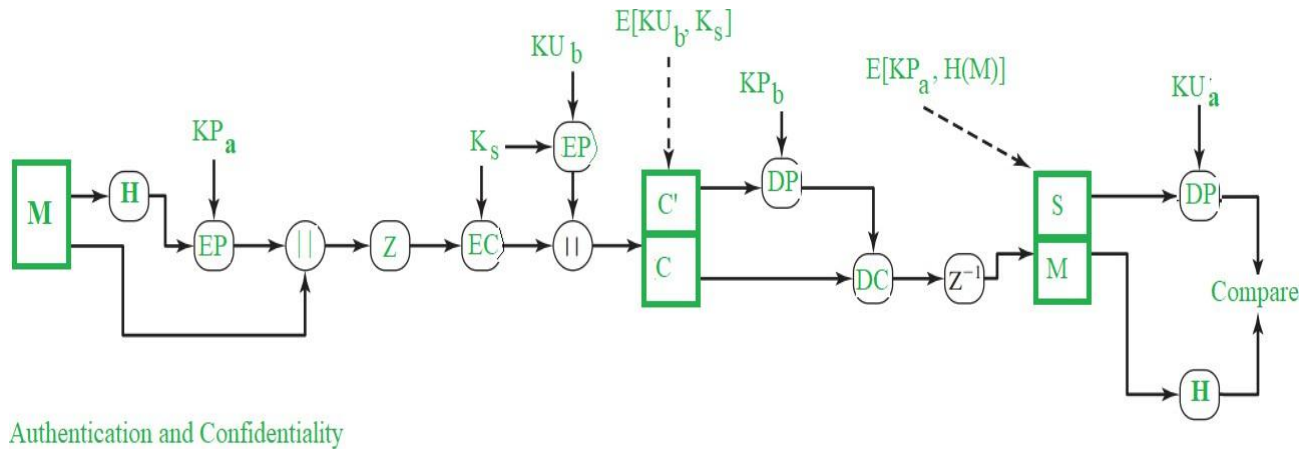


The process of confidentiality is described in the following steps

1. At source machine messages is created and a random 128-bit number is used as a session key.
2. By using symmetric encryption algorithm, CAST-128 or IDEA or triple DES message is encrypted with the session key.
3. Confidentiality step taken place know, session key is encrypted with RSA by using the destination public key, and is added to the message.
4. Receiver decrypts the message by using private key of destination and recovers the session key.
5. Now receiver uses this session key to decrypt the message.

Confidentiality and Authentication:

To increase the trust of any service, we should provide both confidentiality and authentication. The following diagram shows both services:



Note:

- M – Message
- H – Hash Function
- K_s – A random Session Key created for Symmetric Encryption purpose
- DP – Public-Key Decryption Algorithm
- EP – Public-Key Encryption Algorithm
- DC – Asymmetric Encryption Algorithm
- EC – Symmetric Encryption Algorithm
- KP_b – A private key of user B used in Public-key encryption process
- KP_a – A private key of user A used in Public-key encryption process
- PU_a – A private key of user A used in Public-key encryption process
- KU_b – A private key of user B used in Public-key encryption process
- || – Concatenation
- Z – Compression Function
- Z⁻¹ – Decompression Function

The above diagram the following steps are taken place.

1. At source machine message is created.
2. Generation of 160-bit hash code of the message by using SHA-1.
3. RSA algorithm is used for encrypting hash code; sender's, private key is used for encryption so that authentication is provided. This hash code is used to the message.
4. A random 128-bit number is used as a session key.
5. By using symmetric encryption algorithm CAST-128 or IDEA or Triple DES signed message is encrypted with the session key.
6. Confidentiality step taken place know, session key is encrypted with RSA by using the destination public key, and is added to the message.
7. Receiver decrypts the signed message by using private key of destination and recovers the session key.
8. Now receiver uses this session key to decrypt the message.
9. Receiver decrypts the message by using public key of sender and recovers the hash code.
10. Now receiver generates hash code for the message and compares it with the decrypted hash code, if both are same then the message is authenticated.

3. Compression

PGP compress the message after applying the signature but before encryption. The main reason behind it is:

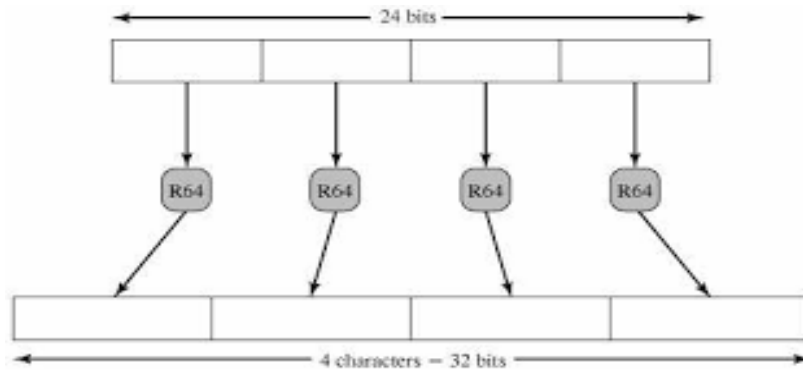
1. Uncompressed message signing is better for the verification.
2. Recompressed message verification is bit default because PGP's compression algorithm. It uses non-deterministic algorithm.

Message encryption is applied after compression (that reduces redundancy) automatically strengthens cryptographic security. Compression algorithm ZIP is used for compressing message in PGP.

The placement of the compression algorithm, indicated by Z for compression and Z-1 for decompression.

4. E-mail compatibility

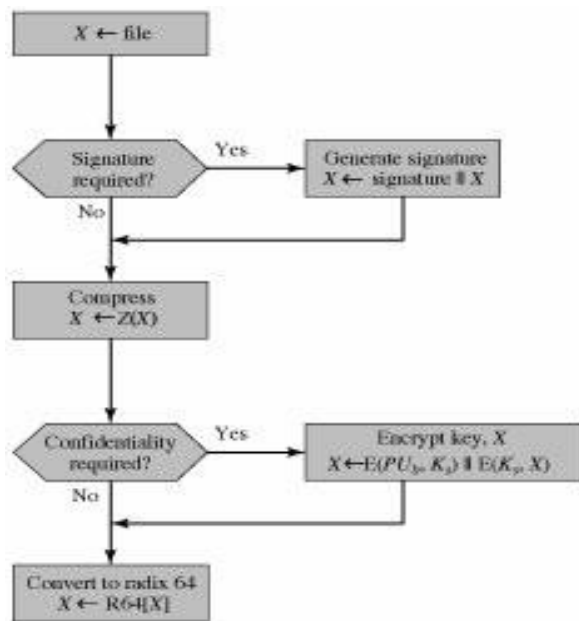
Radix-64 encoding conversion is E-mail compatible conversion. Each group of three octets of binary data (24 bits) is mapped into four ASCII characters (32 bits). It also adds cyclic redundancy check (CRC), useful for finding transmission errors. The following diagram shows this conversion process.



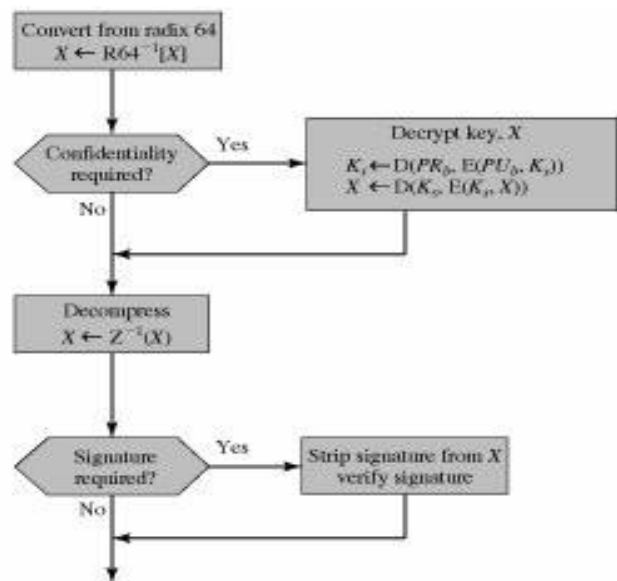
5. Segmentation and Reassembly

E-mail systems impose maximum length on transmission. Some systems made this limit as 50 kb. For that reason PGP provides an automatic process that divides the message into small segments and this process is called segmentation. This was done after all other operations are over. Before transmitting the message, segmentation takes place. At the receiving end reassembly of these segmented packets automatically done by PGP.

The following diagrams show all security services taken place at source and destination except segmentation and reassembly.



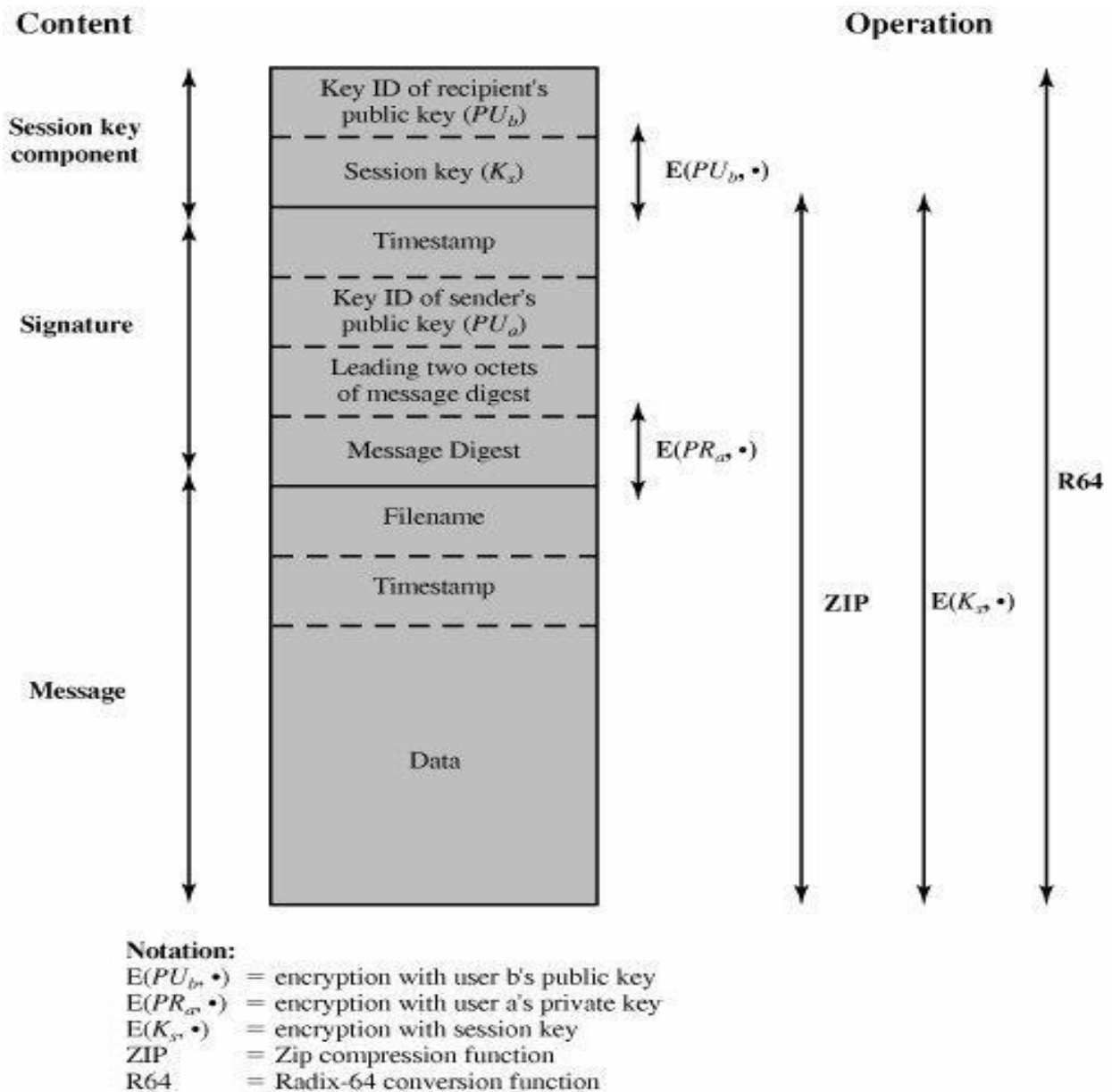
(a) Generic transmission diagram (from A)



(b) Generic reception diagram (to B)

General Format of PGP:

Figure: General format of PGP message (from A to B).



Key Management of PGP:

PGP uses the following four types of keys for its transmission.

1. One-time Session Symmetric Keys.
2. Public keys (of Public key cryptography)
3. Private keys (of public key cryptography)
4. Pass phrase-based symmetric keys.

