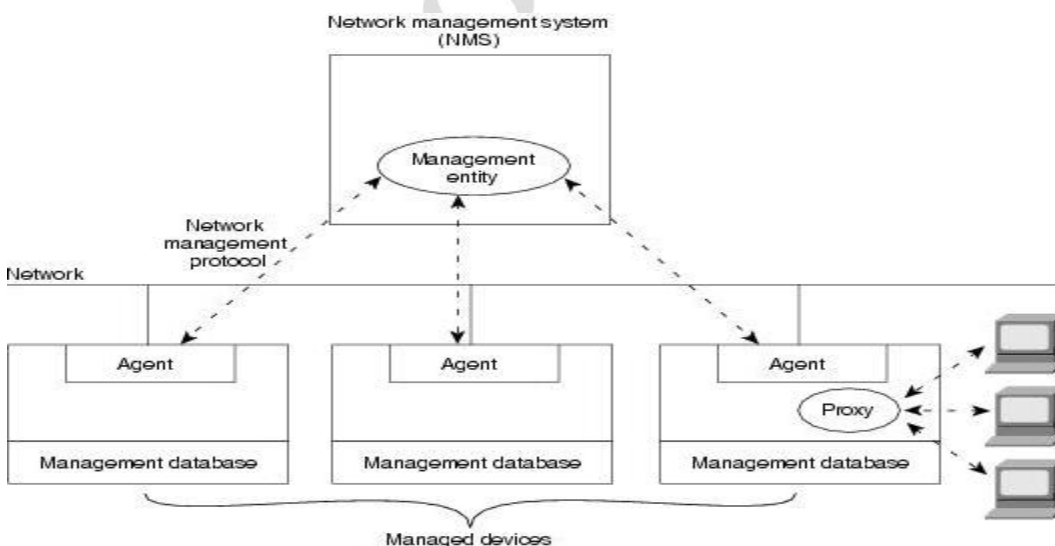


Network management

Network management refers to the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems.

The functions performed by a network management system can be categorized into the following five areas:-

- **Fault management:** - The goal of fault management is to detect, log, notify users of, and automatically fix network problems to keep the network running effectively. Because faults can cause downtime or unacceptable network degradation, Fault management involves first determining symptoms and isolating the problem. Then the problem is fixed and the solution is tested on all-important subsystems.
- **Configuration management:** - The goal of configuration management is to monitor network and system configuration information so that the effects on network operation of various versions of hardware and software elements can be tracked and managed.
- **Accounting management:** - The goal of accounting management is to measure network utilization parameters so that individual or group uses on the network can be regulated appropriately.
- **Performance management:** - The goal of performance management is to measure and make available various aspects of network performance so that internetwork performance can be maintained at an acceptable level. Example includes network throughput, user response times, and line utilization.
- **Security management:** - The goal of security management is to control access to network resources according to local guidelines so that the network cannot be sabotaged (intentionally or unintentionally) and sensitive information cannot be accessed by those without appropriate authorization. A security management subsystem, for example, can monitor users logging on to a network resource and can refuse access to those who enter inappropriate access codes.



- ❖ Network contains number of managed devices such as routers, bridges, switches, and hosts
- ❖ Network management involves monitoring and altering the configuration of such devices.
- ❖ An Agent is a part of network management system that resides in a managed device.

- ❖ Agents provide management information about the managed device and to accepts instructions for configuring the device.
- ❖ Network management station provides graphical view of the entire system.
- ❖ The manager exchanges management information with agent by using a network management protocol.

Simple Network Management Protocol

- ❖ **SNMP** consists of three key components: managed devices, agents, and network-management systems (NMSs).
- ❖ A managed device is a node that has an SNMP agent and resides on a managed network. These devices can be routers and access server, switches and bridges, hubs, computer hosts, or printers.
- ❖ An agent is a software module residing within a device. This agent translates information into a compatible format with SNMP.
- ❖ An NMS runs monitoring applications. They provide the bulk of processing and memory resources required for network management.
- ❖ The SNMP manager provides the interface between the human network manager and the management system.
- ❖ The **SNMP** agent provides the interface between the manager and the physical device(s) being managed.
- ❖ The SNMP manager and agent use an **SNMP** Management Information Base (MIB) and a relatively small set of commands to exchange information.
- ❖ The **SNMP MIB** is organized in a tree structure with individual variables, such as point status or description, being represented as leaves on the branches.

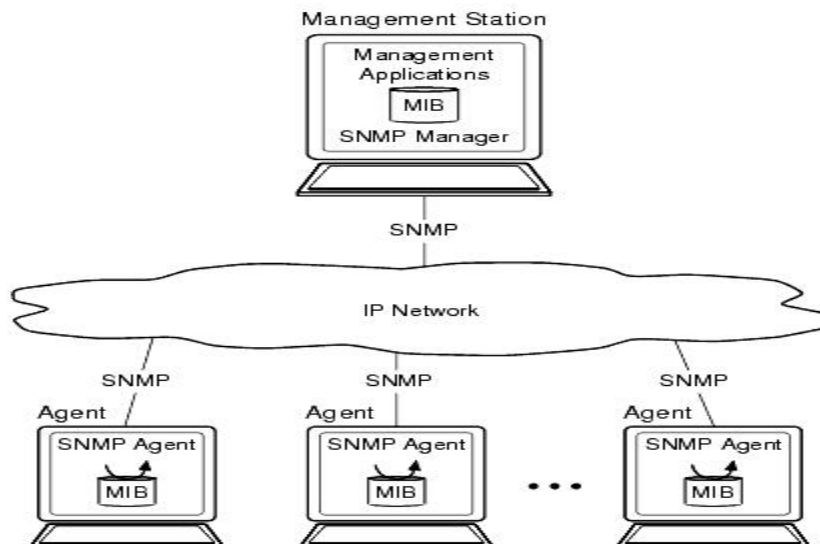
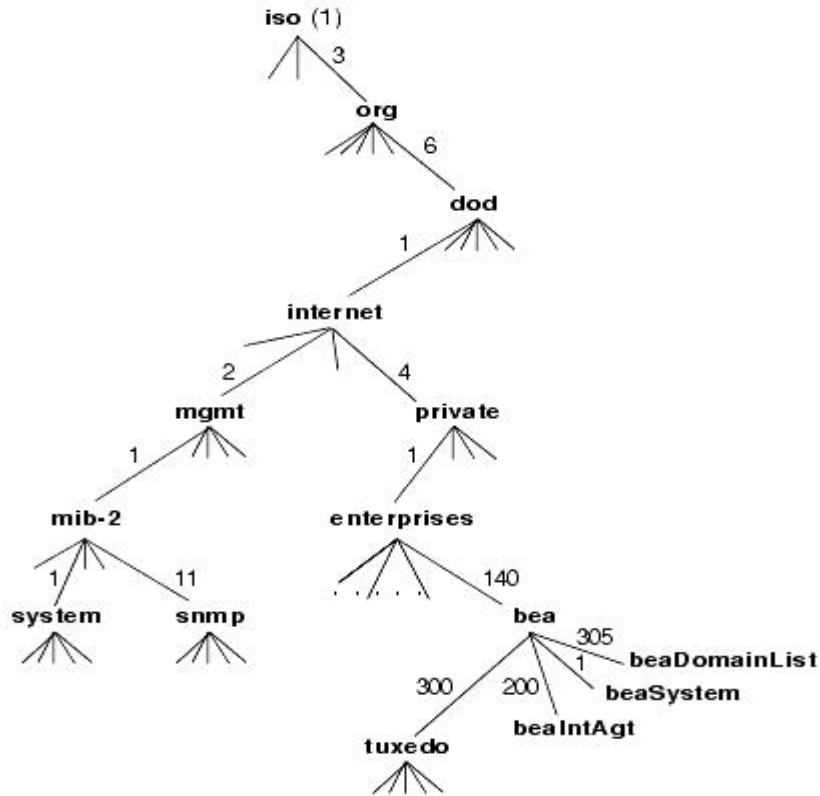


Figure: - SNMP-Managed Configuration.

- ❖ A long numeric tag or object identifier (OID) is used to distinguish each variable uniquely in the MIB and in SNMP messages.
- ❖ SNMP uses five basic messages (GET, GET-NEXT, GET-RESPONSE, SET, and TRAP) to communicate between the SNMP manager and the SNMP agent.
- ❖ The GET and GET-NEXT messages allow the manager to request information for a specific variable.
- ❖ The agent, upon receiving a GET or GET-NEXT message, will issue a GET-RESPONSE message to the **SNMP manager** with either the information requested or an error indication as to why the request cannot be processed.
- ❖ A SET message allows the SNMP manager to request a change be made to the value of a specific variable in the case of an alarm remote that will operate a relay.
- ❖ The SNMP agent will then respond with a GET-RESPONSE message indicating the change has been made or an error indication as to why the change cannot be made.
- ❖ The **SNMP TRAP** message allows the agent to spontaneously inform the SNMP manager of an "important" event.

Structure of Management Information (SMI)

- ❖ SMI defines the structure of the MIB information and the allowable data types. The SMI identifies how resources within the MIB are represented and named.
- ❖ The philosophy behind SMI is to encourage simplicity and extensibility within the MIB.
- ❖ The SNMP specification includes a template, known as an Abstract Syntax Notation One (ASN.1) OBJECT TYPE macro, which provides the formal model for defining objects and tables of objects in the MIB.
- ❖ Several data types are allowed in SMI. the primitive data types consists of INTEGER, OCTET STRING, NULL, and OBJECT IDENTIFIER
- ❖ Primitive data types are written in uppercase, while user defined data types start with an uppercase letter but contain at least one character other than an uppercase letter.
- ❖ An OBJECT IDENTIFIER is represented as a sequence of nonnegative integers where each integer corresponds to a particular node in the tree.
- ❖ Data type is used to identify a managed object and relating its place in the object hierarchy.



.1.3.6.1.4.1.140.300 = absolute OID for "tuxedo" MIB

Management Information Base (MIB)

- ❖ MIBs are a collection of information organized hierarchically which define the properties of the managed object within the device to be managed (such as a router, switch, etc.)
- ❖ Each managed device keeps a database of values for each of the definitions written in the MIB. As such, it is not actually database but implementation dependant.
- ❖ Each vendor of SNMP equipment has an exclusive section of the MIB tree structure under their control and these are accessed using a protocol such as SNMP.
- ❖ There are two types of MIBs: scalar and tabular.
- ❖ Scalar objects define a single object instance whereas tabular objects define multiple related object instances grouped in MIB tables.

The following keywords are used to define a MIB object:

- ❖ **Syntax:** - Defines the abstract data structure corresponding to the object type. The SMI purposely restricts the ASN.1 constructs that can be used to promote simplicity.
- ❖ **Access:** - Defines whether the object value may only be retrieved but not modified (read-only) or whether it may also be modified (read-write).
- ❖ **Description:** - Contains a textual definition of the object type. The definition provides all semantic definitions necessary for interpretation; it typically contains information of the sort that would be communicated in any ASN.1 commentary annotations associated with the object.

MIB Object Identifiers

- ❖ Each object in the MIB has an *object identifier* (OID), which the management station uses to request the object's value from the agent.
- ❖ An OID is a sequence of integers that uniquely identifies a managed object by defining a path to that object through a tree-like structure called the *OID tree* or registration tree.
- ❖ When an SNMP agent needs to access a specific managed object, it traverses the OID tree to find the object.
- ❖ The MIB object identifier hierarchy and format is shown in the above figure.

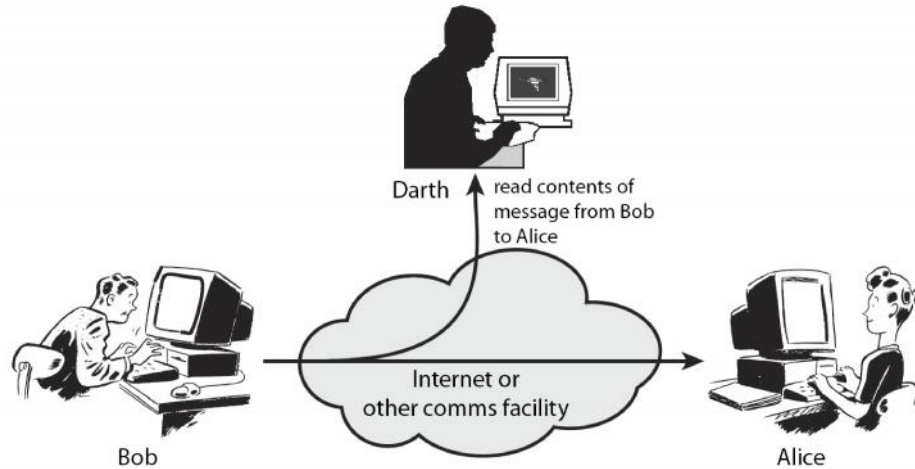
RMON (Remote Network Monitoring)

- ❖ RMON provides standard information that a network administrator can use to monitor, analyze, and troubleshoot a group of distributed local area networks (LANs) and interconnecting T-1/E-1 and T-2/E-3 lines from a central site.
- ❖ RMON specifically defines the information that any network monitoring system will be able to provide.
- ❖ The latest level is RMON Version 2 (sometimes referred to as "RMON 2" or "RMON2").
- ❖ RMON can be supported by hardware monitoring devices (known as "probes") or through software or some combination.
- ❖ A software agent can gather the information for presentation to the network administrator with a graphical user interface.
- ❖ A number of vendors provide products with various kinds of RMON support.
- ❖ RMON collects nine kinds of information, including packets sent, bytes sent, packets dropped, statistics by host, by conversations between two sets of addresses, and certain kinds of events that have occurred.
- ❖ A network administrator can find out how much bandwidth or traffic each user is imposing on the network and what Web sites are being accessed. Alarms can be set in order to be aware of impending problems.

What is Network Security?

Security refers to techniques for ensuring that data stored in a computer cannot be read or compromised by any individuals without authorization. Most security measures involve data encryption and passwords. Data encryption is the translation of data into a form that is unintelligible without a deciphering mechanism. A password is a secret word or phrase that gives a user access to a particular program or system.

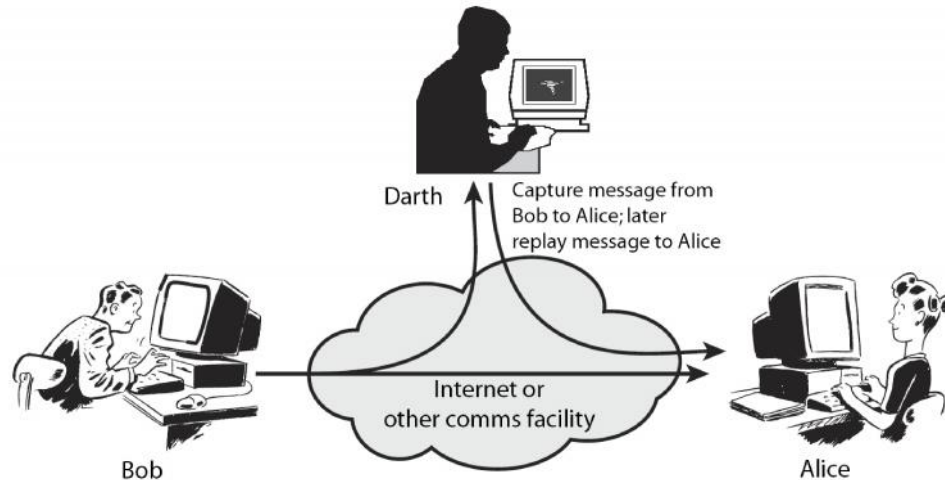
- **Passive** - An attack such as listening to communications then attacking the encryption scheme off line may be done.



- Attempt to learn or make use of information from the system but do not affect system resources
- Two types:
 - *Release of message contents*

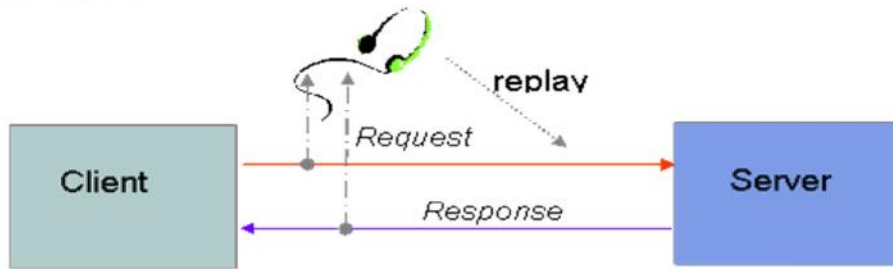
ex: telephone conversation, sensitive info in the form of a file, etc.

 - *Traffic analysis*
 - ✓ Pattern analysis
- Difficult to detect, so emphasis on prevention rather than detection
- **Active** - A common attack of this type is the man in the middle attack. During this attack the attacker may try to convince the victim that they are communicating with another party when they are really communicating with the attacker. The attacker may use the attack to gain passwords or other vital information.



- Attempt to modify data stream or create a false stream.
- Easy to detect but difficult to prevent.
- Types:
 - Masquerade - impersonating by replay of valid authentication sequence.
 - Replay – capture data unit and use it in retransmissions to produce unauthorized effect.
- **Dictionary attack** - A means attacking a system to determine passwords from hashed or encrypted passwords.

Eavesdropping



- Information transmitted over network can be observed and recorded by eavesdroppers (using a packet sniffer)
- Information can be replayed in attempts to access server
- Requirements: privacy, authentication, non-repudiation

Impersonating a client is another way for impostors to gain access to sensitive information stored on a server:



- Imposters attempt to gain unauthorized access to server
 - Ex. bank account or database of personal records
 - For example, in IP spoofing imposter sends packets with false source IP address
- Requirements: privacy, authentication

A denial of service (DoS) attacker can flood a network with requests, with the result that legitimate network users will no longer have access:

Denial of Service Attack



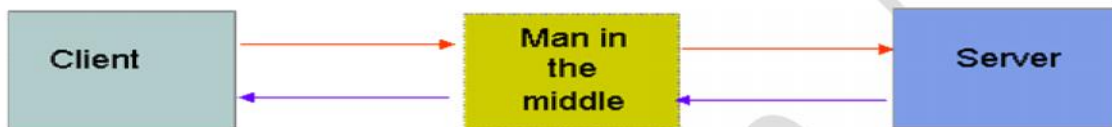
- Attacker can flood a server with requests, overloading the server resources
 - Results in *denial of service* to legitimate clients
- Distributed denial of service attack on a server involves coordinated attack from multiple (usually hijacked) computers
- Requirement: availability

An impostor can gain access to sensitive information by impersonating a legitimate server:

Server Imposter

- An imposter impersonates a legitimate server to gain sensitive information from a client
 - E.g. bank account number and associated user password
- Requirements: privacy, authentication, non-repudiation

An impostor can also mount a man-in-the middle attack by simultaneously impersonating both a legitimate client and a legitimate server.

Man-in-the-Middle Attack

- An impostor manages to place itself as *man in the middle*
 - convincing the server that it is legitimate client
 - convincing legitimate client that it is legitimate server
 - gathering sensitive information and possibly hijacking session
- Requirements: integrity, authentication

A client machine can be infected with malicious code, such as a worm or virus, that is downloaded from an untrustworthy server:

Malicious Code

- A client becomes infected with malicious code
 - Opening attachments in email messages
 - Executing code from bulletin boards or other sources
- Virus: code that, when executed, inserts itself in other programs
- Worms: code that installs copies of itself in other machines attached to a network
- Many variations of malicious code
- Requirements: privacy, integrity, availability

To deal with these threats, the following security requirements are needed: privacy or confidentiality, integrity, authentication, non-repudiation, etc.

1) **Confidentiality** (the information cannot be understood by anyone for whom it was unintended)

only sender, intended receiver should “understand” message contents

- sender encrypts message, receiver decrypts message

2) **Integrity** (the information cannot be altered in storage or transit between sender and intended receiver without the alteration being detected)

Sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

3) **Non-repudiation** (the creator/sender of the information cannot deny at a later stage his or her intentions in the creation or transmission of the information)

4) **Authentication** (the sender and receiver can confirm each others identity and the origin/destination of the information)

Applications of Cryptography to Security

- The science and art of manipulating messages to make them secure is called **cryptography**.
- Original message to be transformed is called **plaintext**.
- Resulting message after the transformation is called **ciphertext**.
- Process of converting plaintext to ciphertext is called **encryption**
- Reverse process is called **decryption**
- Algorithm used for encryption and decryption is called **Cipher**.

Ex : Substitution and Transposition Ciphers

Substitution Cipher

- Substitution ciphers are a common technique for altering messages in games and puzzles.
- Each letter of the alphabet is mapped into another letter.

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

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

Transposition Cipher :

- Here the order in which the letters of the message appear is altered.

- Substitution and transposition techniques are easily broken.

Cryptographic method must meet several requirements

1. It must be easy to implement
2. It should be deployable on large scale
3. It must provide security to all of its users.
4. It should prevent an attacker from deriving the key even when a large sample of the plaintext and corresponding ciphertext is known

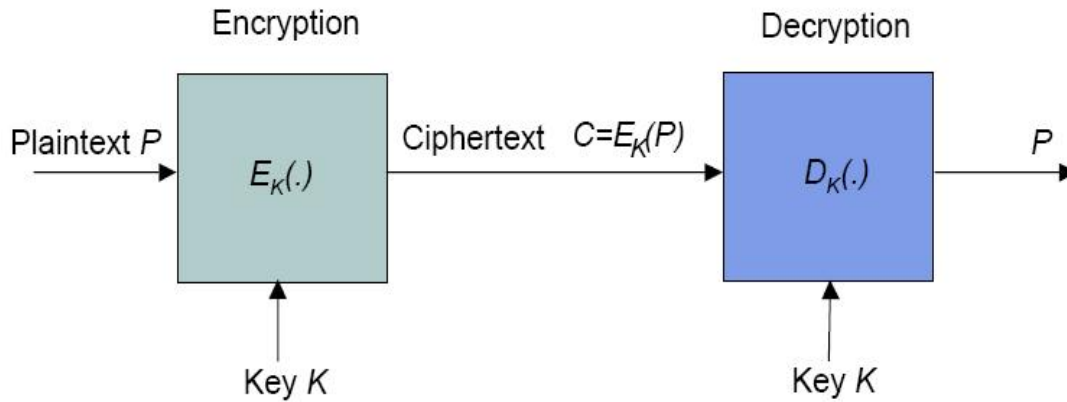
Cryptography is best known as a way of keeping the contents of a message secret. Confidentiality of network communications, for example, is of great importance for e-commerce and other network applications.. In particular, cryptography allows the network business and customer to verify the authenticity and integrity of their transactions. If the trend to a global electronic marketplace continues, better cryptographic techniques will have to be developed to protect business transactions.

Cryptography is best known as a way of keeping the contents of a message secret. Confidentiality of network communications, for example, is of great importance for e-commerce and other network applications.. In particular, cryptography allows the network business and customer to verify the authenticity and integrity of their transactions. If the trend to a global electronic marketplace continues, better cryptographic techniques will have to be developed to protect business transactions.

Sensitive information sent over an open network may be scrambled into a form that cannot be understood by a hacker or eavesdropper. This is done using a mathematical formula, known as an encryption algorithm, which transforms the bits of the message into an unintelligible form. The intended recipient has a decryption algorithm for extracting the original message. There are many examples of information on open networks, which need to be protected in this way, for instance, bank account details, credit card transactions, or confidential health or tax records.