Key Rings:

To overcome the difficulty of key identifications with every message PGP provides two types of key rings, one is private key ring, and other one is public key ring. Private Key ring includes Timestamp, Key ID, public key, encrypted private key and user identification. Public key ring includes Timestamp, key id, public key, owner trust, user id, key legitimacy, signature and signature trust. Purpose of each field is as follows:

General Structure of Private and Public Key Rings

Private-Key Ring

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Public-Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
T_i	$PU_i \bmod 2^{64}$	PU_i	trust_flag _{i}	User i	trust_flag _{i}		
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

* = field used to index table

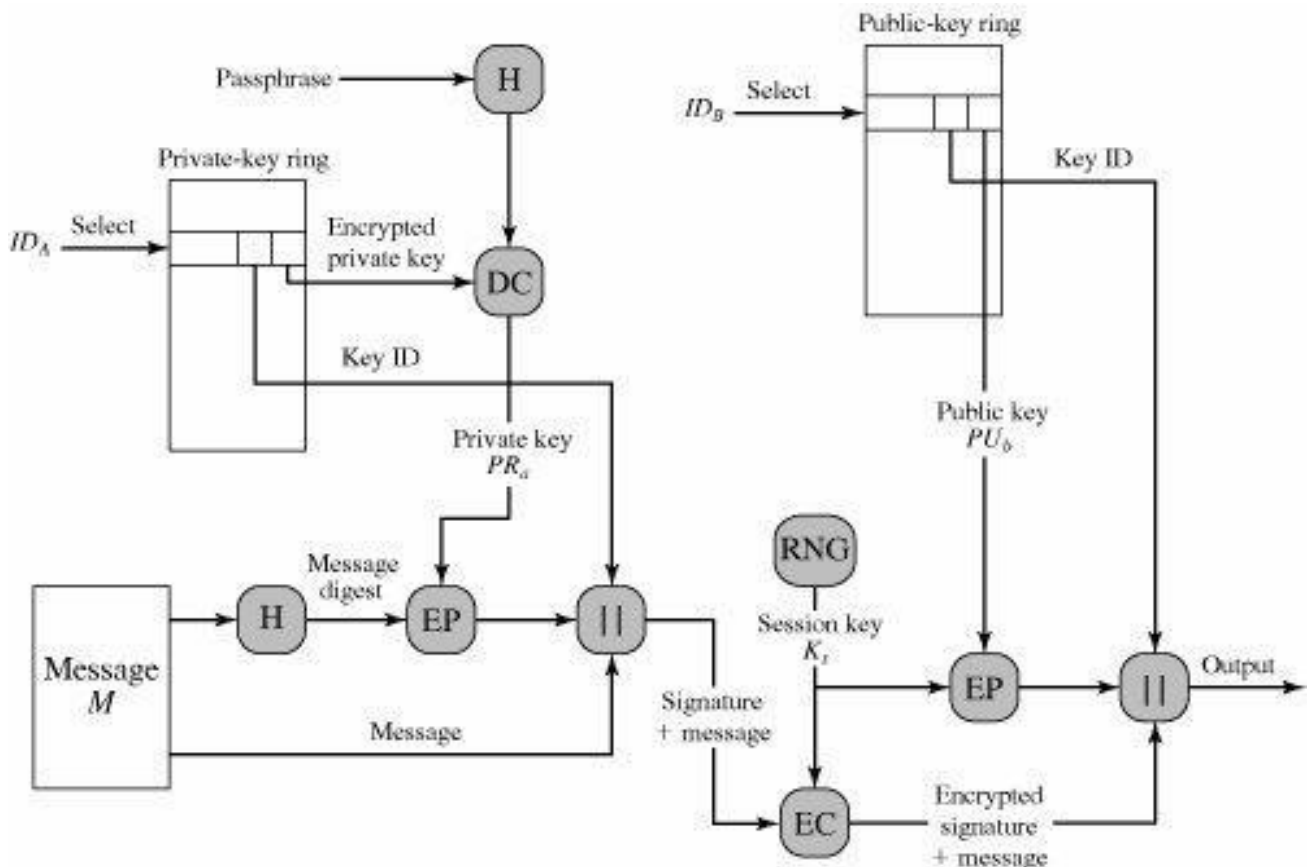
PGP Message Generation:

Sending steps for the PGP entity are as follows:

1. Signing the message
 - a. PGP retrieves the sender's private key from the private-key ring using your_userid as an index. If your_userid was not provided in the command, the first private key on the ring is retrieved.
 - b. PGP prompts the user for the passphrase to recover the unencrypted private key.
 - c. The signature component of the message is constructed.

2. Encrypting the message

- a. PGP generates a session key and encrypts the message.
- b. PGP retrieves the recipient's public key from the public-key ring using her_userid as an index.
- c. The session key component of the message is constructed.



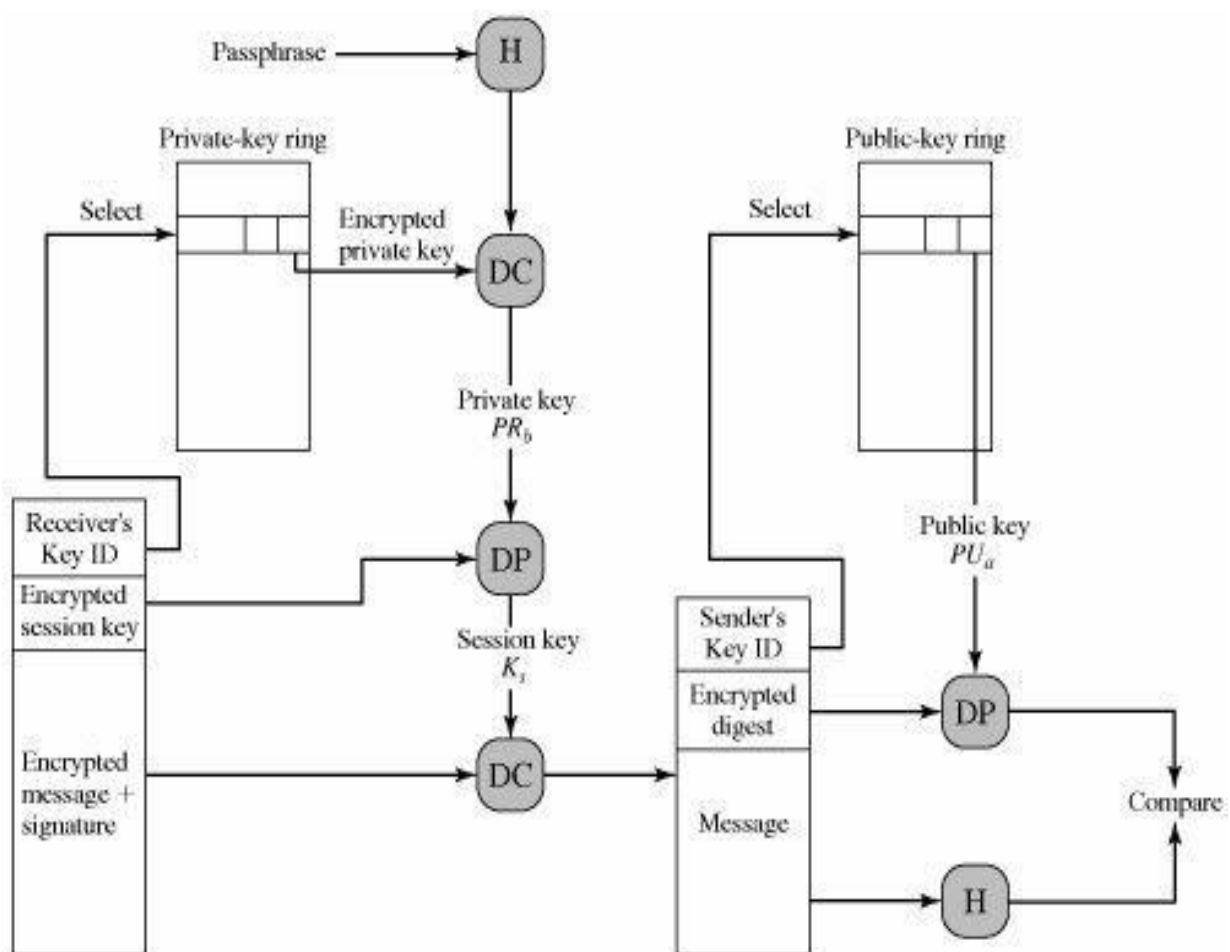
Receiving steps for the PGP entity are as follows:

1. Decrypting the message.

- a. PGP retrieves the receiver's private key from the private-key ring, using the Key ID field in the session key component of the message as an index.
- b. PGP prompts the user for the passphrase to recover the unencrypted private key.
- c. PGP then recovers the session key and decrypts the message.

2. Authenticating (Verifying) the message

- a. PGP retrieves the sender's public key from the public-key ring, using the Key ID field in the signature key component of the message as an index.
- b. PGP recovers the transmitted message digest.
- c. PGP computes the message digest for the received message and compares it to the transmitted message digest to authenticate.



S/MIME

S/MIME stands for **Secure/Multipurpose Internet Mail Extensions**; it is security enhance for S/ MIME Internet email standard. Actual Internet RFC 822 email can be able to transfer only text content only. MIME provides facility for transferring of varying content types and multi part messages. It used encoding of binary data to textual form. S/MIME has support in various modern mail agents like MS Outlook, Netscape etc.

Multipurpose Internet Mail Extension (MIME):

Mainly MIME was developed to overcome the problems and limitations of the use of SMTP (Simple Mail Transfer Protocol), Some of the limitations and problems of SMTP and some of them addressed by MIME are:

SMTP Cannot:

1. Transmit Executable files.
2. Transmit text data that contains Unicode characters or national language characters.
3. Transfer over a size limit.
4. Handle non textual data included in X, 400 messages

Common Problems are:

1. Handling of carriage return and line feed characters.
2. Wrapping of lines.
3. Removal of trailing white space.
4. Padding of lines in a message to the same length.
5. Conversion of tabs into space characters.

Overview of MIME:

MIME specifications include five new messages that provide information about the body of the message, a number of content formats and transfer encoding techniques. The five header fields are MIME-Version, Content Type, Content Transfer Encoding technique, Content ID and Content Description. Purpose of each field is as follows :

1. **MIME version:** Must have the value 1.0 indication of proper RFC.
2. **Content-Type:** Type of the content that is given in body is given here.
3. **Content-Transfer-Encoding:** Type of transformation that is used to represent the body of the message in a way that is acceptable for mail transport.
4. **Content-ID:** It is used for identification of MIME entities.
5. **Content-Description:** Description of the object with the body. When you have audio kind of thing then this description is useful to know body content in detail.

MIME Content Types:

There are seven content types along with some sub types of them. The following table gives complete content types of MIME.

Type	Sub Type	Description
Text	Plain	Unformatted text
	Enriched	Formatted, rich text
Multipart	Mixed	Combination of different parts , but has some order
	Parallel	Same as Mixed but without any order
	Alternative	Different parts are alternative versions of the same
	Digest	Same as Mixed, but format varieties like msg/rfc
Message	Rfc 822	Body is itself an encapsulated message that confirms to
	Partial	Fragmentation of large items
	External-body	A pointer reference to an external object
Image	Jpeg	JPEG format
	Gif	GIF format
Video	Mpeg	MPEG format
Audio	Basic	Single channel encoding technique
Application	Post Script	Adobe Post script format
	Octet-Stream	Normal binary data that consists of 8-bit bytes.

Generally content type declares the general type of data, and the subtype is used to specify a particular format for that type of data. There are two subtypes for the text type. One is plain and other one is enriched. Plain text is a string of ASCII the characters, where as enriched text allows greater flexibility regarding format.

MIME Transfer Encodings:

Other major component of the MIME specification is definition of transfer encoding techniques. There are six different MIME transfer encoding techniques. The following table represents the MIME transfer encoding techniques.

Encoding	Description	Usage
7 bit	Short lines of ASCII	SMTP message transfer
8 bit	Short, but can have non	Other mail transfer context
Binary	ASCII, non-ASCII and not	Other mail transfer context
Quoted-Printable	Human understandable form	Introduces non-safe characters
Base 64	Mapping of 6 bit blocks to 8	Used in PGP
x-token	Named nonstandard encoding	Vendor specific or application

Functionality of S/MIME:

In terms of general functionality, S/MIME is very similar to PGP. Both offer the ability to sign and/or encrypt messages. In this subsection, we briefly summarize S/MIME capability. We then look in more detail at this capability by examining message formats and message preparation.

Functions: S/MIME provides the following functions:

- **Enveloped data:** This consists of encrypted content of any type and encrypted-content encryption keys for one or more recipients.
- **Signed data:** A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.
- **Clear-signed data:** As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using

base64. As a result, recipients without S/ MIME capability can view the message content, although they cannot verify the signature.

- **Signed and enveloped data:** Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

S/MIME Example:

```
From: attacker@efail.de
To: victim@company.com
Content-Type: multipart/mixed;boundary="BOUNDARY"

--BOUNDARY
Content-Type: text/html


--BOUNDARY--
```

Cryptographic algorithms for S/MIME:

Service/Function	MUST	SHOULD
Message digest used in	SHA-1	SHA-1
Encrypted digital signatures	DSS	RSA
Session key encryption.	Deffie-Hellman	RSA
Message encryption	Decryption Triple DES	Encryption with Triple

In the above table second and third columns refers..

1. MUST indicate an absolute requirement of the specification.
2. SHOULD indicate not compulsory requirement but recommended..