Symmetric-key cryptography

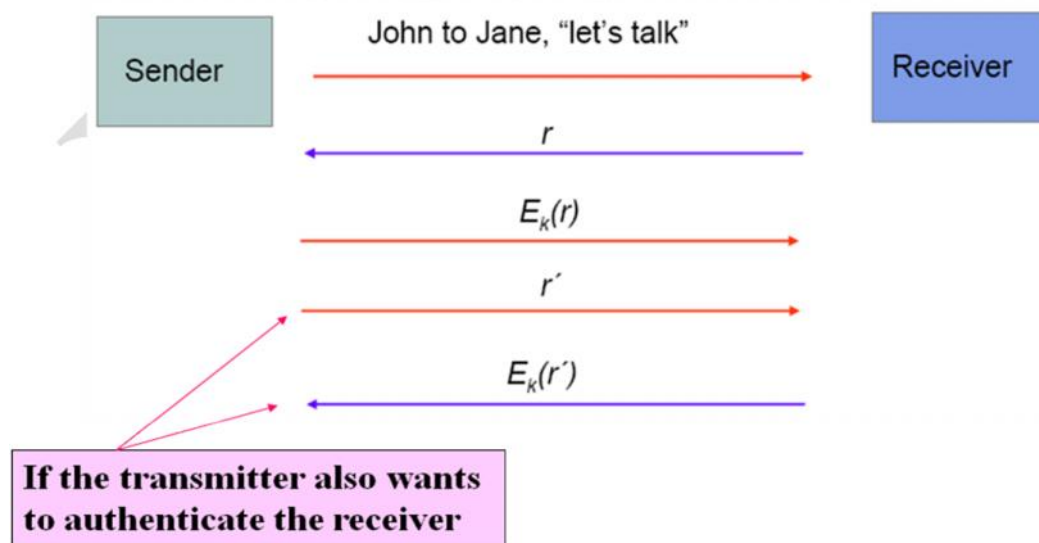
- ❖ An encryption system in which the sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message.
- ❖ Symmetric-key systems are simpler and faster, but their main drawback is that the two parties must somehow exchange the key in a secure way. Public-key encryption avoids this problem because the public key can be distributed in a non-secure way, and the private key is never transmitted.
- ❖ Symmetric-key cryptography is sometimes called secret-key cryptography. The most popular symmetric-key system is the Data Encryption Standard (DES).
- ❖ The disadvantage of symmetric cryptography is that it presumes two parties have agreed on a key and been able to exchange that key in a secure manner prior to communication.



Secret key cryptography addresses the privacy requirement.

Example : Data Encryption Standard (DES)

Secret Key Authentication



r is some random number sent by sender

$E_k(r)$ is encryption of random number r

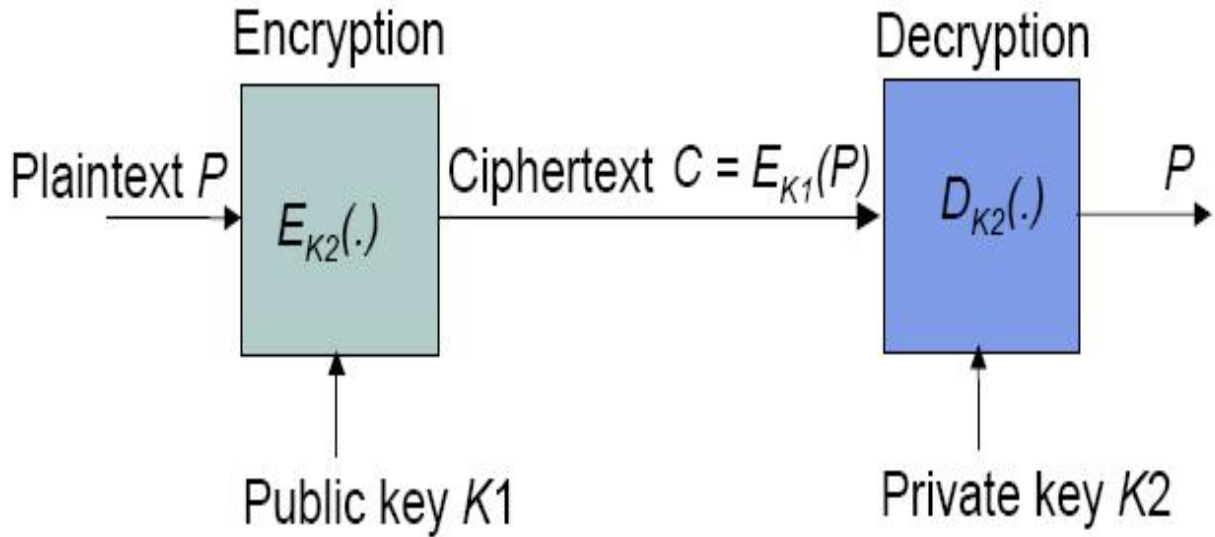
r' is some random number sent by receiver

$E_k(r')$ is encryption of random number r

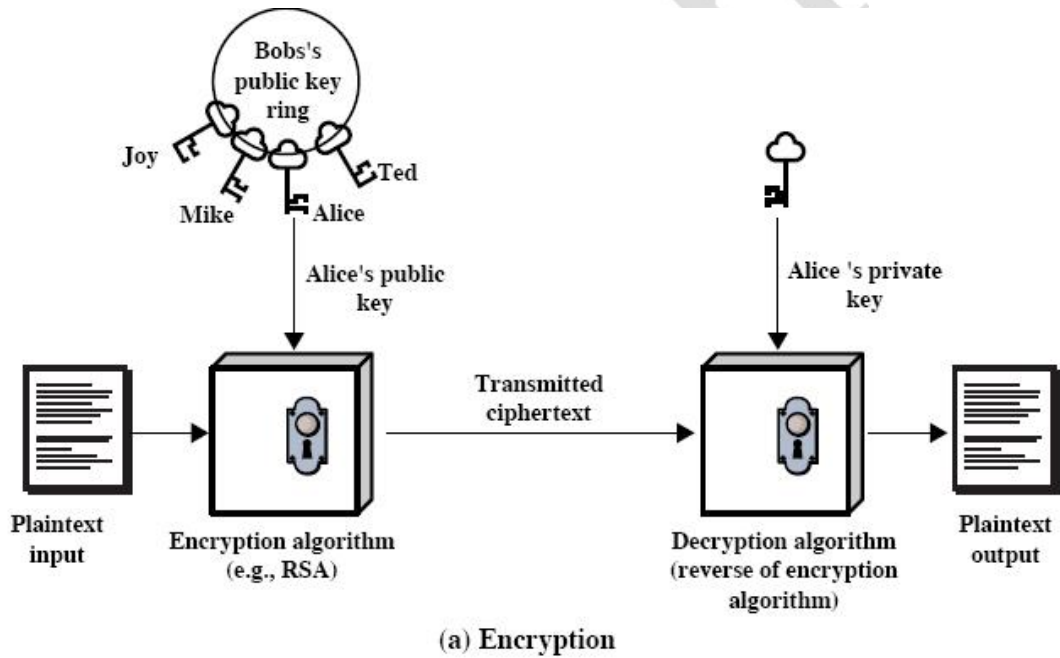
Before two parties can send information securely, they must first exchange a secret key. how can the two parties exchange a key secretly before they can communicate in secret? Even if the sender and receiver found a channel that they believed to be secure, in the past there has been no way to test the secrecy of each key. Quantum cryptography solves this problem. It allows the sender and receiver to test and guarantee the secrecy of each individual key.

Public-key cryptography

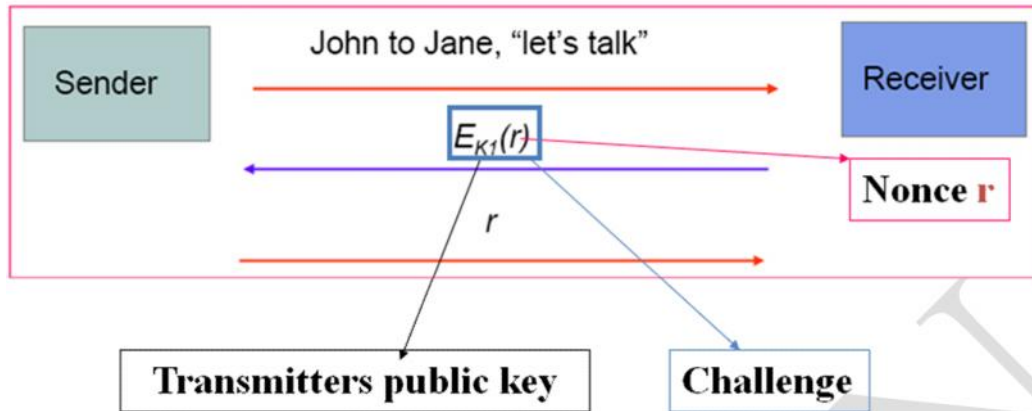
- ❖ A cryptographic system that uses two keys -- a *public key* known to everyone and a *private* or *secret key* known only to the recipient of the message.
- ❖ When John wants to send a secure message to Jane, he uses Jane's public key to encrypt the message. Jane then uses her private key to decrypt it.
- ❖ An important element to the public key system is that the public and private keys are related in such a way that only the public key can be used to encrypt messages and only the corresponding private key can be used to decrypt them. Moreover, it is virtually impossible to deduce the private key if you know the public key.
- ❖ Public key cryptography was invented in 1976 by Whitfield Diffie and Martin Hellman. For this reason, it is sometime called *Diffie-Hellman encryption*.
- ❖ It is also called *asymmetric encryption* because it uses two keys instead of one key (*symmetric encryption*).



One important requirement for public key cryptography is that it must not be possible to determine K_2 from K_1 .
 Example : RSA (Rivest Shamir and Adleman) Algorithm ;



Public key cryptography can also be used for authentication



Public key cryptography developed to address two key issues:

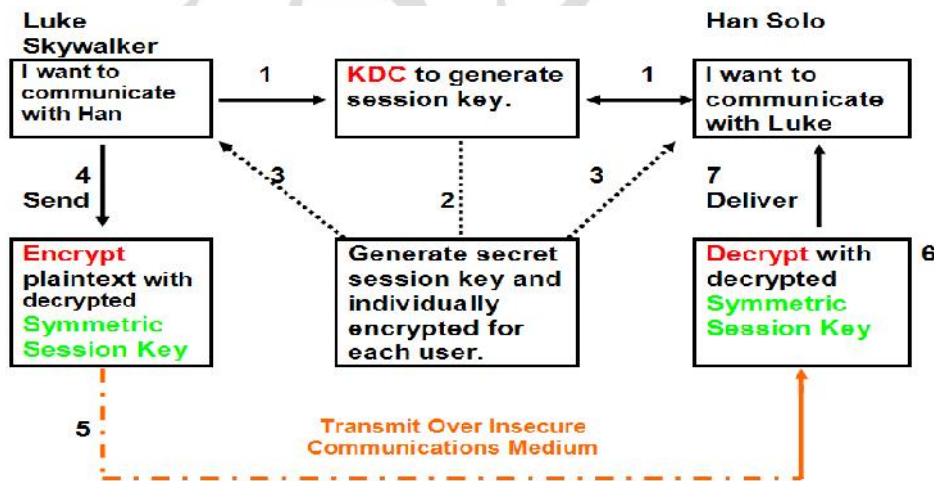
- key distribution – how to have secure communications in general without having to trust a KDC with your key
- digital signatures – how to verify a message comes intact from the claimed sender

Key Distribution center (KDC)

Secret key systems require every pair of users to share a separate key.

Consequently the number of keys grow as the square of the number of users making these systems unfeasible for large scale use.

This problem can be addresses through the introduction of KDC



For example, assume the KDC shares a different secret key with each user. Han Solo contacts the KDC with a request to communicate with Luke Skywalker (1). The KDC creates a symmetric secret session key, encrypts it with the secret key shared with Han (2) and transmits the key to Han (3). Next the KDC encrypts the secret session key with the secret key shared with Luke (2), then transmits the message to Luke (3). Luke and Han now have the secret session key and commence the communication. Note that changing session keys enhances the security of the communications. The message from the KDC to Luke and Han will normally specify the algorithm (including padding scheme, etceteras) in addition to the session key.

Certification Authority (CA)

- Public key systems require only one pair of keys per user, but they still face the problem of how public keys are to be distributed.
- The public keys must be certified somehow.
- One approach to address this problem is to establish a Certification Authority (CA).
- To issue certificates that consist of signed message stating the name of given user, his or her public key, a serial number identifying the certificate and an expiration date

Diffie-Hellman key exchange

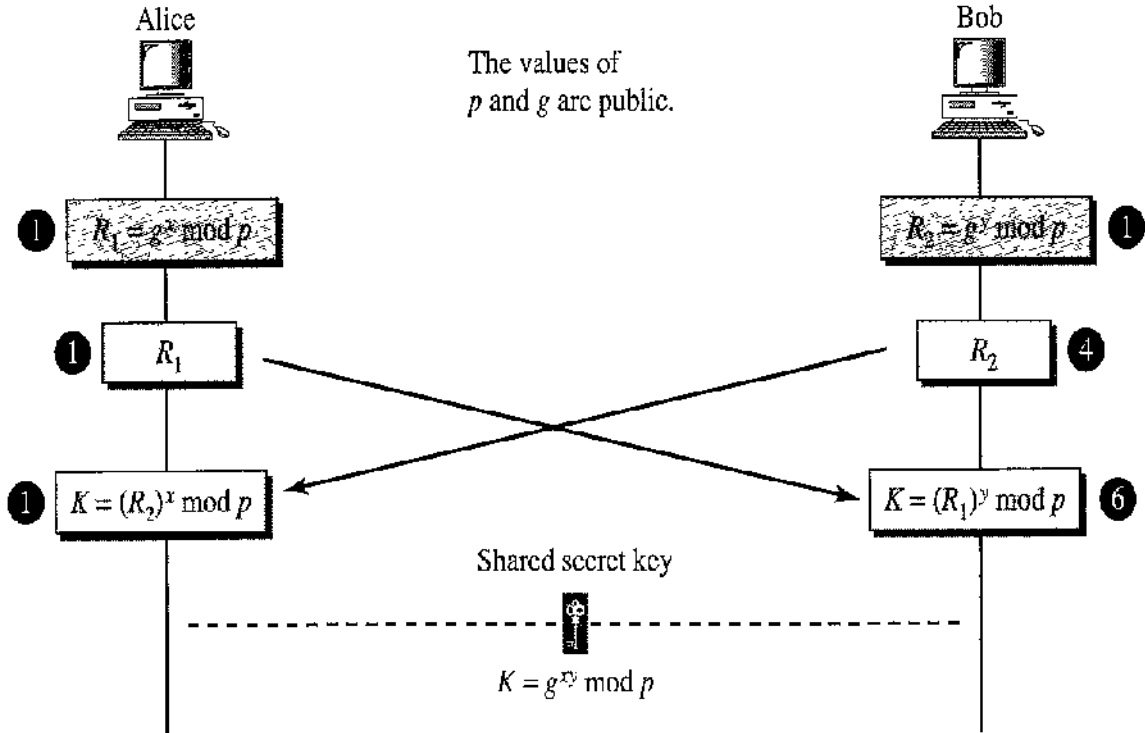
In 1976, Whitfield Diffie and Martin Hellman published their method for public exchange of a secret key. This became known as Diffie-Hellman key exchange, and was the first known way to distribute keys for secure communication.

The exchange method relies on simple mathematical fact: $(x^y)^z = (x^z)^y$

Here's how Diffie-Hellman key exchange works:

- Alice and Bob agree on a long prime number, p , and a base g . The base, g , doesn't need to be large; it is normally 2 or 5.
- Alice creates a long prime number, a , which is her private key. She calculates $A = g^a \text{ mod } p$. She sends A to Bob.
- Bob creates his own long prime number, b , which is his private key. He calculates $B = g^b \text{ mod } p$. He sends B to Alice.
- Alice gets B from Bob, and calculates the shared key: $K = B^a \text{ mod } p$.
- Bob gets A from Alice, and calculates the same shared key: $K = A^b \text{ mod } p$.

- Now, Alice and Bob have calculated respectively $K = (g^a)^b \text{ mod } p$ and $K = (g^b)^a \text{ mod } p$. From the first formula above, we know these are actually the same value: the shared key which Alice and Bob will use to communicate securely.



Example


Let us give a trivial example to make the procedure clear. Our example uses small numbers, but note that in a real situation, the number are very large . Assume $g=7$ and $P=23$

Alice chooses $x = 3$

Bob chooses $y = 6$

$$R_1 = 7^3 \text{ mod } 23 = 21$$

$$R_2 = 7^6 \text{ mod } 23 = 4$$

Alice calculates the symmetric key $K = 4^3 \text{ mod } 23 = 18$  Bob calculates the symmetric key $K = 21^6 \text{ mod } 23 = 18$

The value of K is the same for both Alice and Bob; $g^{xy} \text{ mod } p = 7^{18} \text{ mod } 23 = 18$.

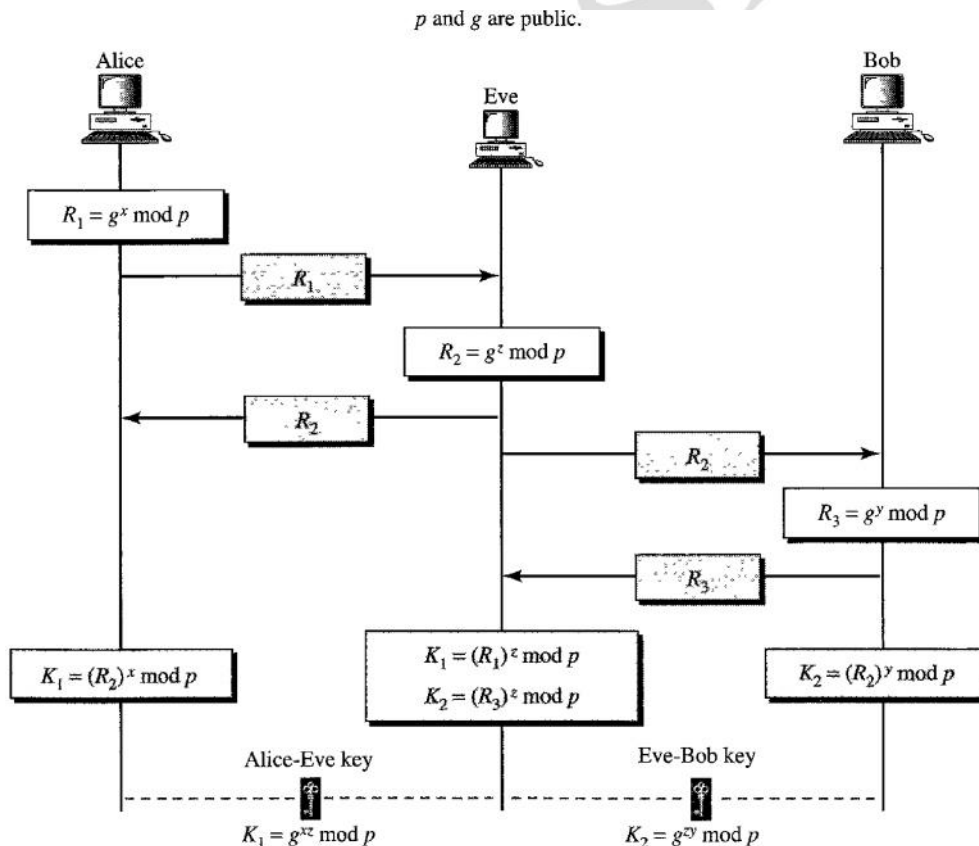
Diffie-Hellman Exchange -Weakness

The required exponentials need many multiplications for large prime numbers p . It could produce a heavy computational burden on a machine and result in Denial Of Service to legitimate Client.

Man in the Middle attack in Diffie-Hellman key exchange

In Below figure

1. Alice choose x , calculates $R_1 = g^x \text{ mod } p$, and sends R_1 to Bob.
2. Eve, the intruder, intercepts R_1 . She chooses Z , calculates $R_2 = g^z \text{ mod } p$ and sends R_2 to both Alice and Bob.
3. Bob chooses y , calculates $R_3 = g^y \text{ mod } p$, and sends R_3 to Alice; R_3 is intercepted by Eve and never reaches Alice.
4. Alice and Eve calculate $K_1 = g^{xz} \text{ mod } p$, which becomes a shared key between Alice and Eve. Alice, however, thinks that it is a key shared between Bob and herself.
5. Eve and Bob calculate $K_2 = g^{zy} \text{ mod } p$, which becomes a shared key between Eve and Bob. Bob, however, thinks that it is a key shared between Alice and himself.

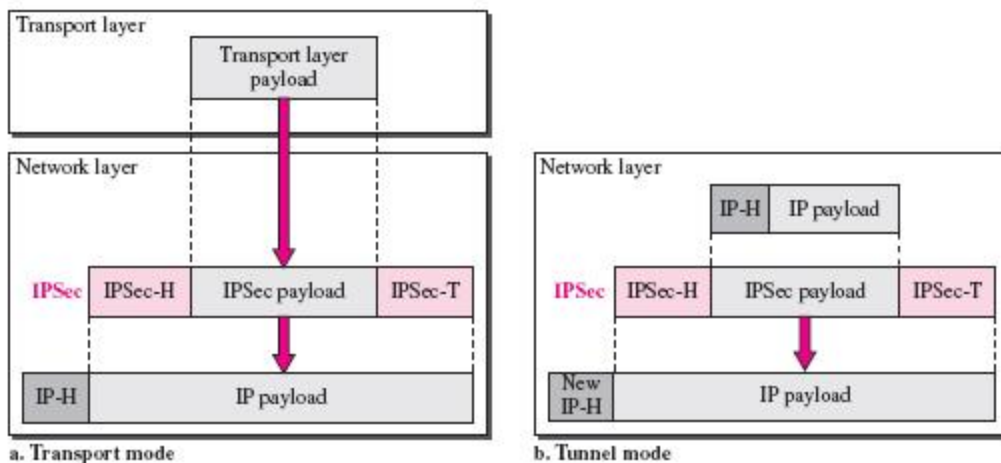


IPsec

- ❖ IPsec (Internet Protocol Security) is a framework for a set of protocols for security at the network or packet processing layer of network communication.
- ❖ Earlier security approaches have inserted security at the application layer of the communications model.
- ❖ IPsec is said to be especially useful for implementing virtual private networks and for remote user access through dial-up connection to private networks.
- ❖ A big advantage of IPsec is that security arrangements can be handled without requiring changes to individual user computers.
- ❖ Cisco has been a leader in proposing IPsec as a standard (or combination of standards and technologies) and has included support for it in its network routers.
- ❖ IPsec provides two choices of security service: Authentication Header (AH), which essentially allows authentication of the sender of data, and Encapsulating Security Payload (ESP), which supports both authentication of the sender and encryption of data as well.
- ❖ The specific information associated with each of these services is inserted into the packet in a header that follows the IP packet header.