S/MIME Message:

As we have seen in the beginning of MIME, Secure MINE also supports all content types. In addition to that SYMIME provides some more content types. Some of the types are as follows:

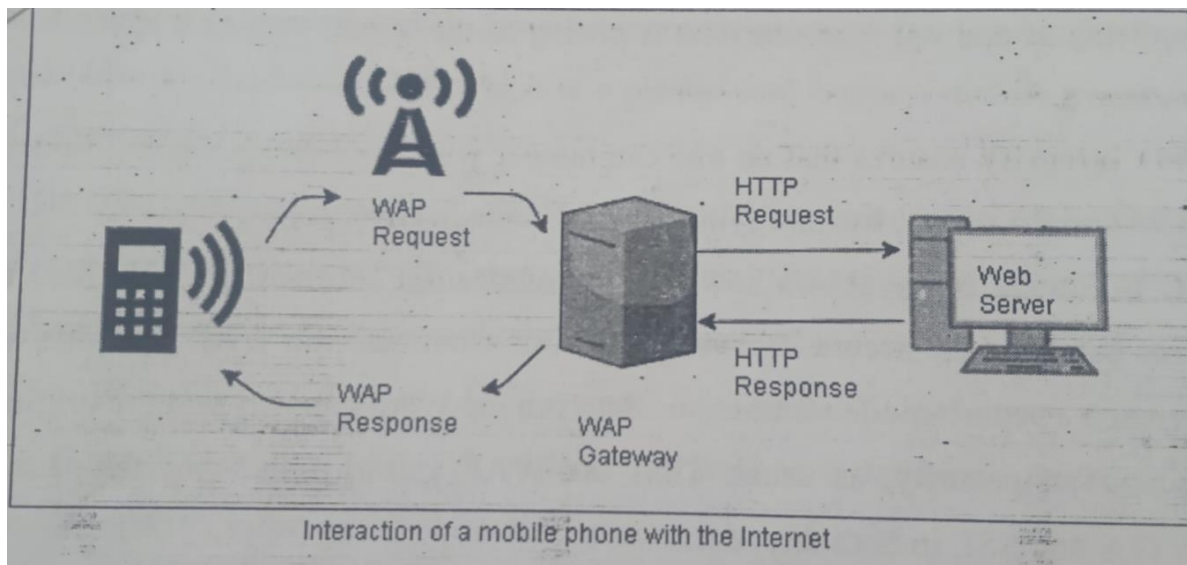
Type.	Subtype	S/MIME parameter	Description
Multi path	Signed		Clear Signed message
Application	Pkcs 7-mime	Signed Data	Signed S/MIME data
	Pkcs 7-mime	Enveloped Data	Encrypted S/MIME Data
	Pkcs 7-mime	Degenerate Signed Data	Contains only public key
	Pkcs 7-mime		Content type of signature
	Pkcs 10-mime		Certificate registration

WAP Security

Wireless Application Protocol (WAP) Commonly known as WAP is used to enable the access of internet in the mobile phones or PDAs. The main purpose of WAP is to enable easy, fast delivery of relevant information and services to mobile users. The important information of WAP is described as the following:

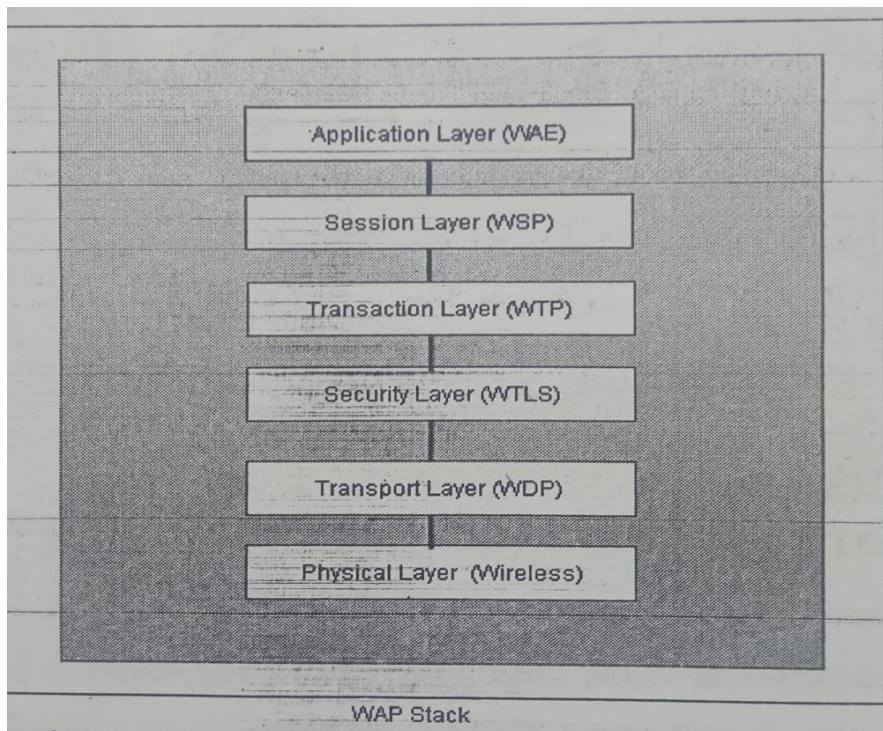
1. WAP is an application communication protocol
2. WAP is used to access services and information
3. WAP is for handheld devices such as mobile phones
4. WAP enables the creating of web applications for mobile devices
5. WAP uses the Wireless Mark-up Language WML (not HTML) WML is defined as an XML 1.0 application.

The WAP architecture has an additional level between the client and the server is **WAP gateway**. The WAP gateway translates client requests to the server from WAP to HTTP and way back from the server to the client, from HATTP to WAP. This is shown in the following figure:



The WAP Stack:

The WAP Stack is based more on the OSI model, rather than the TCP/IP model. The structure of WAP stack is shown in the following figure:

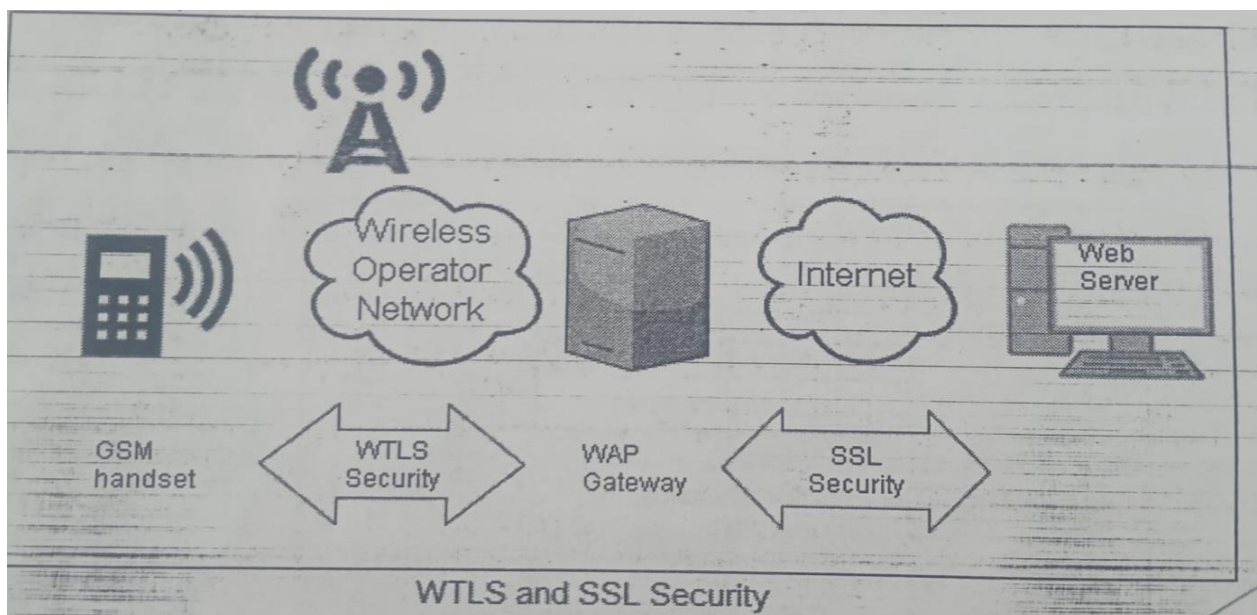


The security layer of WAP stack is also called Wireless Transport Layer Security (WTLS) protocol. It is an optional layer, that when present, provides features such as authentication, privacy and secure connections - as required by

many e-commerce and m-commerce (Mobile Commerce) applications. WTLS ensures the following four things:

1. **Privacy** ensures that the message passing between the client and the server are not accessible to anybody else.
2. **Server authentication** gives the client a confidence that the server is indeed what it is depicting as, and not someone who is posing as the server, with or without malicious intentions.
3. **Client authentication** gives the server a confidence that the client is indeed what is depicting as and not someone who is posing as the client, with or without malicious intentions.
4. **Data integrity** ensures that no one can tamper with the messages going between the client and the server, by modifying their contents in any manner.

The following figure shows how the communication between a WAP client and the origin server can be made secure. Between the WAP client and the WAP gateway, we have WTLS to ensure a secure-mode transaction. Between the WAP gateway and the origin server, SSL takes care of security, as usual. Thus the WAP gateway performs the transactions between WTLS and SSL in both directions.



Security in GSM (Global System for Mobile Communication):

GSM is the most secured cellular telecommunications system available today. GSM has its security methods standardized. GSM maintains end-to-end security by retaining the confidentiality of calls and anonymity of the GSM subscriber. There are three key aspects to GSM security:

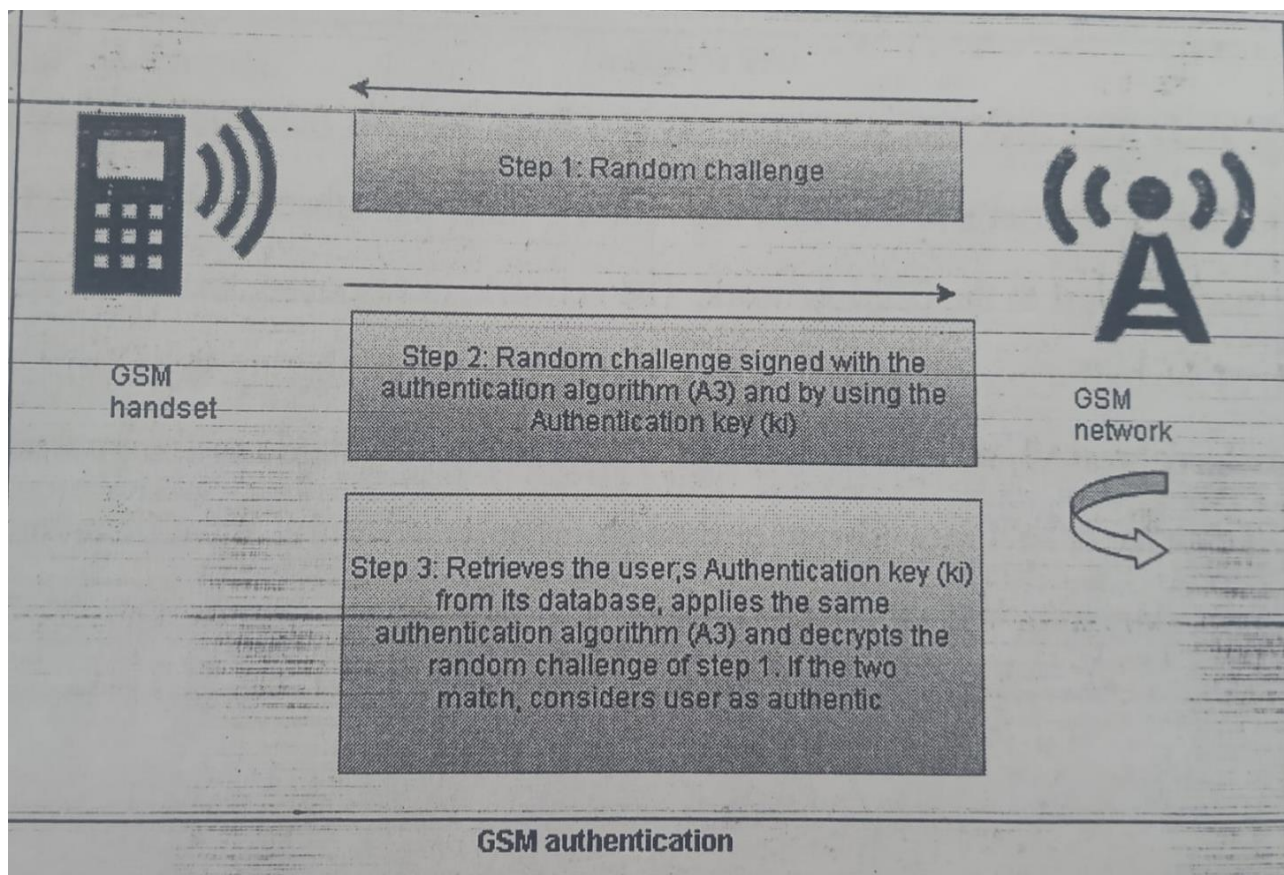
1. Subscriber identity authentication
2. Signaling data confidentiality
3. User Data confidentiality

Each subscriber is identified with a unique International Mobile Subscriber Identity (IMSI). Each subscriber also has a unique subscriber authentication key (Ki). GSM authentication and encryption work in such a way that this sensitive information is never transmitted across the mobile network. Instead, a challenge-response mechanism is used to perform authentication. The actual transmissions are encrypted with help of temporary, randomly generated ciphering key (Kc).

The security is distributed in three different elements of the GSM infrastructure: the Subscriber Identity Module (SIM), which is a plastic card inside a mobile phone, the GSM handset and the GSM network.

1. The SIM contains the IMSI, Ki, the ciphering key generation algorithm (A8), the authentication algorithm (A3) as well as Personal Identification Number (PIN).
2. The GSM handset contains ciphering algorithm (A5).
3. The Authentication Center (AUC), which is a part of GSM network, contains the encryption algorithm (A3, A5, and A8) as well as a database of identification and authentication information about the subscribers.