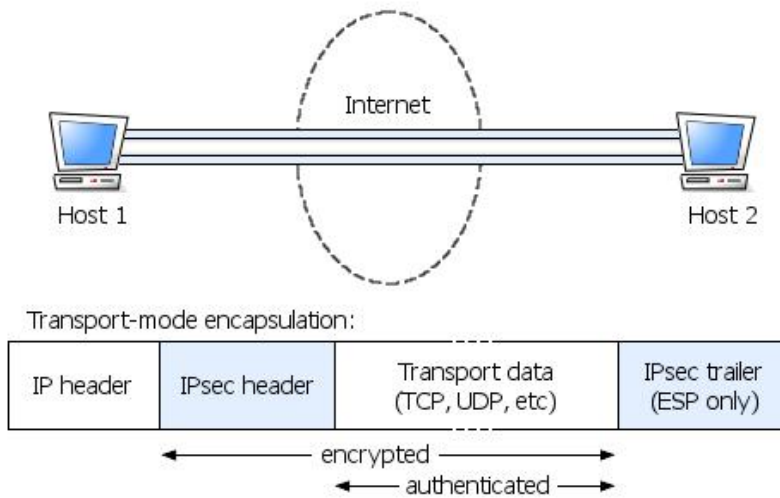
There are two modes of IPsec operation:

Figure 32.3 Transport mode and tunnel modes of IPsec protocol



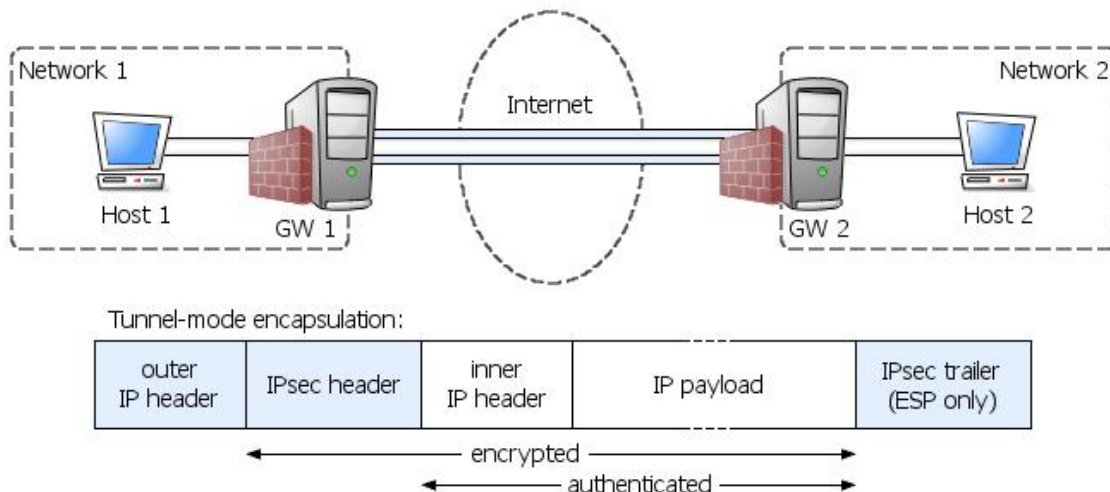
Transport mode

In transport mode, only the payload (the data you transfer) of the IP packet is encrypted and/or authenticated.



Tunnel mode

In tunnel mode, the entire IP packet (data and IP header) is encrypted and/or authenticated.

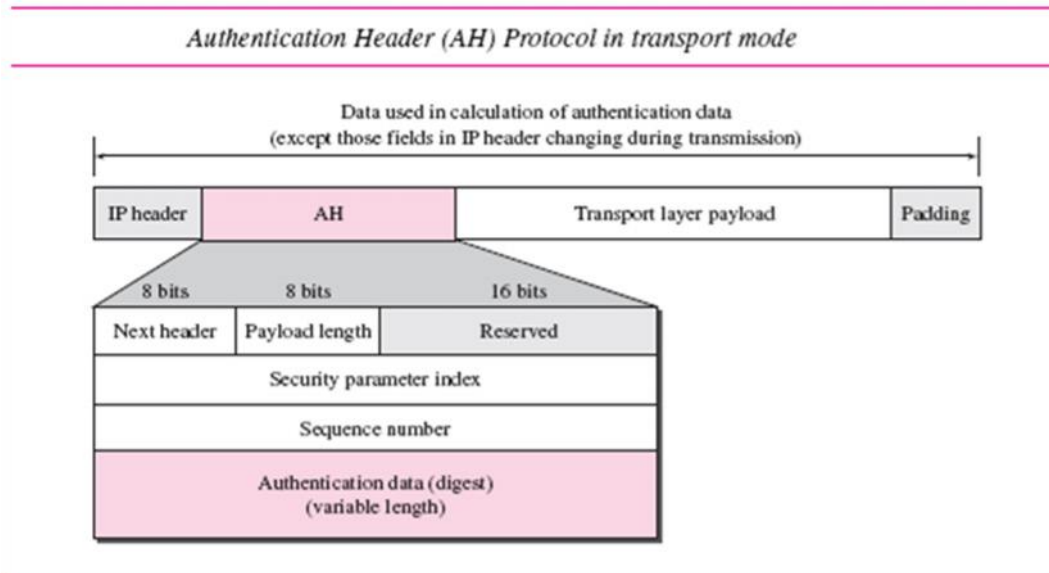


Authentication Header (AH)

- ❖ AH is a member of the IPsec protocol suite.
- ❖ AH is intended to guarantee connectionless integrity and data origin authentication of IP packets.
- ❖ AH protects the IP payload and all header fields of an IP datagram except for mutable fields (i.e. those that might be altered in transit).

- ❖ In IPv4, mutable (and therefore unauthenticated) IP header fields include DSCP/TOS, Flags, Fragment Offset, TTL and Header Checksum.
- ❖ AH operates directly on top of IP, using IP protocol number 51.

The following AH packet diagram shows how an AH packet is constructed and interpreted:



1. An authentication header is added to the payload with the authentication data field set to zero.
2. Padding may be added to make the total length even for a particular hashing algorithm, and it tells which hashing algorithm is used to calculate Authentication data.
3. Hashing is based on the total packet. However, only those fields of the IP header that do not change during transmission are included in the calculation of the message digest (authentication data).
4. The authentication data are inserted in the authentication header.
5. The IP header is added after the value of the protocol field is changed to 51.

A brief description of each field follows:

- **Next header.** The 8-bit next-header field defines the type of payload carried by the IP datagram (such as TCP, UDP, ICMP, or OSPF). It has the same function as the protocol field in the IP header before encapsulation. In other words, the process copies the value of the protocol field in the IP datagram to this field. The value of the protocol field in the new IP datagram is now set to 51 to show that the packet carries an authentication header.
- **Payload length.** The name of this 8-bit field is misleading. It does not define the length of the payload; it defines the length of the authentication header in 4-byte multiples, but it does not include the first 8 bytes.

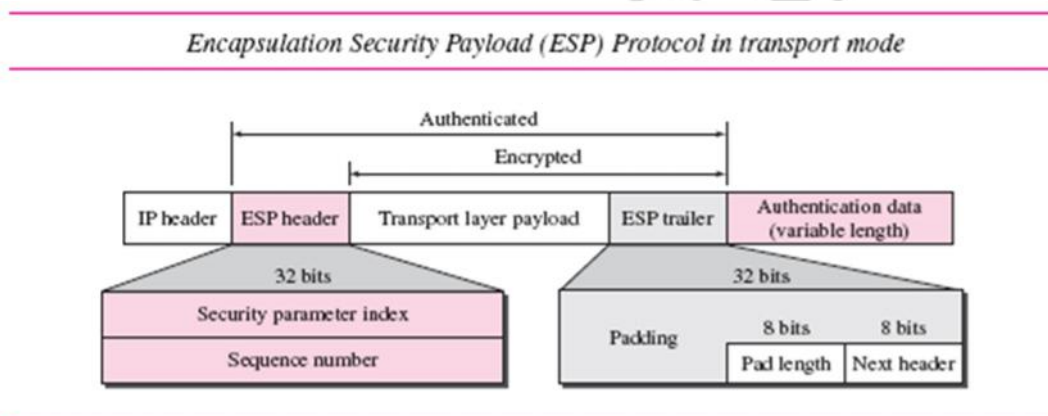
- **Security parameter index.** The 32-bit security parameter index (SPI) field plays the role of a virtual-circuit identifier and is the same for all packets sent during a connection called a security association (discussed later).
- **Sequence number.** A 32-bit sequence number provides ordering information for a sequence of datagrams. The sequence numbers prevent a playback. Note that the sequence number is not repeated even if a packet is retransmitted. A sequence number does not wrap around after it reaches 2^{32} ; a new connection must be established.
- **Authentication data.** Finally, the authentication data field is the result of applying a hash function to the entire IP datagram except for the fields that are changed during transit (e.g., time-to-live).

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

Encapsulating Security Payload (ESP)

- ❖ ESP is a member of the IPsec protocol suite. It is the portion of IPsec that provides origin authenticity, integrity, and confidentiality protection of packets.
- ❖ Unlike Authentication Header (AH), ESP does not protect the IP packet header.

The following ESP packet diagram shows how an ESP packet is constructed and interpreted:



When an IP datagram carries an ESP header and trailer, the value of the protocol field in the IP header is 50. A field inside the ESP trailer (the next-header field) holds the original value of the protocol field (the type of payload being carried by the IP datagram, such as TCP or UDP). The ESP procedure follows these steps:

1. An ESP trailer is added to the payload.
2. The payload and the trailer are encrypted.
3. The ESP header is added.
4. The ESP header, payload, and ESP trailer are used to create the authentication data.
5. The authentication data are added to the end of the ESP trailer.

6. The IP header is added after the protocol value is changed to 50.

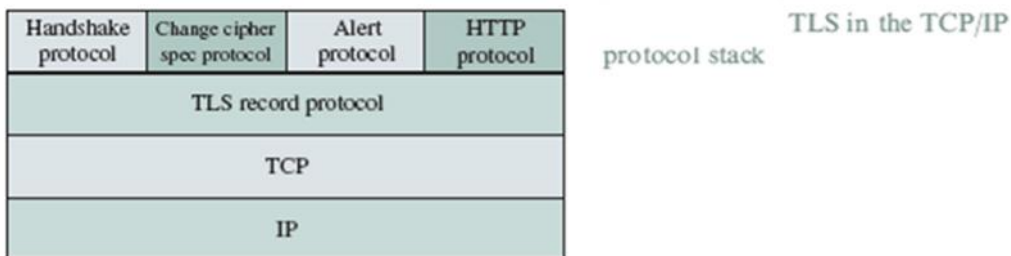
The fields for the header and trailer are as follows:

- **Security parameter index.** The 32-bit security parameter index field is similar to that defined for the AH Protocol.
- **Sequence number.** The 32-bit sequence number field is similar to that defined for the AH Protocol.
- **Padding.** This variable-length field (0 to 255 bytes) of 0s serves as padding.
- **Pad length.** The 8-bit pad length field defines the number of padding bytes. The value is between 0 and 255; the maximum value is rare.
- **Next header.** The 8-bit next-header field is similar to that defined in the AH Protocol.
It serves the same purpose as the protocol field in the IP header before encapsulation.
- **Authentication data.** Finally, the authentication data field is the result of applying an authentication scheme to parts of the datagram. Note the difference between the authentication data in AH and ESP. In AH, part of the IP header is included in the calculation of the authentication data; in ESP, it is not.

ESP provides source authentication, data integrity, and privacy.

SSL / TLS: Transport Layer Security Protocol

- ❖ TLS is based on the Secure Socket Layer (SSL), a protocol originally created by Netscape.
- ❖ One advantage of TLS is that it is application protocol independent.
- ❖ The TLS protocol runs above TCP/IP and below application protocols such as HTTP or IMAP.
- ❖ The HTTP running on top of TLS or SSL is often called HTTPS.
- ❖ Transport Layer Security (TLS) Protocol provides privacy and data integrity between two communicating applications.
- ❖ The protocol is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol.
- ❖ At the lowest level, layered on top of some reliable transport protocol (TCP) is the TLS Record Protocol.



The TLS Record protocol provides a secure connection with the attributes of privacy and reliability. The connection provides *privacy* through the use of symmetric (secret key) encryption. The specific encryption algorithms can be selected from a wide range of standard algorithms. The secret keys that are used in the symmetric encryption algorithms are generated uniquely for each connection and are derived from a secret that is negotiated by another protocol, for example, the TLS Handshake protocol. The TLS Record protocol can operate without

encryption. The connection provides *reliability* through the use of a keyed message authentication code (MAC).⁴ The specific hash function for a connection can be selected from a set of standard hash functions. The TLS Record protocol typically operates without a MAC only while a higher-layer protocol is negotiating the security parameters.

The TLS Handshake protocol, along with the Change Cipher Spec protocol and the Alert protocol is used to negotiate and instantiate the security parameters for the record layers, to authenticate the users, and to report error conditions. The TSL Handshake protocol is used by a server and a client to establish a **session**.

The TLS Record Protocol provides connection security that has two basic properties:

- Private - symmetric cryptography is used for data encryption (DES, RC4 , etc.) The keys for this symmetric encryption are generated uniquely for each connection and are based on a secret negotiated by another protocol (such as the TLS Handshake Protocol). The Record Protocol can also be used without encryption.
- Reliable - message transport includes a message integrity check using a keyed MAC. Secure hash functions (SHA, MD5, etc.) are used for MAC computations. The Record Protocol can operate without a MAC, but is generally only used in this mode while another protocol is using the Record Protocol as a transport for negotiating security parameters.

The TLS Record Protocol is used for encapsulation of various higher level protocols.

One such encapsulated protocol, the TLS Handshake Protocol, allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data.

The TLS Handshake Protocol provides connection security that has three basic properties:

- The peer's identity can be authenticated using asymmetric or public key, cryptography (RSA, DSS, etc.). This authentication can be made optional, but is generally required for at least one of the peers.
- The negotiation of a shared secret is secure: the negotiated secret is unavailable to eavesdroppers, and for any authenticated connection the secret cannot be obtained, even by an attacker who can place himself in the middle of the connection

- The negotiation is reliable: no attacker can modify the negotiation communication without being detected by the parties to the communication.

Protocol Structure - TLS: Transport Layer Security Protocol

TLS protocol includes two protocol groups: TLS Record Protocol and TLS Handshake protocols, which have many messages with different formats. We only summarized the protocols here without details, which could be found in the reference documents.

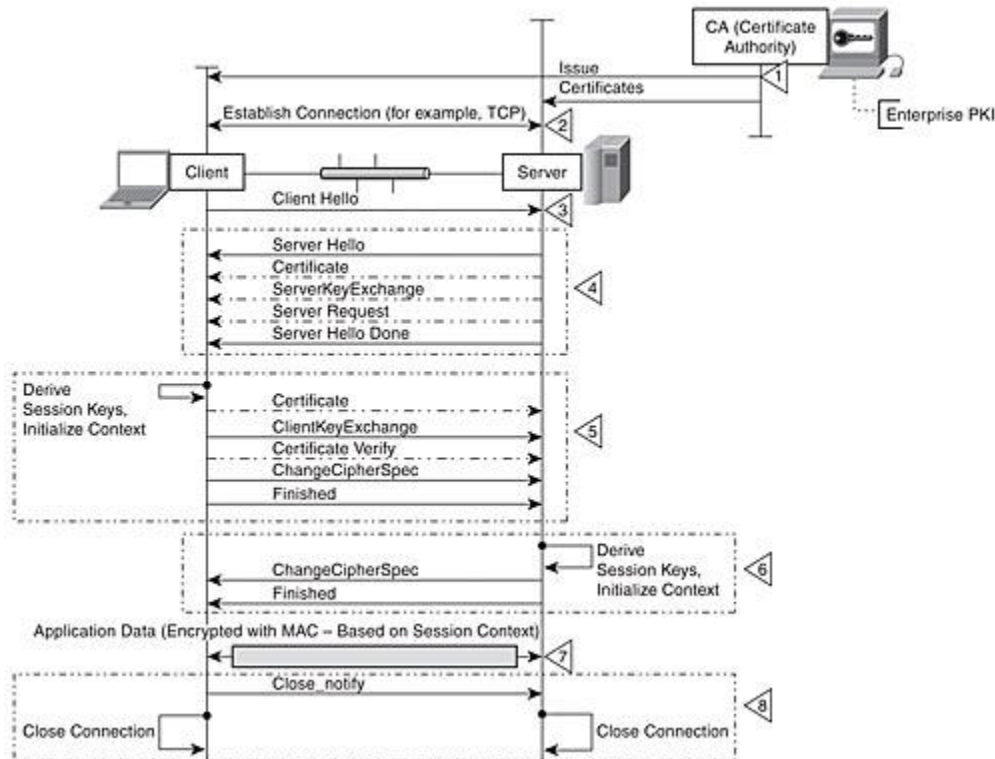
TLS Record Protocol: a layered protocol. At each layer, messages may include fields for length, description, and content. The Record Protocol takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result. Received data is decrypted, verified, decompressed, and reassembled, then delivered to higher level clients. Here are the layers:

- TLS connection state: is the operating environment of the TLS Record Protocol. It specifies a compression algorithm, encryption algorithm, and MAC algorithm. Connection states.
- TLS Record Layer: receives uninterpreted data from higher layers in non-empty blocks of arbitrary size.
- Key calculation: The Record Protocol requires an algorithm to generate keys, IVs, and MAC secrets from the security parameters provided by the handshake protocol.

TLS Handshake Protocol: consists of a suite of three sub-protocols which are used to allow peers to agree upon security parameters for the record layer, authenticate themselves, instantiate negotiated security parameters, and report error conditions to each other.

- Change cipher spec protocol
- Alert protocol
- Handshake protocol

TLS handshake process



- *ClientHello* - The client asks the server to establish a communication by sending, together with that information, the version number of the supported SSL, random number, current date time and the information on the private key encryption algorithms supported by the client.
- *ServerHello* - The server sends to the client the identification number of the SSL protocol version supported and the settings of the private key encryption algorithms in use.
- The client proceeds with the authentication of the server by examining the provided certificate, checking that the CA that it was undersigned with appears in the the list of trusted CAs.
- The server requests the certificate to the client for the authentication.
- The client sends the certificate to the server. If the server is not able to authenticate it, then an encrypted SSL connection cannot be established, instead if the authentication is successful we move on to the next phase.
- *ClientKeyExchange* - The client creates a premaster secret (session key) that can be used only for the present exchange of information and data, it is encrypted with the server's public key (contained in the server's certificate) and it sends the encrypted session key to the server.
- If the server has requested authentication to the client (optional step) the clients sends part of the data in this session and digitally signs this data and sends it's certificate together with the encrypted session key.

- *ChangeCipherSpec* - Client and Server communicate to each other that the data that will be exchanged in the next phase will be encrypted with the session key previously exchanged.
- *Finished* - The server sends an encrypted message indicating, on its behalf, the end of the handshake session, the client consequently responds. The handshake phase ends and the real SSL session begins. The client and the server use the session key to encrypt and decrypt the data that they mutually exchange to validate the integrity.

The TLS handshake process is shown in Figure

Step 1: The client and server exchange hello messages to negotiate algorithms, exchange random values, and initiate or resume the session.

the client sends a ClientHello message to request a connection. The ClientHello message includes the client version of the TLS protocol; the current time and date in standard UNIX 32-bit format and a 28-byte random value; an optional session ID (if not empty, this value identifies the session whose security parameters the client wishes to reuse, this field empty if no session ID available or if new security parameters are sought); a CipherSuite list that contains the combinations of key exchange, bulk encryption, and MAC algorithms that are supported by the client; and a list of compression algorithms supported by the client.

The server examines the contents of the ClientHello message. The server sends a ServerHello message if it finds an acceptable set of algorithms in the lists proposed by the client. If the server cannot find a match, it sends a handshake failure alert message and closes the connection. The ServerHello message contains the following parameters: the server version, a random value that is different from and independent of the value sent by the client, a session ID, a CipherSuite selected from the list proposed by the client, and a compression algorithm from the list proposed by the client.

Step 2: The client and server exchange cryptographic parameters to allow them to agree on a premaster secret. If necessary, they exchange certificates and cryptographic information to authenticate each other. They then generate a master secret from the premaster secret and exchange random values.

after the ServerHello message the server may also send the following three messages. The Certificate message is sent if the server needs to be authenticated. The message generally includes an X509.v3 certificate that contains a key for the corresponding key exchange method. The ServerKeyExchange is sent immediately after the Certificate message and is required when the server certificate message does not contain enough data to allow the client to exchange a premaster secret, as for example in a Diffie-Hellman exchange. The

CertificateRequest message is sent by the server if the client is required to be authenticated. Finally, the server sends the ServerHelloDone message, indicating that it is done sending messages to support the key exchange. The server then waits for the client's response.

The client examines the messages from the server and prepares appropriate responses. If required to, the client sends a certificate message with a suitable certificate. If it has no suitable certificate, the client may reply with a certificate message containing no certificate, but the server may then respond with a fatal handshake failure alert message that results in a termination of the connection. The client may follow the certificate message with a ClientKeyExchange message that provides information to set the premaster secret. The CertificateVerify message is sent by a client after it has sent a certificate message. The purpose of the message is to explicitly verify the client certificate. The client prepares a digital signature of the sequence of messages that have been exchanged from the client hello up to but not including this message. The client uses its private key to prepare the signature. This step allows the server to verify that the client owns the private key for the client certificate.

Step 3. The client and server provide their record layer with the security parameters. The client and server verify that their peer has calculated the same security parameters and that the handshake occurred without tampering by an attacker.

The ChangeCipherSpec message is part of the Change Cipher protocol that signals changes in the ciphering strategy. The protocol consists of a single message that is encrypted and compressed according to the current connection state. When sent by the client, the ChangeCipherSpec message notifies the server that subsequent records will be protected under a new CipherSpec and keys. After the client sends the ChangeCipherSpec message in Figure 11.18, the server copies its pending CipherSpec into the current CipherSpec. The client follows immediately with a finished message, which is prepared using the new CipherSpec algorithms. The finished message allows the server to verify that the key exchange and authentications have been successful.

The server responds with ChangeCipherSpec and finished messages of its own. Once the client and server have validated the finished messages they receive, they finally can begin to exchange information over the connection.

Cryptographic Algorithms

Two specific cryptographic algorithms are the **Data Encryption Standard (DES)** and **RSA (Rivest, Shamir, and Adleman)**.

DES

- ❖ DES is a 64-bit block cipher. It is widely used as a shared key cryptographic system. A top-level block diagram of the algorithm's internals is shown below:
- ❖ Data Encryption Standard (DES) is a widely-used method of data encryption using a private (secret) key
- ❖ For each given message, the key is chosen at random from among this enormous number of keys. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key.

DES Algorithm

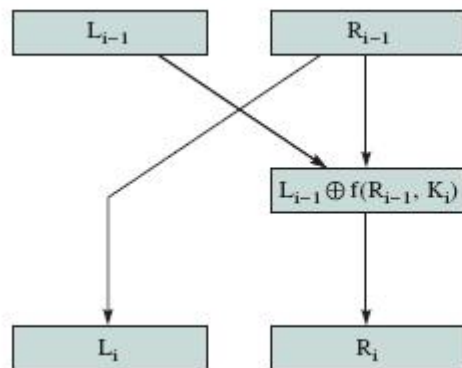
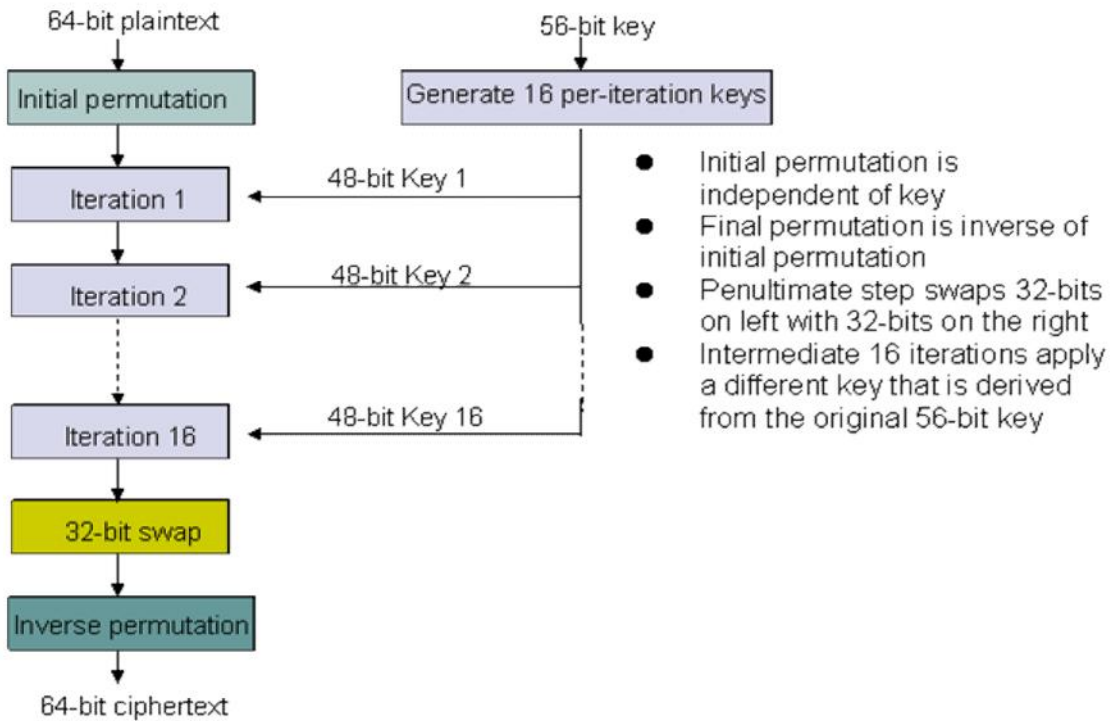