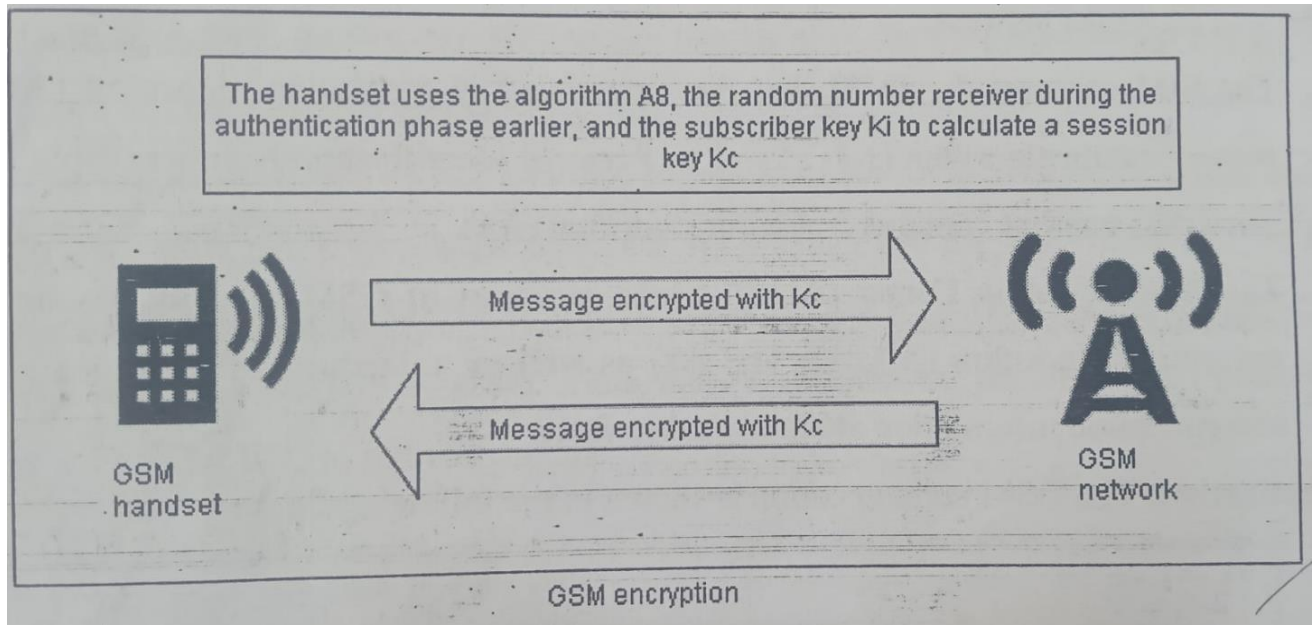
Authentication: The GSM authentication is shown in the following figure.



The process begins with a challenge-response mechanism. The network sends a 128-bit random number to the subscriber when authentication begins. After this, 32-bit signed response using the authentication algorithm (A3) and the subscriber authentication key (Ki) is prepared by the handset and sent back to the network. The network retrieves its value of Ki from its database, performs the same operation using the A3 algorithm on the original 128-bit random number and compares this result with the one received from the handset. If the two match, the user is considered as successfully authenticated. Since the calculation of the signed response takes place inside the SIM, the IMSI or Ki never have to leave the SIM. That makes authentication secure.

Signaling and Data Confidentiality: The SIM contains the ciphering key generation algorithm (A8). This is used to produce the 64-bit ciphering key (Kc). The value of Kc is obtained by applying the same random number as used in authentication to the A8 algorithm with the individual subscriber authentication key (Ki). This key (Kc) is later used for secure communications between the subscriber

and the mobile telephony base station. This process is shown in the following figure:



Voice and Data Security: The A5 algorithm is used to encrypt the voice and data traffic between the user's handset and the GSM network. For this, the subscriber's handset sends a ciphering mode request to the GSM network. The network, in response, starts encryption and decryption of the traffic using ciphering algorithm (A5) and the ciphering key (K_c).

Note: The algorithms A3, A5, A8 are kept secret and are not available to the general public. However, they have been discovered, published on the Internet, and their implementations in C and other programming languages are available in many books/resources.

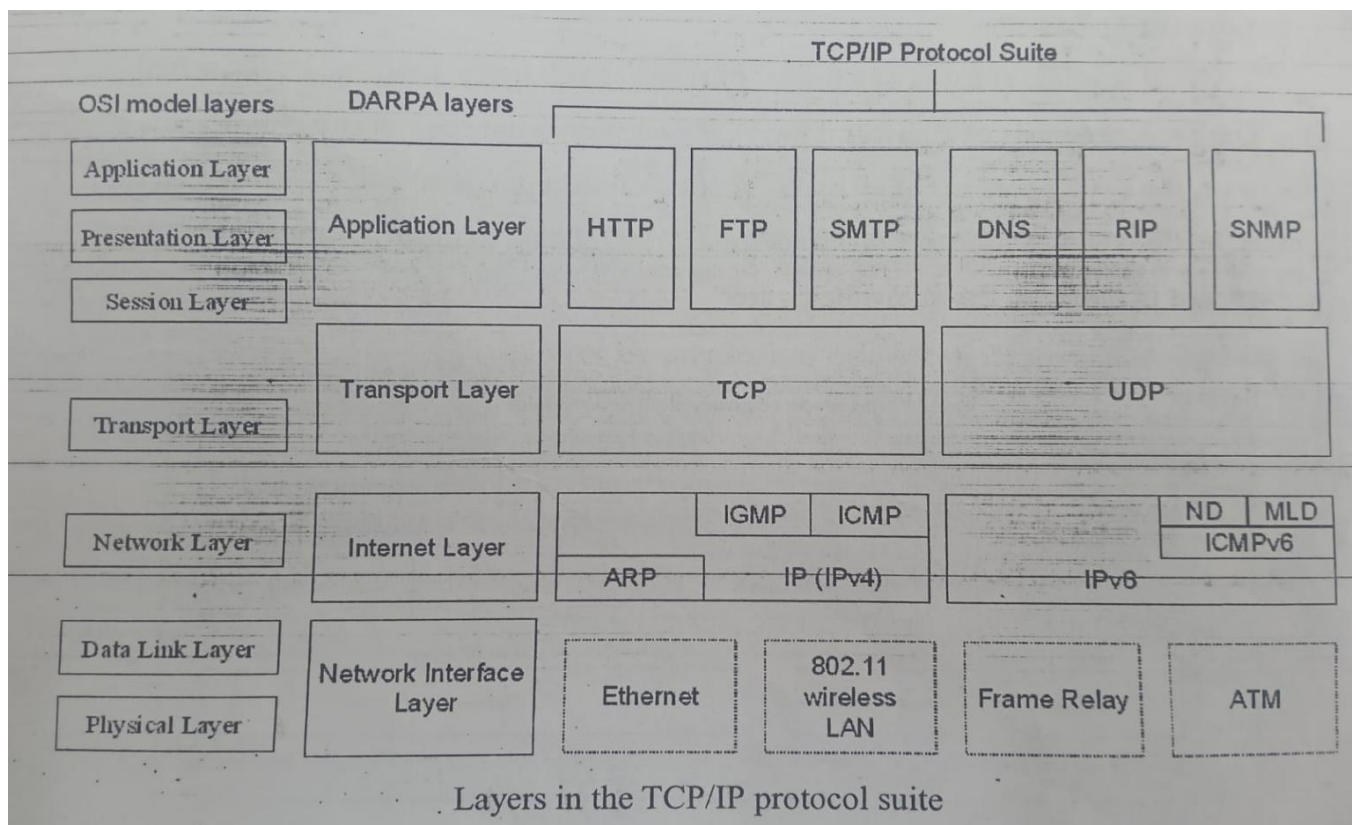
Network Security

Introduction to TCP/IP

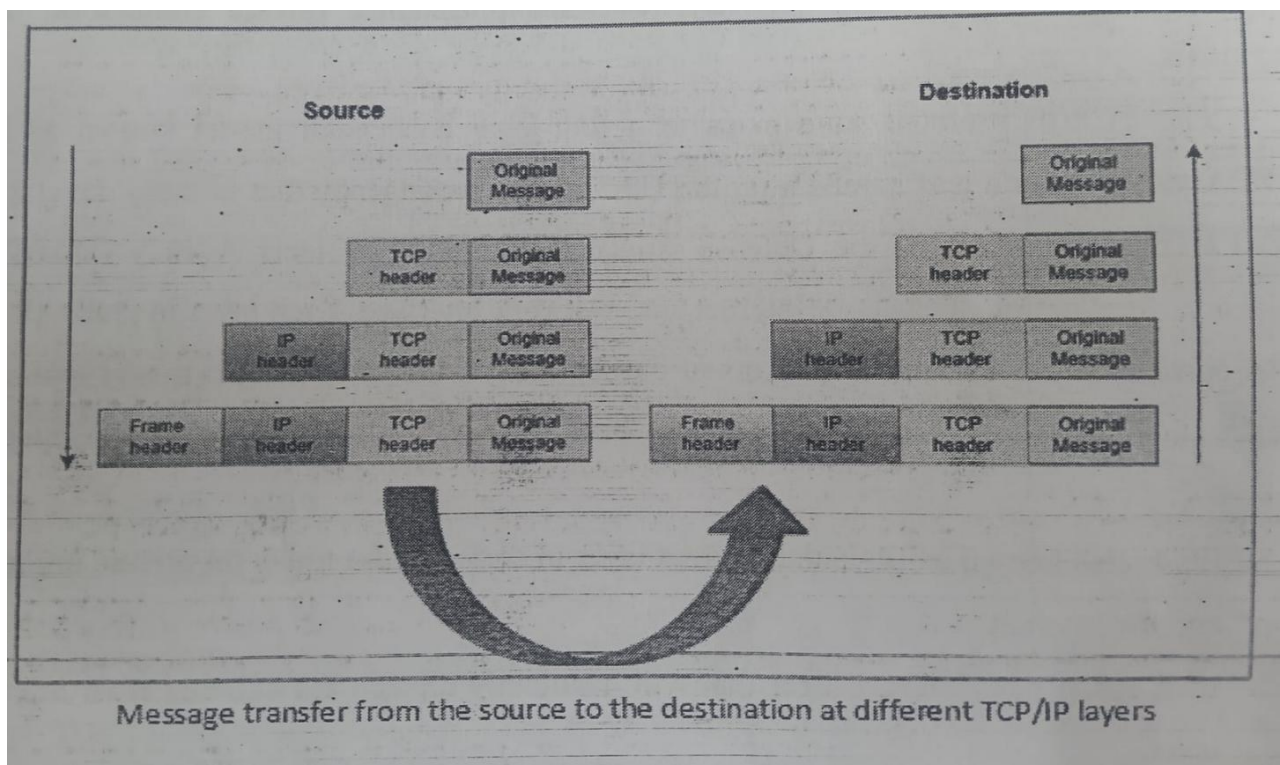
Transmission Control Protocol/Internet Protocol (TCP/IP) is the protocol that is used on the Internet, which is the collection of thousands of networks worldwide that connect research facilities, universities, libraries, government agencies, private companies, and individuals.

The TCP/IP protocol suite maps to a four-layer conceptual model known as the DARPA model, which was named after the U.S. government agency that initially developed TCP/IP. The four layers of the Defense Advanced Research Projects Agency (DARPA) model are: Application, Transport, Internet, and Network Interface. Each layer in the DARPA model corresponds to one or more layers of the seven-layer OSI model. This model is shown in the following figure. The TCP/IP protocol suite has two sets of protocols at the Internet layer.

1. IPv4, also known as IP, is the Internet layer in common use today on private intranets and the Internet.
2. IPv6 is the new Internet layer that will eventually replace the existing IPv4 Internet layer.

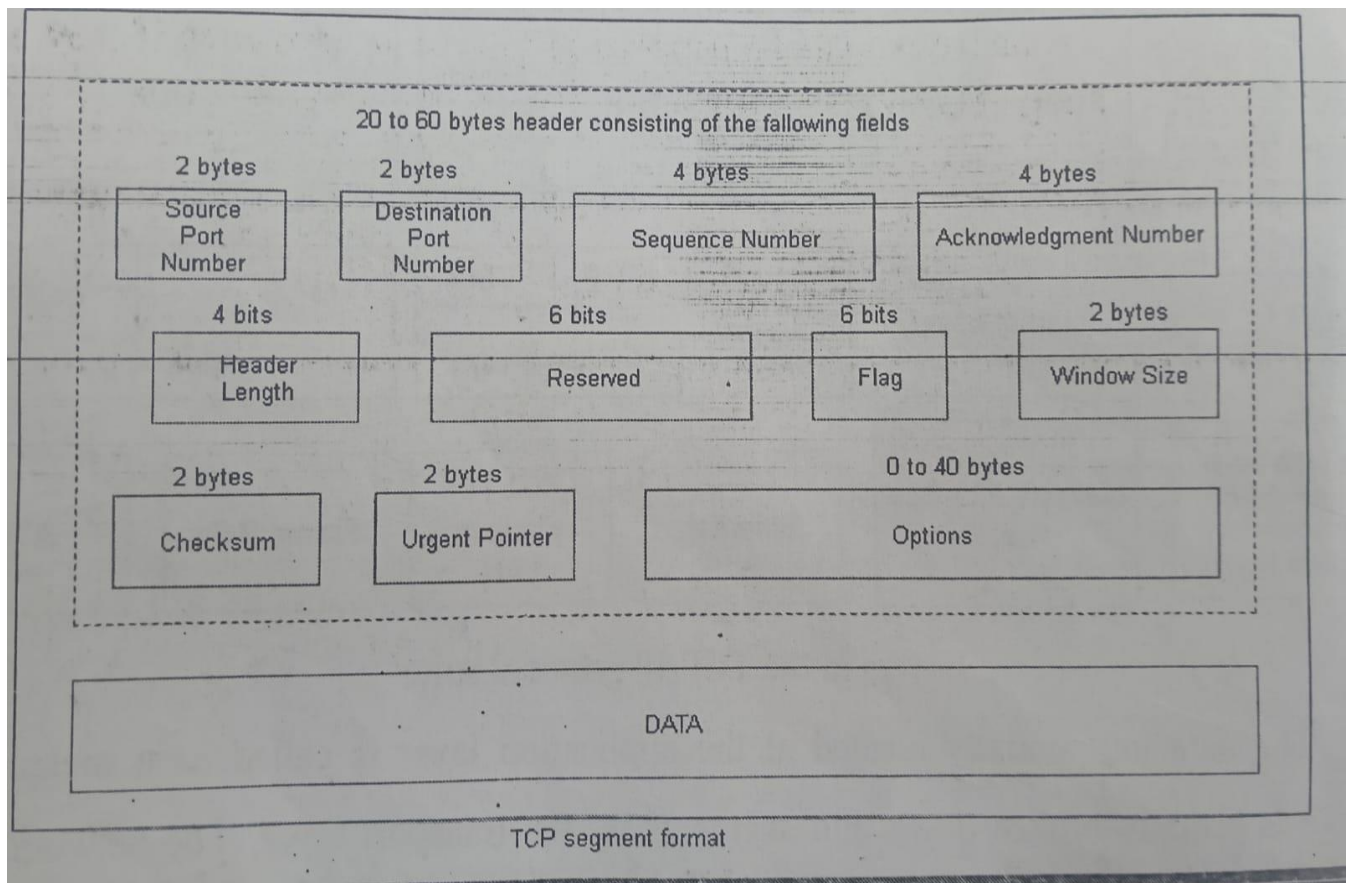


The data unit initially created at the application layer is called as a message. A message is actually broken down into segments by the transport layer. The network layer adds the IP header to this block and gives the result to the data link layer. The data link layer adds the frame header and gives it to the physical layer for transmission. At the physical layer, the actual bits transmitted as voltage pulses. At opposite process happens at the destination end, where each layer removes the previous layer's header and finally the application layer receives the original message. This is shown in the following figure:



TCP Segment Format:

A TCP segment consists of a header of size 20 to 60 bytes, followed by the actual data. The header consists of 20 bytes if the TCP packet does not contain any options. Otherwise, the header consists of 60 bytes. That is, a maximum of 40 bytes are reserved for options. Options can be used to convey additional information to the destination. The TCP segment format is in the following figure:

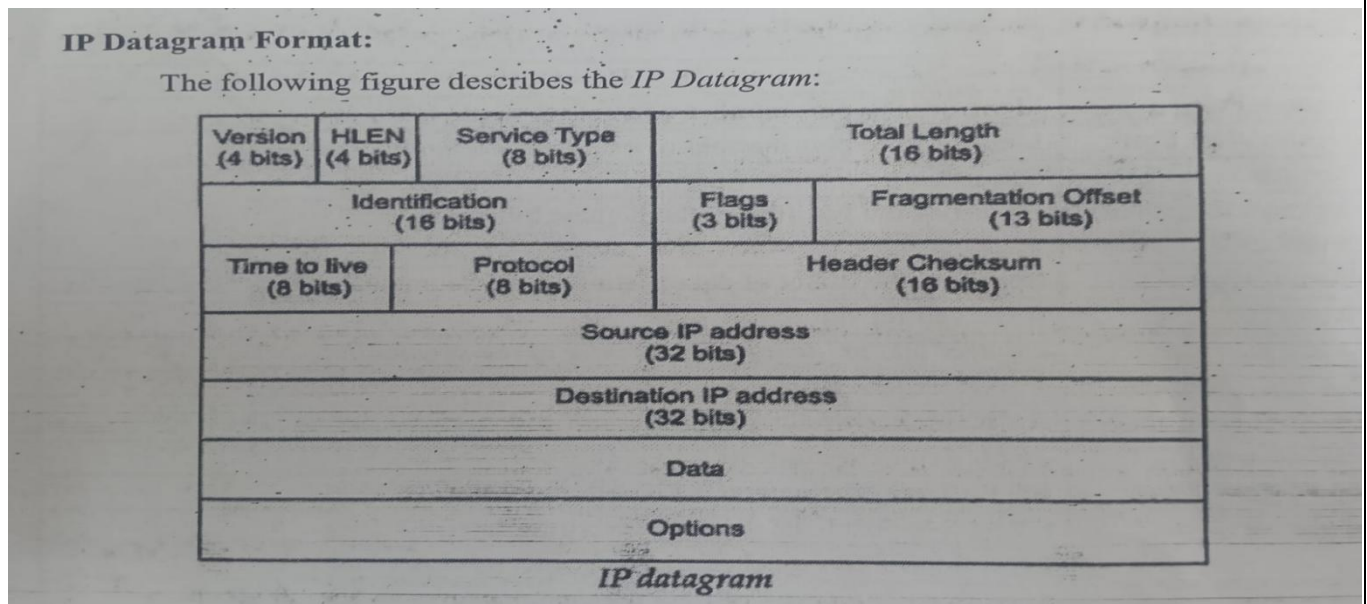


Explanation:

Item	Description
Source Port	Identifies the port number of a source application program.
Destination Port	Identifies the port number of a destination application program.
Sequence Number	Specifies the sequence number of the first byte of data in this segment.
Acknowledgment Number	Identifies the position of the highest byte received.
Data Offset	Specifies the offset of data portion of the segment.
Reserved	Reserved for future use.
Code	<i>Control bits to identify the purpose of the segment:</i>
	URG: Urgent pointer field is valid.
	ACK: Acknowledgement field is valid.
	PSH: Segment requests a PUSH.
	RST: Resets the connection.
	SYN: Synchronizes the sequence numbers.
	FIN: Sender has reached the end of its byte stream.
Window	Specifies the amount of data the destination is willing to accept.
Checksum	Verifies the integrity of the segment header and data.
Urgent Pointer	Indicates data that is to be delivered as quickly as possible. This pointer specifies the position where urgent data ends.
Options	End of Option List
	Indicates the end of the option list. It is used at the final option, not at the end of each option individually. This option needs to be used only if the end of the options would not otherwise coincide with the end of the TCP header.
	No Operation
	Indicates boundaries between options. Can be used between other options; for example, to align the beginning of a subsequent option on a word boundary. There is no guarantee that senders will use this option, so receivers must be prepared to process options even if they do not begin on a word boundary.
	Maximum Segment Size
	Indicates the maximum segment size TCP can receive. This is only sent in the initial connection request.

IP Datagram Format:

The following figure describes the IP Datagram:



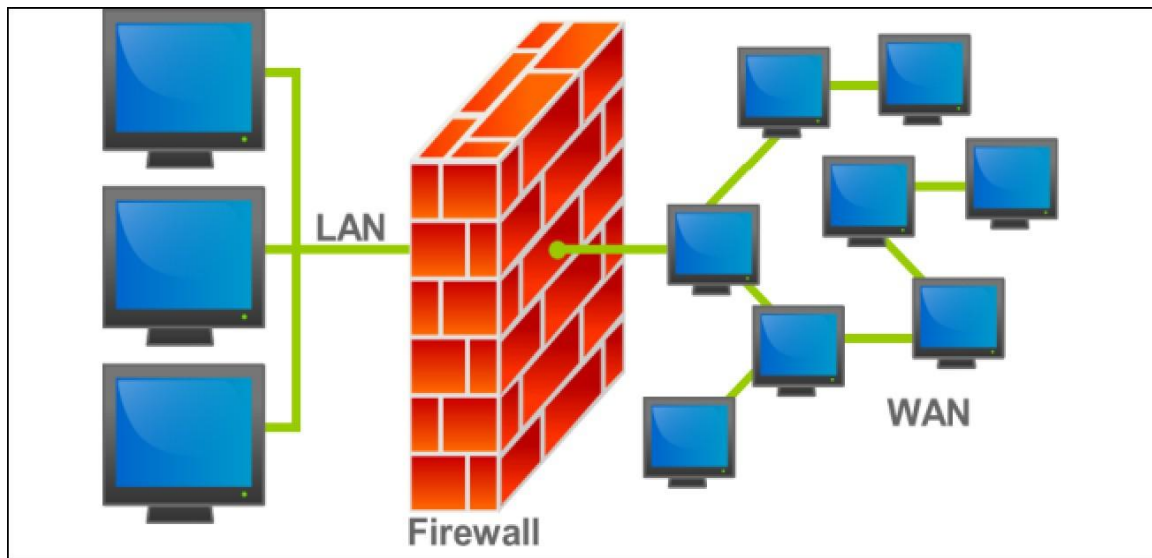
Explanation:

1. Version - currently has the value 4
2. Header length - the number of 32-bit words in the header
3. Type of Service - contains a 3-bit precedence field (that is ignored today), 4 service bits, and 1 unused bit.
4. Total Length - specified in bytes.
5. Identification - uniquely identifies the datagram. Usually incremented by 1 each time a datagram is sent.
6. Flags and Fragmentation Offset - used for fragmentation
7. Time to Live - Upper limit of routers
8. Protocol - Tells IP where to send the datagram up to.
9. Header checksum - Only covers the header, not the data.
10. Source IP address - the sender
11. Destination IP address - the final destination
12. Options - optional data.

Firewalls

A **Firewall** is a network security system, either hardware or software based, that monitors and controls incoming and outgoing network traffic based on set of security rules.

A firewall typically establishes a barrier between a trusted internal network and untrusted external network, such as the Internet.



Firewall Design Principles:

Information systems in corporations, government agencies, and other organizations have undergone a steady evolution:

1. Centralized data processing system, with a central mainframe supporting a number of directly connected terminals
2. Local area networks (LANs) interconnecting PCs and terminals to each other and the mainframe
3. Premises network, consisting of a number of LANs, interconnecting PCs, servers, and perhaps a mainframe or two.
4. Enterprise-wide network, consisting of multiple, geographically distributed premises networks interconnected by a private wide area network (WAN)
5. Internet connectivity, in which the various premises networks all hook into the Internet and may or may not also be connected by a private WAN

Firewall Characteristics

The following are design goals for a firewall:

1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall.
2. Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies.
3. The firewall itself is immune to penetration. This implies that use of a trusted system with a secure operating system.

The following four general techniques that firewalls use to control access and enforce the site's security policy. Originally, firewalls focused primarily on service control, but they have since devolved to provide all four:

1. **Service control:** Determines the types of Internet services that can be accessed, inbound or outbound. The firewall may filter traffic on the 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 a Web or mail service.
2. **Direction control:** Determines the direction in which particular service requests 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. This feature is typically applied to users inside the firewall perimeter (local users). It may also be applied to incoming traffic from external users; the latter requires some form of secure authentication technology, such as is provided in IPSec.
4. **Behavior control:** Controls how particular services are used. For example, the firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local Web server.

The following capabilities are within the scope of a firewall:

1. 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. The use of a single choke point simplifies security management because security capabilities are consolidated on a single system or set of systems.
2. A firewall provides a location for monitoring security-related events. Audits and alarms can be implemented on the firewall system.
3. A firewall is a convenient platform for several Internet functions that are not security related. These include a network address translator, which maps local addresses to Internet addresses, and a network management function that audits or logs Internet usage.
4. A firewall can serve as the platform for IPSec. Using the tunnel mode capability, the firewall can be used to implement virtual private networks.

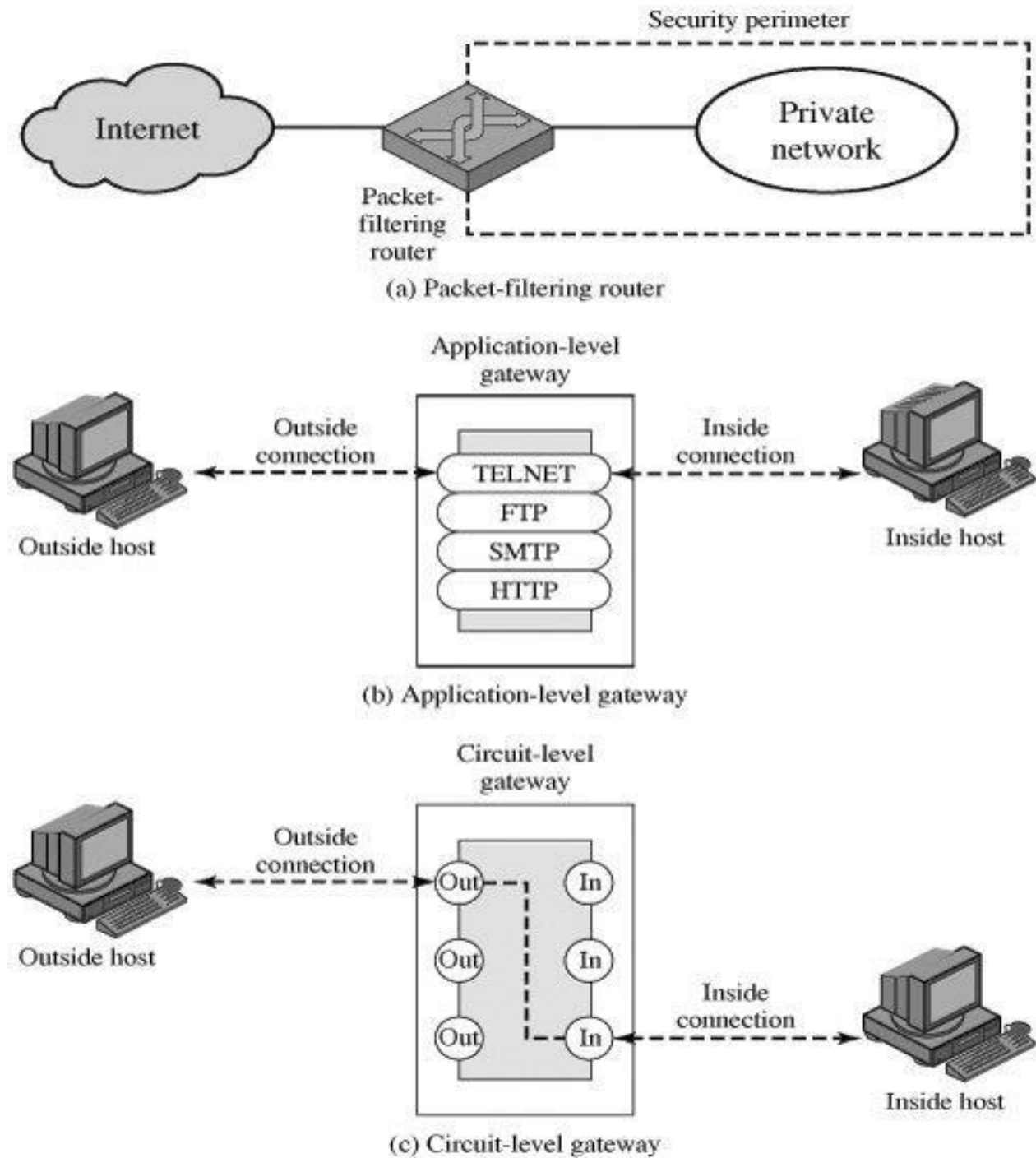
Limitations of Firewalls:

Firewalls have their limitations, including the following:

1. The firewall cannot protect against attacks that bypass the firewall. Internal systems may have dial-out capability to connect to an ISP. An internal LAN may support a modem pool that provides dial-in capability for traveling employees and telecommuters.
2. The firewall does not protect against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.
3. 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.