FIGURE 11.20 Each iteration in DES

OR

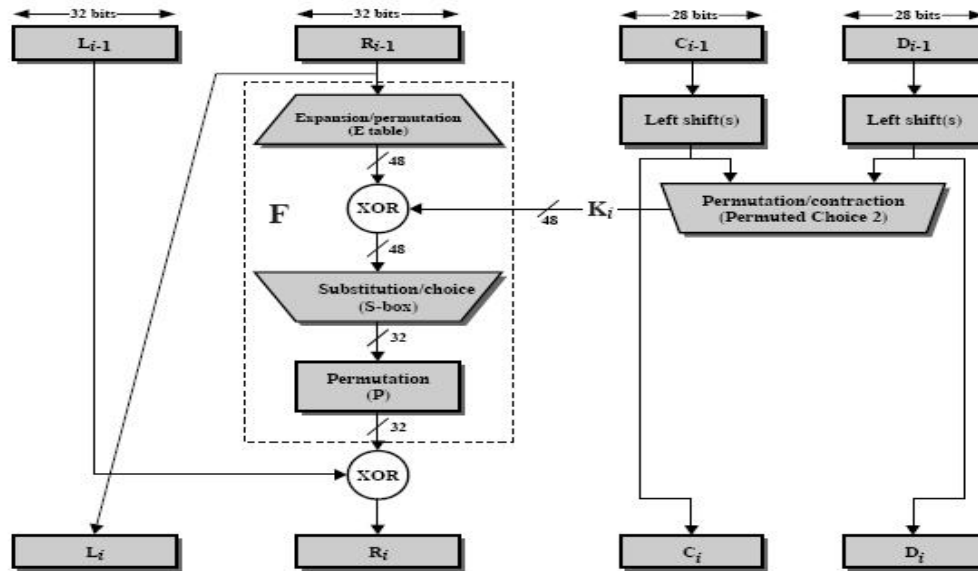
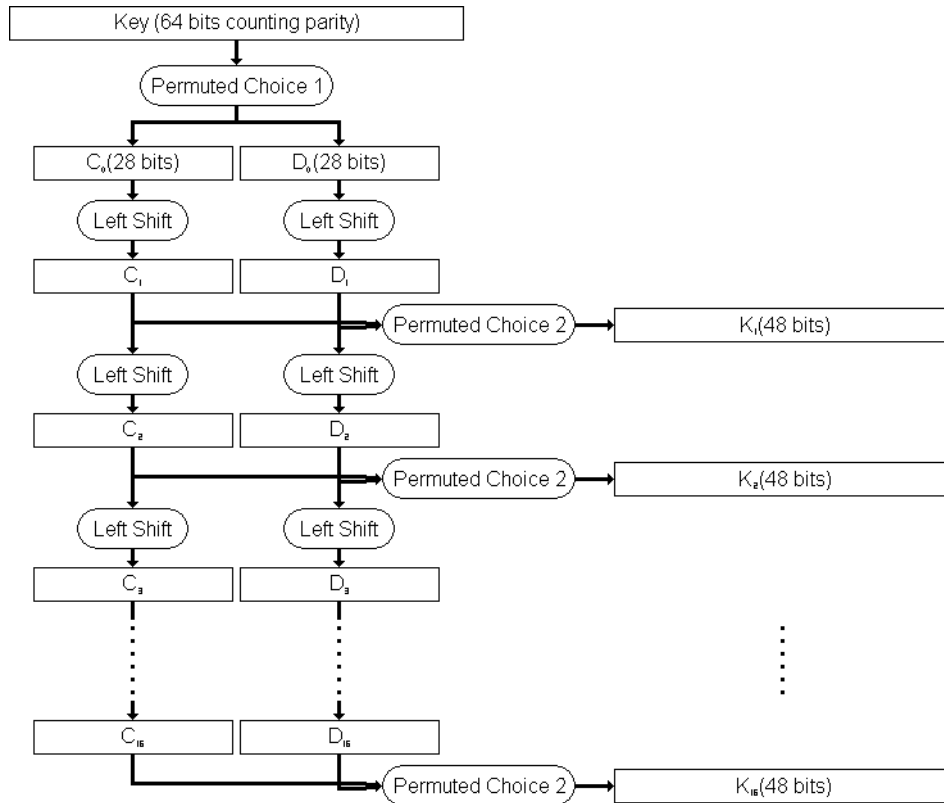


Figure 3.8 Single Round of DES Algorithm

- ❖ DES applies a 56-bit key to each 64-bit block of data.
- ❖ The process can run in several modes and involves 16 rounds or operations.
- ❖ Although this is considered "strong" encryption, many companies use "triple DES", which applies three keys in succession.
- ❖ DES originated at IBM in 1977 and was adopted by the U.S. Department of Defense
- ❖ DES algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bitstring of the same length.
- ❖ In case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt.
- ❖ there are 16 identical stages of processing, termed *rounds*.
- ❖ There is also an initial and final permutation, termed *IP* and *FP*, which are inverses (*IP* "undoes" the action of *FP*, and vice versa).
- ❖ *IP* and *FP* have almost no cryptographic significance, but were apparently included in order to facilitate loading blocks in and out of mid-1970s hardware, as well as to make DES run slower in software.

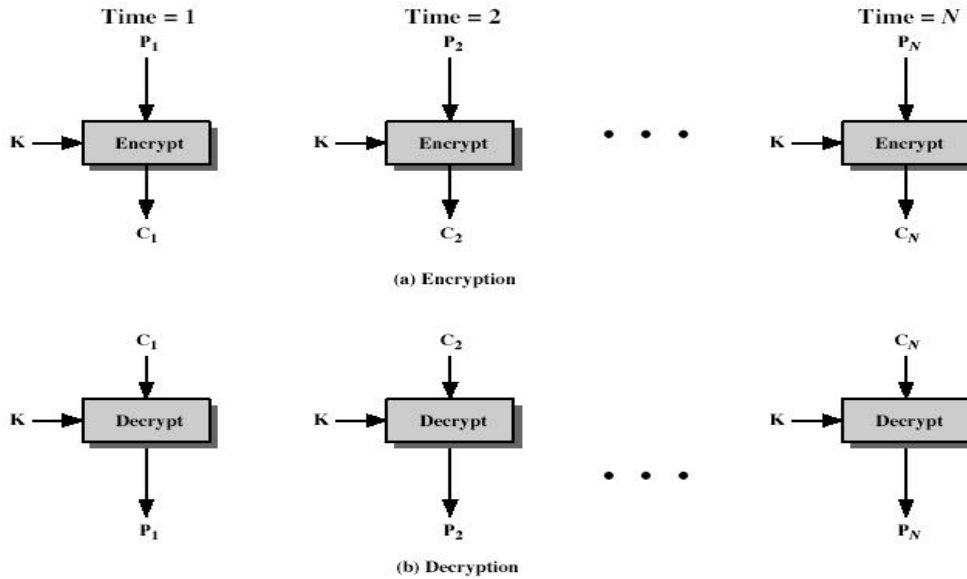
Subkey Generation

To generate the subkeys, start with the 56-bit key (64 bits if you include the parity bits). These are permuted and divided into two halves called C and D. For each round, C and D are each shifted left circularly one or two bits (the number of bits depending on the round). The 48-bit subkey is then selected from the current C and D bits.



Electronic Code Book (ECB)

- Message is broken into independent blocks which are encrypted
- each block is a value which is substituted, like a codebook, hence name
- each block is encoded independently of the other blocks
- $C_i = DES_{K1}(P_i)$
- DES algorithm make use of ECB
- uses: secure transmission of single values

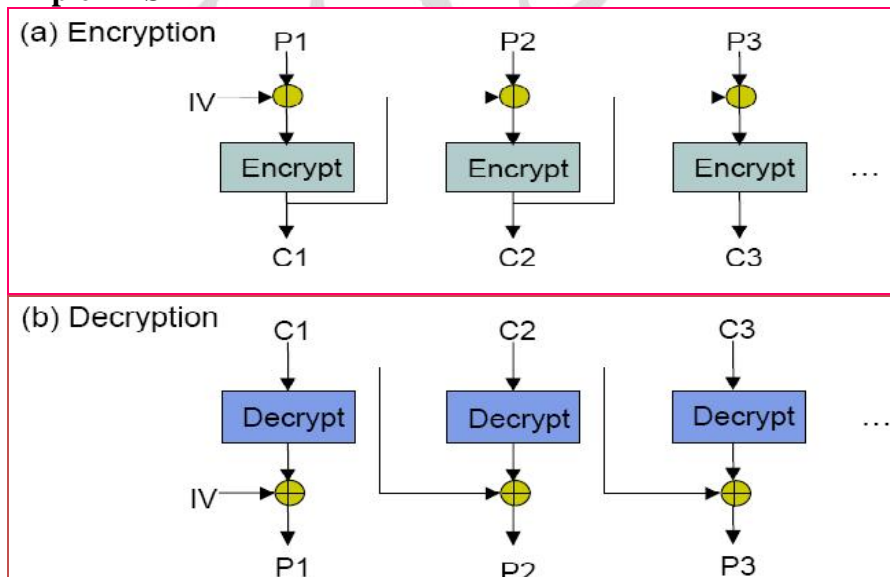


If hacker knows the structure of the algorithm then he can break the algorithm to overcome above problem introduced Cipher Block Chaining

Cipher Block Chaining (CBC)

- Message is broken into blocks
- linked together in encryption operation
- each previous cipher blocks is chained with current plaintext block, hence name
- use Initial Vector (IV) to start process
- $C_i = DES_{K1}(P_i \text{ XOR } C_{i-1})$
- $C_{-1} = IV$
- uses: bulk data encryption, authentication