Types of Firewalls:

The following Figure illustrates the three common types of firewalls: packet filters, application-level gateways, and circuit-level gateways.



1. Packet-Filtering Router

A packet-filtering router applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet. The router is typically configured to filter packets going in both directions (from and to the internal network). Filtering rules are based on information contained in a network packet:

- a. **Source IP address:** The IP address of the system that originated the IP packet (e.g., 192.178.1.1)
- b. **Destination IP address:** The IP address of the system the IP packet is trying to reach (e.g., 192.168.1.2)
- c. **Source and destination transport-level address:** The transport level (e.g., TCP or UDP) port number, which defines applications such as SNMP or TELNET
- d. **IP protocol field:** Defines the transport protocol
- e. **Interface:** For a router with three or more ports, which interface of the router the packet came from or which interface of the router the packet is destined for

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:

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

The default discard policy is 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 more likely to see the firewall as a hindrance (obstruction).

2. Application-Level Gateway:

An application-level gateway, also called a proxy server, acts as a relay of application-level traffic ([Figure b](#)). 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. A prime disadvantage of this type of gateway is the additional processing overhead on each connection.

3. Circuit-Level Gateway:

A third type of firewall is the circuit-level gateway ([Figure c](#)).

- i. This can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications.
- ii. Relays two TCP connections
- iii. Requires security by limiting which such connections are allowed
- iv. Once created usually relays traffic without examining contents
- v. Typically used when trust internal users by allowing general outbound connections
- vi. SOCKS package (Example of this gateway) commonly used for this

Bastion Host

A bastion host is a system identified by the firewall administrator as a critical strong point in the network's security. Typically, the bastion host serves as a platform for an application-level or circuit level gateway. Common characteristics of a bastion host include the following:

- i. Highly secure host system
- ii. Potentially exposed to “hostile” elements
- iii. Hence is secured to withstand this
- iv. May support 2 or more net connections
- v. May be trusted to enforce trusted separation between network connections

- vi. Runs circuit/application level gateways
- vii. Or provides extremely accessible services

Firewall Configurations:

The following figure illustrates three common firewall configurations.

1. **Single-homed bastion** configuration ([Figure a](#)), the firewall consists of two systems: a packet-filtering router and a bastion host. Typically, the router is configured so that

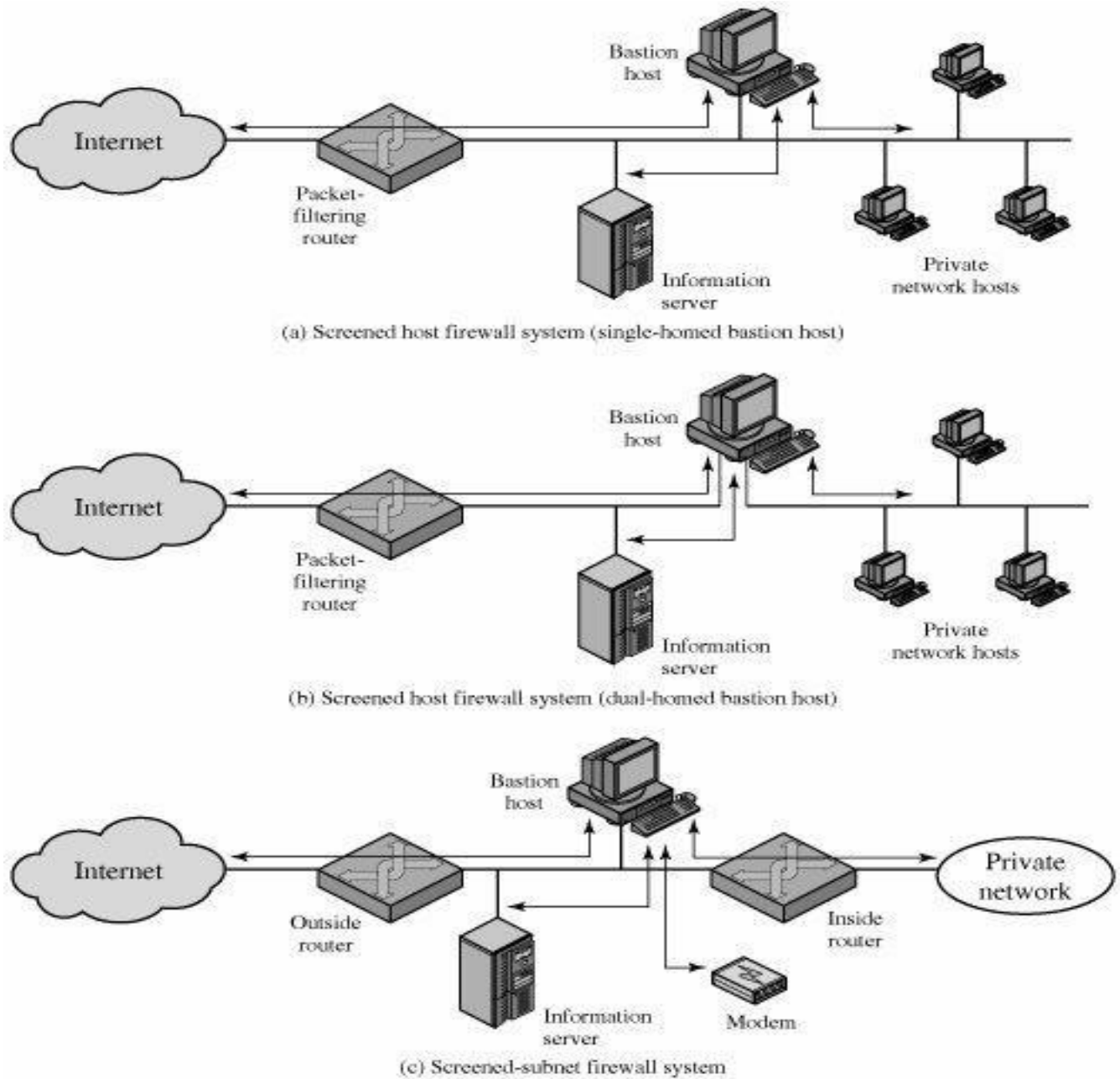
- i. For traffic from the Internet, only IP packets destined for the bastion host are allowed in.
- ii. For traffic from the internal network, only IP packets from the bastion host are allowed out.

2. In the single-homed configuration, if the packet-filtering router is completely compromised, traffic could flow directly through the router between the Internet and other hosts on the private network. The **screened host firewall, dual-homed bastion** configuration physically prevents such a security breach ([Figure b](#)). The advantages of dual layers of security that were present in the previous configuration are present here as well. Again, an information server or other hosts can be allowed direct communication with the router if this is in accord with the security policy.

3. The **screened subnet firewall** configuration of [Figure c](#) is the most secure of those we have considered. In this configuration, two packet-filtering routers are used, one between the bastion host and the Internet and one between the bastion host and the internal network. This configuration offers several advantages:

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

Figure: Firewall Configurations



IP Security

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

The network layer in the Internet is often referred to as Internet protocol or IP layer. IP Sec helps create authentication and confidential packets for the IP layer.

$$\text{IPSec} = \text{AH} + \text{ESP} + \text{IPcomp} + \text{IKE}$$

AH - Authentication Header

ESP - Encapsulation Security Payload

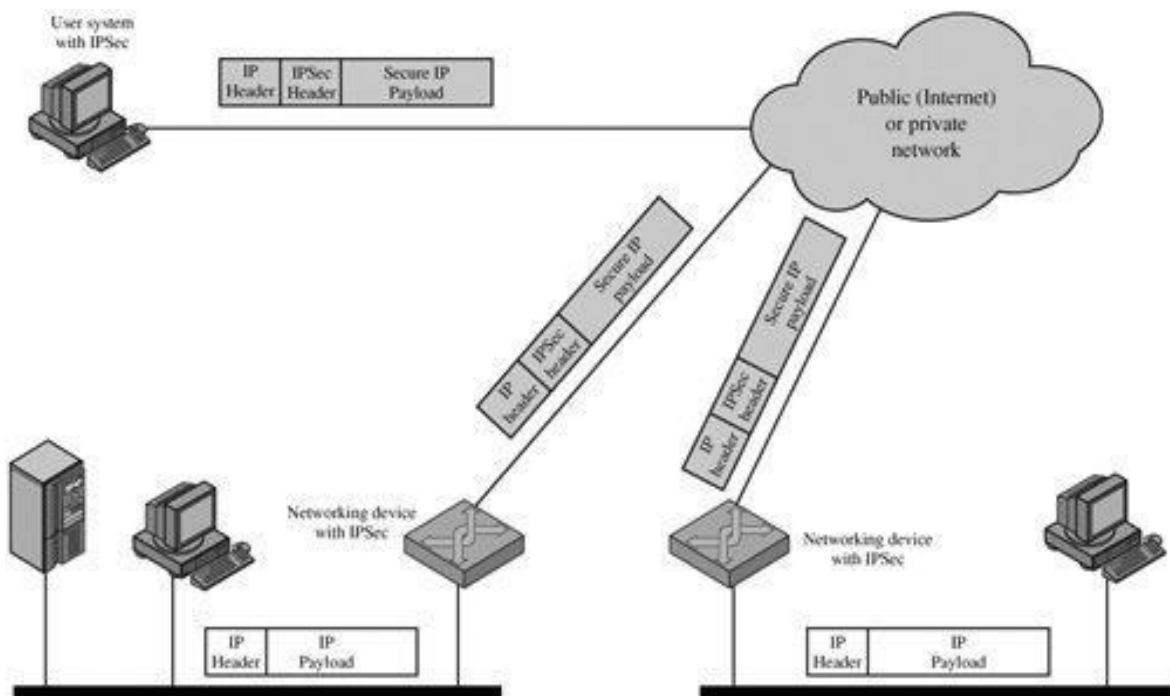
IPcomp - IP Compression

IKE - Sets up keys and algorithms for IP and ESP

IP Security Issues:

1. Eavesdropping
2. Modification of packets in transit
3. Identity spoofing (forget source IP addresses)
4. Denial of service
5. Many solutions are application-specific: TLS (Transport Layer Security) for Web, S/MIME (Secure/Multipurpose Internet Mail Extension) for email, SSH (Secure Shell) for remote login
6. IP Sec aims to provide a framework of open standards for secure communications over IP
7. Protect every protocol running on top of IPv4 and IPv6.

An IP Security Scenario:



IP Security Architecture

The IPsec specification has become quite complex. To get a feel for the overall architecture, we begin with a look at the documents that define IPsec.

IPsec Document Overview:

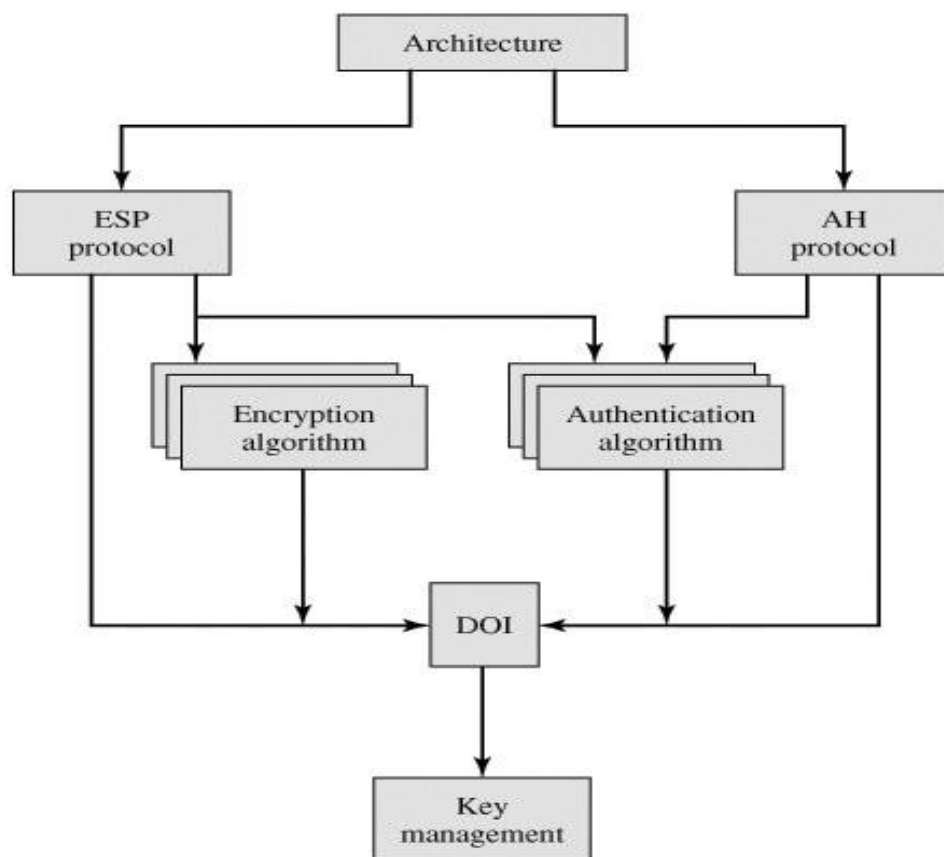
The IPsec specification consists of numerous documents. The most important of these, issued in November of 1998, are RFCs 2401, 2402, 2406, and 2408:

- RFC 2401: An overview of a security architecture
- RFC 2402: Description of a packet authentication extension to IPv4 and IPv6
- RFC 2406: Description of a packet encryption extension to IPv4 and IPv6
- RFC 2408: Specification of key management capabilities

All the IP protocols must support these documents. In IPv6, all these documents are followed, whereas in IPv4 only some are followed.

In both these cases, the security features are implemented as extension headers that follow the IP header. The extension header for authentication is known as “Authentication Header (AH) protocol”.

The extension header for encryption is known as ESP protocol (Encapsulating Security Payload). These documents are divided into 7 groups as shown in the following diagram.



1. **Architecture:** Covers the general concepts, security requirements, definitions, and mechanisms defining IPsec technology.
2. **Encapsulating Security Payload (ESP):** Covers the packet format and general issues related to the use of the ESP for packet encryption and, optionally, authentication.

3. **Authentication Header (AH):** Covers the packet format and general issues related to the use of AH for packet authentication.
4. **Encryption Algorithm:** A set of documents that describe how various encryption algorithms are used for ESP.
5. **Authentication Algorithm:** A set of documents that describe how various authentication algorithms are used for AH and for the authentication option of ESP.
6. **Key Management:** Documents that describe key management schemes.
- 7 **Domain of Interpretation (DOI):** Contains values needed for the other documents to relate to each other. These include identifiers for approved encryption and authentication algorithms, as well as operational parameters such as key lifetime.

IPSec Services

IPSec provides security services at the IP layer by enabling a system to select required security protocols. Two protocols are used to provide security: an authentication protocol designated by the header of the protocol, Authentication Header (AH); and a combined encryption/authentication protocol designated by the format of the packet for that protocol, Encapsulating Security Payload (ESP). The services are

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets (a form of partial sequence integrity)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

Applications of IP Security:

IP Security provides capability of secured communication over the LAN, communication across some private network stations, documentation from private LAN to WAN and across the internet. The following are the examples:

1. Using IP Security, the company can build a secured private network with that he can communicate with another WAN or Internet or LAN.
2. An end user can communicate with Internet.

The main feature of IP Security is to support various applications that can encrypt & authenticate at the IP level.

Benefits of IP Security:

1. When implemented in a firewall or router, it provides security to all traffic crossing the firewall/router perimeter.
2. Because IP Security is below the transport layer, it is transparent to applications.
3. IP Security can provide security to individual users if required.
4. Additional capabilities when applied to a router.
5. Only an authorized router can advertise the presence of a new router.
6. A redirect message must come from the router to which the initial packet was sent.
7. Routing update cannot be forged.

Suppose user B receives an IP Security protected packet from one or more sources. Then B has to know the cryptographic key and algorithm to process that packet. This is done by inserting IP Security header in the packet. IP security packet may be AH (Authentication Header) or ESP (Encapsulation Security Payload). The following concepts are included in IP Security.

1. Security Association (SA), SA Database
2. Security Policy Database
3. AH and ESP
4. Transport and Tunnel Mode

1. SA: Security Association (SA) is a one-way relationship between a sender and a receiver that provides security services to the traffic carried on it.

- a. Security association (SA) defines
- b. Protocol used (AH, ESP)

- c. Mode (transport, tunnel)
- d. Encryption or hashing algorithm to be used
- e. Negotiated keys and key lifetimes
- f. Lifetime of this SA
- g. ...plus other info

A system implementing IP security must keep a SA Database. When transmitting to destination 'X', the sender looks up for 'X' in the Table or SA Database. It consists information about how to transmit to 'X' I.e., it will provide SPI (Security Parameter Index), a key of 'X' and an algorithm, a sequence number etc.,. Now the sender will use the key, algorithm and send the packet to the 'X'. After receiving, the 'X' will again contact the SA database to get key algorithm and use that information to get plain data.

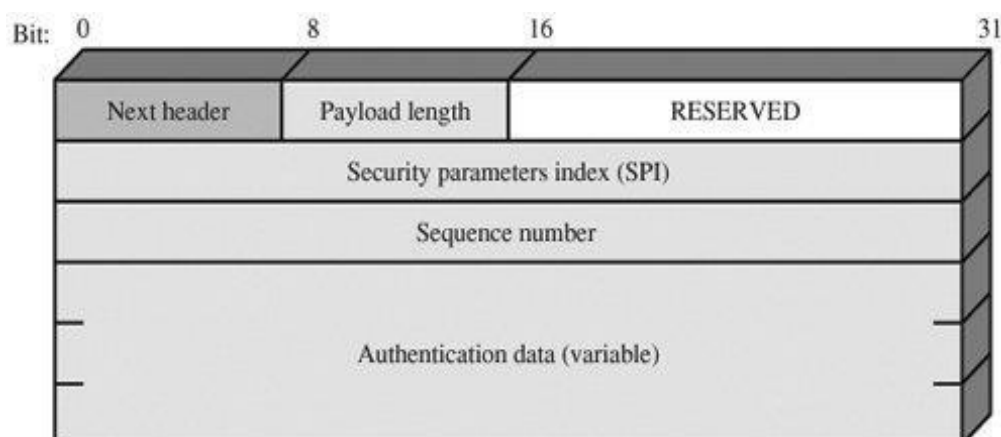
2. Security Policy Database(SPD): It contains entries which define the SA for all current traffic. Just like in firewalls, we can configure or program, so that, only some packets are received and some packets are sending out. This type of security policies are defined in the Security Policy Database. All the traffic whether incoming / outgoing must satisfy Security Policy in the Database.

3. AH and ESP: The AH (Authentication Header) is defined in RFC 2402 (Request For Comment) and ESP (Encapsulating Security Payload) defined in RFC 2406. These are two types of IP security Headers.

AH provides Integrity and Authentication. ESP provides Encryption and Integrity. The integrity is provided by both AH and ESP. But there is a slight difference between Integrity Protection by these two. AH provides Integrity for some fields inside IP Header. Both provide Integrity to the Header.

The elements of AH and ESP is described as the following:

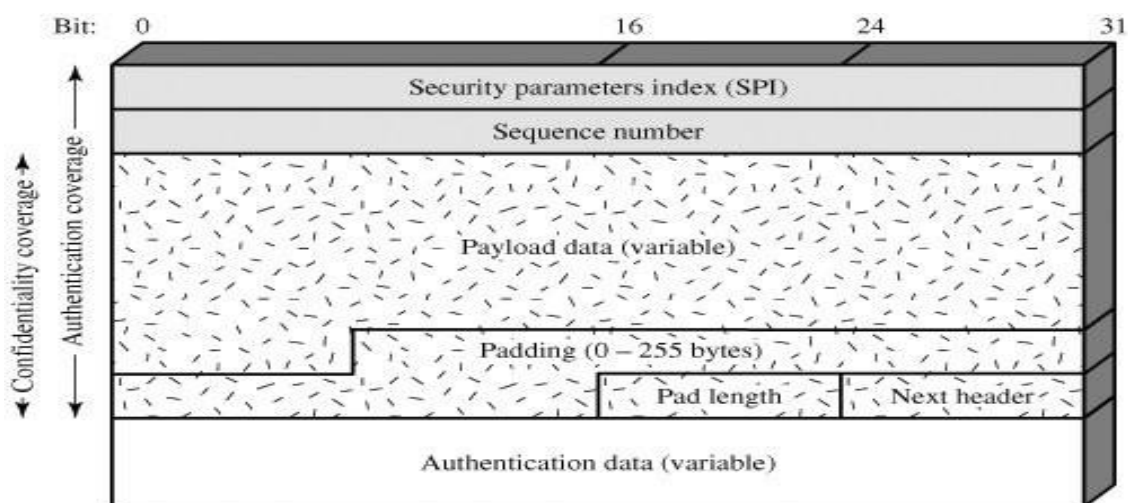
AH:



- **Next Header (8 bits) or version:** It indicates IP version which is 4 or 6 (IPv4 or IPv6) .
- **Payload Length (8 bits):** It denotes size of AH header.
- **Reserved (16 bits):** For future use.
- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** A monotonically increasing counter value.
- **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value (ICV), or MAC, for this packet.

ESP:

Figure 16.7. IPSec ESP format



The format of an ESP packet. It contains the following fields:

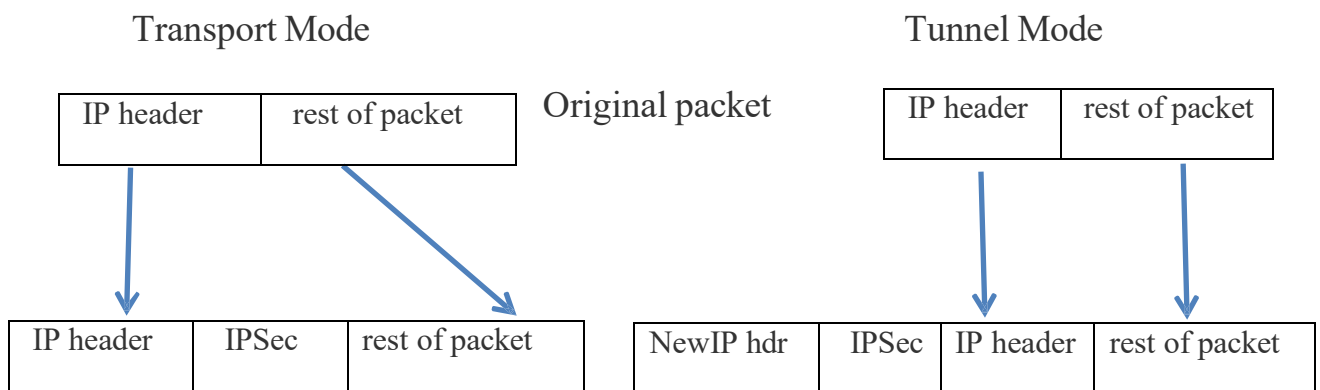
- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** Same as for AH.
- **Payload Data (variable):** This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
- **Padding (0-255 bytes):** The purpose of this field is discussed later.
- **Pad Length (8 bits):** Indicates the number of pad bytes.
- **Next Header (8 bits):** Same as in AH.

- **Authentication Data (variable):** Same as in AH.

4. Transport and Tunnel Mode: There are two types of modes in IP Security. They are

- Transport Mode:** It provides protection for upper layer protocols in the IP packet. In this mode, the IP Security information is placed between the IP Header and rest of the data.
- Tunnel Mode:** It provides protection to the entire IP packet. In this mode, IP Security information is appended to the original packet with a new header and IP Security information.

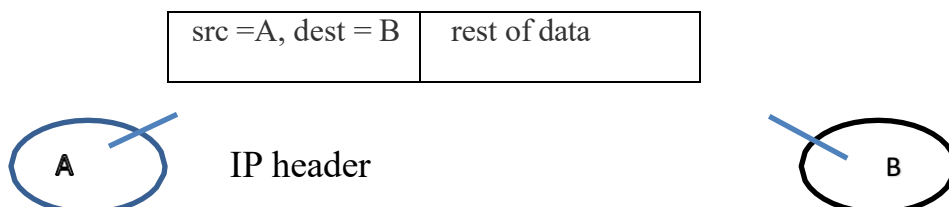
This is explained in the following figure:



Transport mode is used when communicating to end-to-end. Tunnel mode is used when communicating between firewall-to-firewall and end-to-end.

Suppose two firewalls (F1,F2) are established in two organizations. User A is in one organization wants to communicating with User B in another organization.

Here organization packet looks like as the following:



The organization packet from user 'A' must pass through F1-Internet-F2 to reach 'B'. The firewall F1 will attaches the secured information to the organization packet in the following way using Tunnel mode.

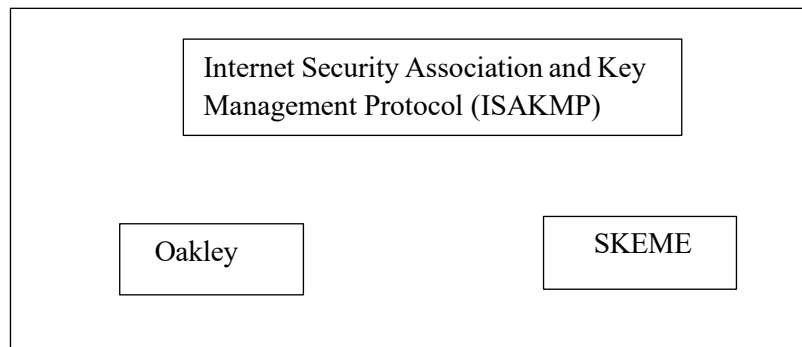
IP: src= F1, dest =F2	ESP	IP: src= A, dest =B	rest of data
-----------------------	-----	---------------------	--------------

In this way, Transport and Tunnel modes are used while communicating between two ends.

IPsec:IKE:

The Internet Key Exchange (IKE) is a protocol designed to create both inbound and outbound Security Associations (SAs) i.e., IKE creates SAs for IP Security.

Components of IKE: The following diagram shows the components of IKE.

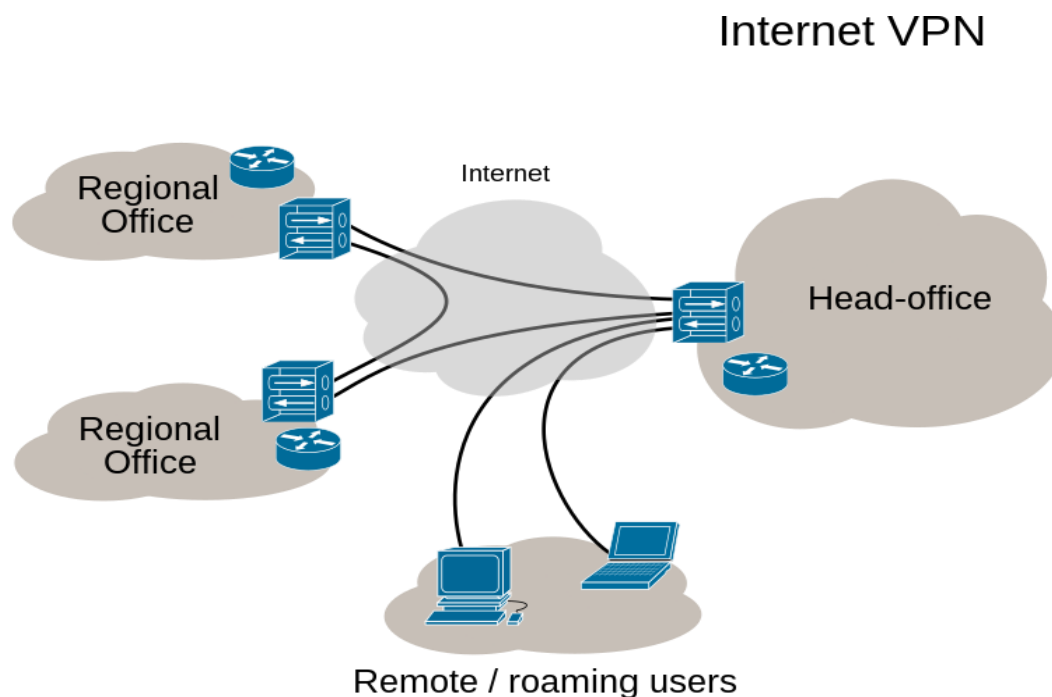


1. The Oakley protocol was developed by Hilarie Orman. It is a key creation protocol based on the Diffie-Hellman key-exchange method.
2. The SKEME (Secure Key Exchange Mechanism), designed by Hugo Krawczyk, is another protocol for key exchange. It uses public-key encryption for entity authentication in a key-exchange protocol.
3. The ISAKMP (Internet Security Association and Key Management) protocol designed by the National Security Agency (NSA) that actually implements the exchanges defined IKE. It defines several packets, protocols and parameters that allow the IKE exchanges to take place in standardized, formatted messages to create SAs.

Virtual Private Networks

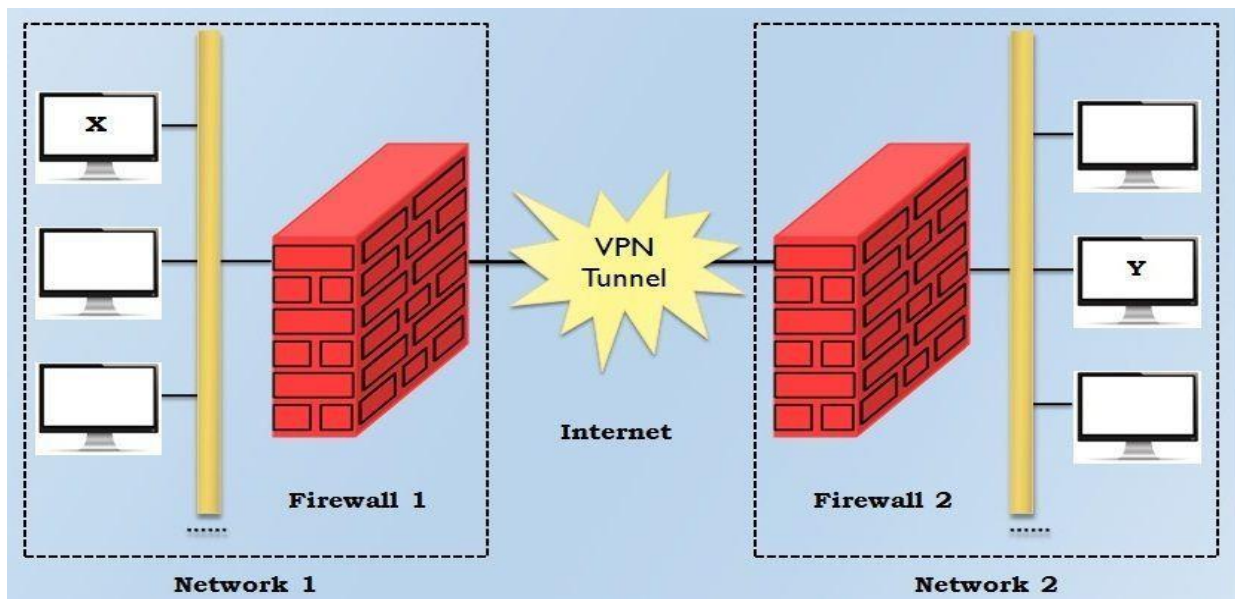
A **virtual private network (VPN)** extends a private network across a public network and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network. Applications running across a VPN may therefore benefit from the functionality, security, and management of the private network.

Virtual Private Networks



VPN Architecture:

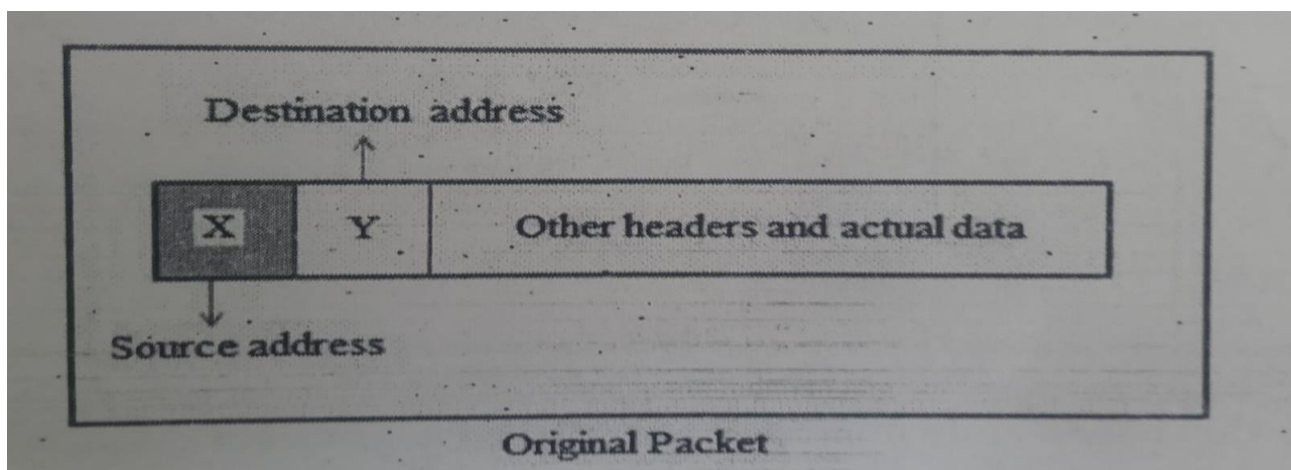
The idea of VPN is actually quite simple to understand. Suppose an organization has two networks, Network1 and Network2, which are physically apart from each other and we want to connect them using the VPN approach. In such a case, we setup two firewalls, Firewall1 and Firewall2. The encryption and decryption are performed by the firewalls. The architecture overview is shown in the following figure:



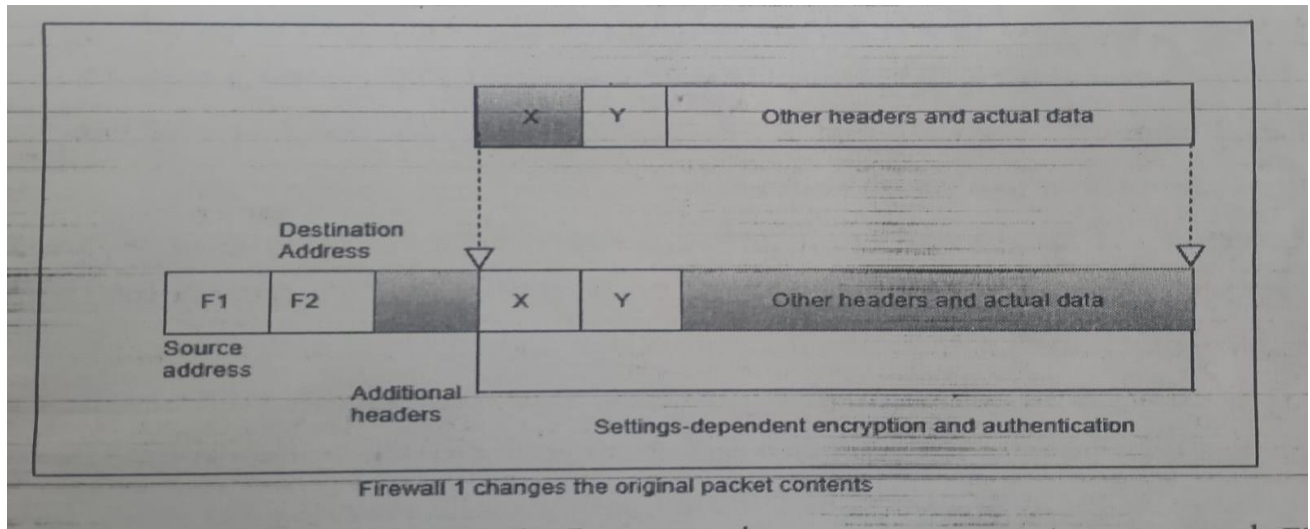
We have shown two networks, Network1 and Network2. Network1 connects to the Internet via a firewall named Firewall1. Similarly, Network2 connects to the Internet with its own firewall, Firewall2. The two firewalls are virtually connected to each other via the Internet. We have shown this with the help of a VPN tunnel between the two firewalls.

With this configuration, let us understand how VPN protects the traffic passing between any two hosts on the two different networks. For this, let us assume that host X on Network1 wants to send a data packet to host Y on Network2. This transmission would work as follows:

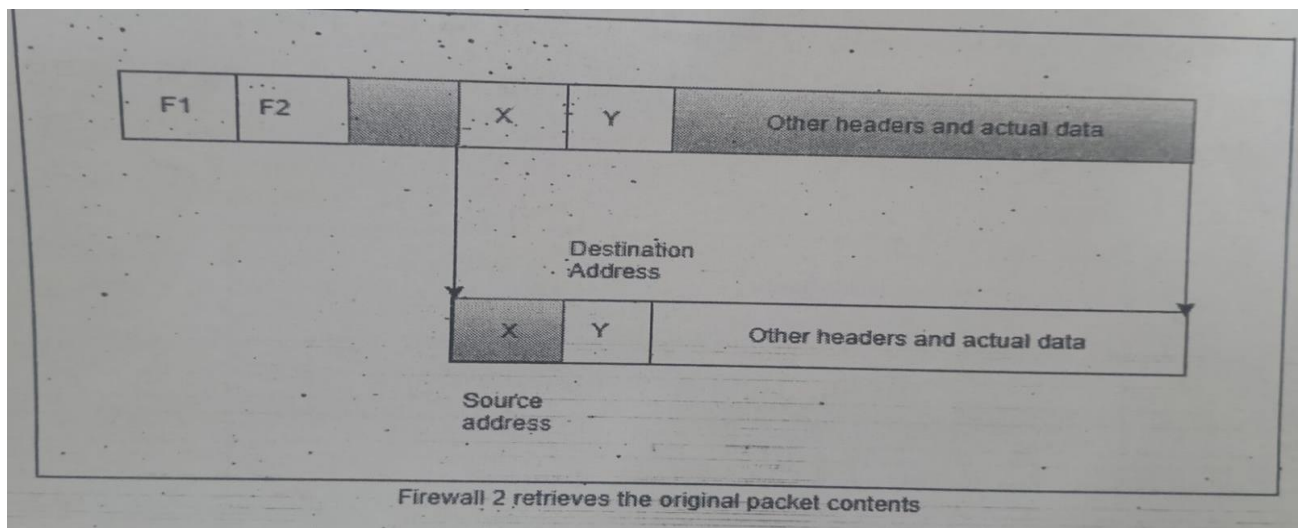
1. Host X creates the packet, inserts its own IP address as the source address and the IP address of host Y as the destination address. This is shown in the following figure. It sends the packet using the appropriate mechanism.



2. The packet reaches Firewall1. Now Firewall1 adds new headers to the packet. In these new headers, it changes the source IP address of the packet from that of host X to its own the IP address (i.e. the IP address of Firewall1, say F1). It also changes the destination IP address of the packet from that of host Y to the IP address of Firewall2, say F2. This is show in the following figure. It also performs the packet encryption and authentication, depending on the settings and sends the modified packet over the Internet.



3. The packet reaches Firewall2 over the Internet, via one or more routers, as usual. Firewall2 discards the outer header and performs the appropriate decryption and other cryptographic functions as necessary. This yields the original packet, as was created by host X in Step 1. This is shown in the following figure. It then takes a look at the plain text contents of the packet and realizes the packet is meant for host Y (because the destination address inside the packet specific host Y). Therefore, it delivers the packet to host Y.



VPN Protocols: There are mainly three types of protocols. They are

1. Point-to-Point Tunneling Protocol (PPTP): PPTP creates a tunnel and encapsulates the data packet. It uses a Point-to-Point Protocol (PPP) to encrypt the data between the connections. PPTP is one of the most widely used VPN protocol and has been in use since the time of Windows 95. Apart from Windows, PPTP is also supported on Mac and Linux.

2. Layer 2 Tunneling Protocol (L2TP): L2TP is a tunneling protocol that is usually combined with another VPN security protocol like IPSec to create a highly secure VPN connection. L2TP creates a tunnel between two L2TP connection points and IPSec protocol encrypts the data and handles secure communication between the tunnels.

3. Internet Protocol Security or IPSec: Internet Protocol Security or IPSec is used to secure Internet communication across an IP network. IPSec secures Internet Protocol communication by authenticating the session and encrypts each data packet during the connection. IPSec operates in two modes, Transport mode and Tunnel mode, to protect data transfer between two different networks. The transport mode encrypts the message in the data packet and the tunneling mode encrypts the entire data packet. IPSec can also be used with other security protocols to enhance the security system.