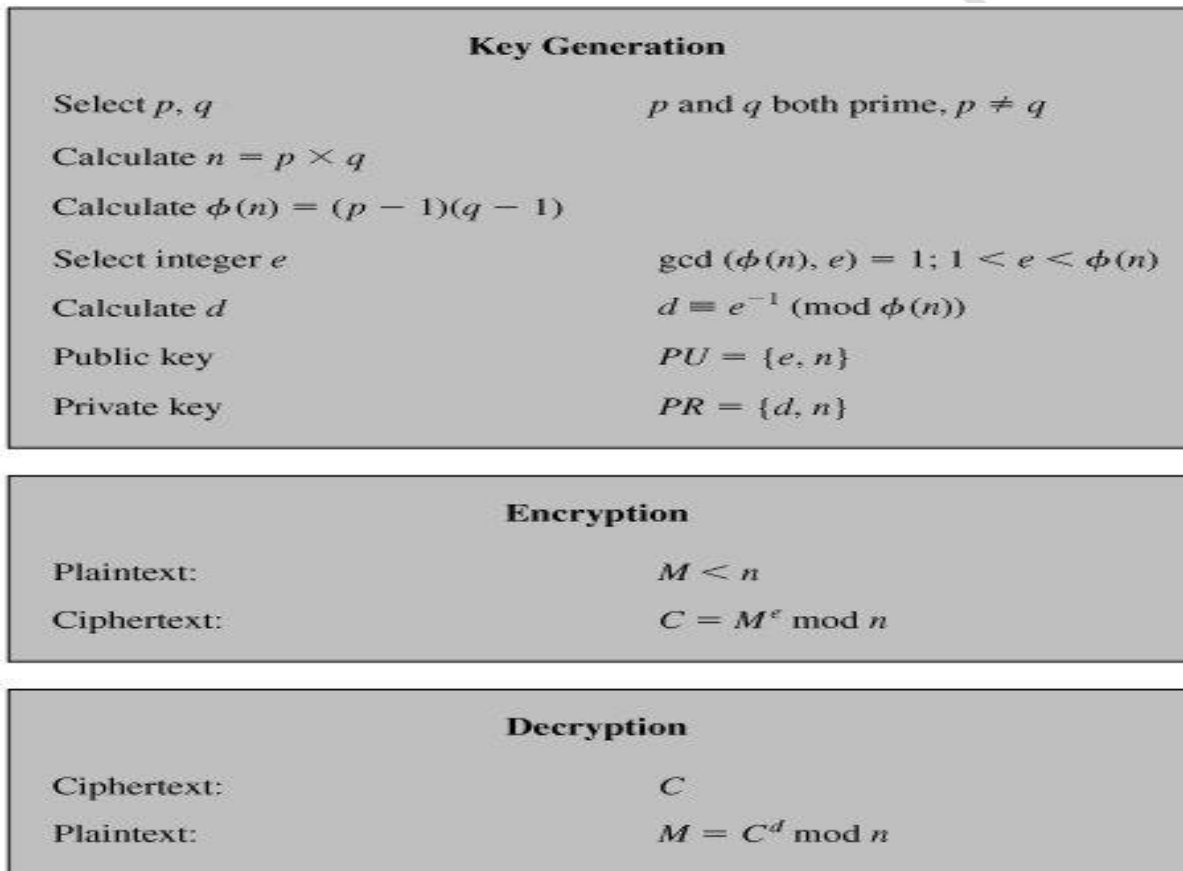
Triple DES



RSA Algorithm

- ❖ The RSA algorithm is named after Ron Rivest, Adi Shamir and Len Adleman,
- ❖ The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers.
- ❖ The RSA scheme is a block Cipher in which the plaintext and ciphertext are integers between 0 and $n-1$ for some n .
- ❖ RSA makes use of an expression with exponentials.
- ❖ Plaintext is encrypted in blocks, with each block having a binary values less than some number n .

Key Generation Algorithm



- n is known as the *modulus*.
- e is known as the *public exponent* or *encryption exponent* or just the *exponent*.
- d is known as the *secret exponent* or *decryption exponent*.

Encryption

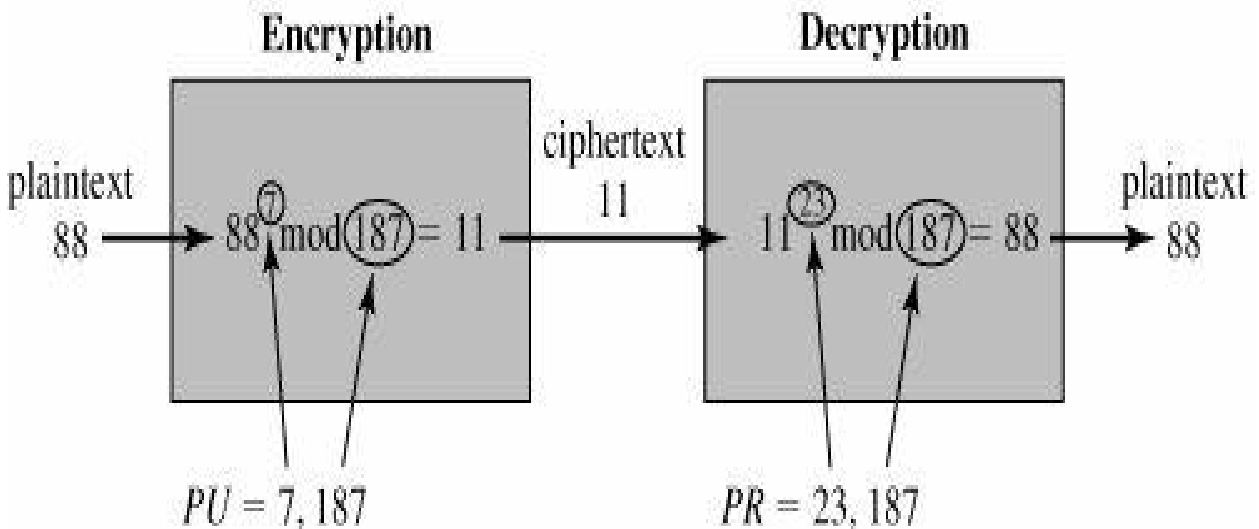
Sender A does the following:-

1. Obtains the recipient B's public key (n, e) .
2. Represents the plaintext message as a positive integer m
3. Computes the ciphertext $c = m^e \bmod n$.
4. Sends the ciphertext c to B

Decryption

Recipient B does the following:-

1. Uses his private key (n, d) to compute $m = c^d \bmod n$.
2. Extracts the plaintext from the message representative m .



Example 1

Key Generation

1) Generate two large prime numbers, p and q

To make the example easy to follow I am going to use small numbers, but this is not secure. To find random primes, we start at a random number and go up ascending odd numbers until we find a prime. Lets have:

$$p = 7$$

$$q = 19$$

2) Let $n = pq$

$$\begin{aligned} n &= 7 * 19 \\ &= 133 \end{aligned}$$

3) Let $m = (p - 1)(q - 1)$

$$\begin{aligned} m &= (7 - 1)(19 - 1) \\ &= 6 * 18 \\ &= 108 \end{aligned}$$

4) Choose a small number, e coprime to m

e coprime to m , means that the largest number that can exactly divide both e and m (their greatest common divisor, or GCD) is 1. Euclid's algorithm is used to find the GCD of two numbers, but the details are omitted here.

$$\begin{aligned} e = 2 &\Rightarrow \text{GCD}(e, 108) = 2 \text{ (no)} \\ e = 3 &\Rightarrow \text{GCD}(e, 108) = 3 \text{ (no)} \\ e = 4 &\Rightarrow \text{GCD}(e, 108) = 4 \text{ (no)} \\ e = 5 &\Rightarrow \text{GCD}(e, 108) = 1 \text{ (yes!)} \end{aligned}$$

5) Find d , such that $de \% m = 1$

This is equivalent to finding d which satisfies $de = 1 + nm$ where n is any integer. We can rewrite this as $d = (1 + nm) / e$. Now we work through values of n until an integer solution for e is found:

$$\begin{aligned} n = 0 &\Rightarrow d = 1 / 5 \text{ (no)} \\ n = 1 &\Rightarrow d = 109 / 5 \text{ (no)} \\ n = 2 &\Rightarrow d = 217 / 5 \text{ (no)} \\ n = 3 &\Rightarrow d = 325 / 5 \\ &= 65 \text{ (yes!)} \end{aligned}$$

To do this with big numbers, a more sophisticated algorithm called extended Euclid must be used.

Public Key Secret Key

$$\begin{array}{ll} n = 133 & n = 133 \\ e = 5 & d = 65 \end{array}$$

Communication

Encryption

The message must be a number less than the smaller of p and q. However, at this point we don't know p or q, so in practice a lower bound on p and q must be published. This can be somewhat below their true value and so isn't a major security concern. For this example, let's use the message "6".

$$\begin{aligned} C &= P^e \% n \\ &= 6^5 \% 133 \\ &= 7776 \% 133 \\ &= 62 \end{aligned}$$

Decryption

This works very much like encryption, but involves a larger exponentiation, which is broken down into several steps.

$$\begin{aligned} P &= C^d \% n \\ &= 62^{65} \% 133 \\ &= 62 * 62^{64} \% 133 \\ &= 62 * (62^2)^{32} \% 133 \\ &= 62 * 3844^{32} \% 133 \\ &= 62 * (3844 \% 133)^{32} \% 133 \\ &= 62 * 120^{32} \% 133 \end{aligned}$$

We now repeat the sequence of operations that reduced 62^{65} to 120^{32} to reduce the exponent down to 1.

$$\begin{aligned} &= 62 * 36^{16} \% 133 \\ &= 62 * 99^8 \% 133 \\ &= 62 * 92^4 \% 133 \\ &= 62 * 85^2 \% 133 \\ &= 62 * 43 \% 133 \\ &= 2666 \% 133 \\ &= 6 \end{aligned}$$

A very simple example of RSA encryption

1. Select primes $p=11$, $q=3$.
2. $n = pq = 11 \cdot 3 = 33$
 $\phi = (p-1)(q-1) = 10 \cdot 2 = 20$
3. Choose $e=3$
 Check $\gcd(e, p-1) = \gcd(3, 10) = 1$ (i.e. 3 and 10 have no common factors except 1),
 and check $\gcd(e, q-1) = \gcd(3, 2) = 1$
 therefore $\gcd(e, \phi) = \gcd(e, (p-1)(q-1)) = \gcd(3, 20) = 1$

4. Compute d such that $ed \equiv 1 \pmod{\phi}$
 i.e. compute $d = e^{-1} \pmod{\phi} = 3^{-1} \pmod{20}$
 i.e. find a value for d such that ϕ divides $(ed-1)$
 i.e. find d such that 20 divides $3d-1$.
 Simple testing ($d = 1, 2, \dots$) gives $d = 7$
 Check: $ed-1 = 3 \cdot 7 - 1 = 20$, which is divisible by ϕ .
5. Public key = $(n, e) = (33, 3)$
 Private key = $(n, d) = (33, 7)$.

This is actually the smallest possible value for the modulus n for which the RSA algorithm works.

Now say we want to encrypt the message $m = 7$,
 $c = m^e \pmod{n} = 7^3 \pmod{33} = 343 \pmod{33} = 13$.
 Hence the ciphertext $c = 13$.

To check decryption we compute
 $m' = c^d \pmod{n} = 13^7 \pmod{33} = 7$.

Note that we don't have to calculate the full value of 13 to the power 7 here. We can make use of the fact that

$$a = bc \pmod{n} = (b \pmod{n}) \cdot (c \pmod{n}) \pmod{n}$$

so we can break down a potentially large number into its components and combine the results of easier, smaller calculations to calculate the final value.

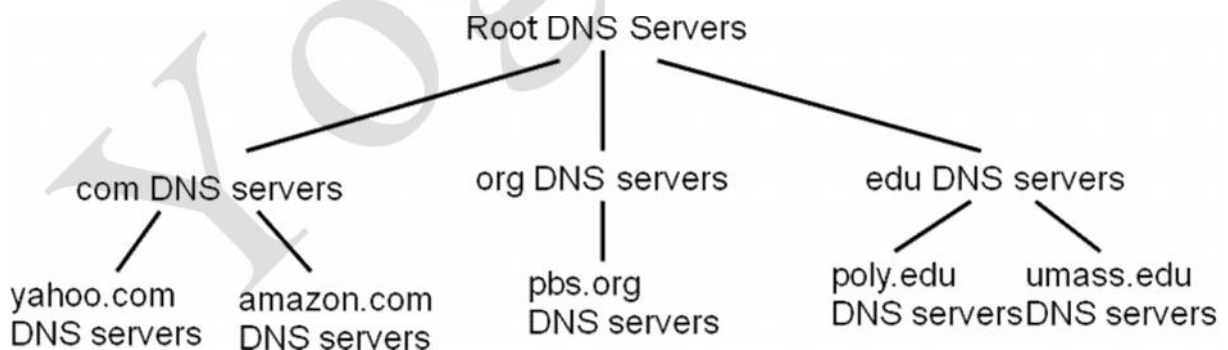
One way of calculating m' is as follows:-

$$\begin{aligned} m' &= 13^7 \pmod{33} = 13^{(3+3+1)} \pmod{33} = 13^3 \cdot 13^3 \cdot 13 \pmod{33} \\ &= (13^3 \pmod{33}) \cdot (13^3 \pmod{33}) \cdot (13 \pmod{33}) \pmod{33} \\ &= (2197 \pmod{33}) \cdot (2197 \pmod{33}) \cdot (13 \pmod{33}) \pmod{33} \\ &= 19 \cdot 19 \cdot 13 \pmod{33} = 4693 \pmod{33} \\ &= 7. \end{aligned}$$

APPLICATIONS:- Domain Name Service (DNS)

- DNS is a host name to IP address translation service
- DNS is a distributed database implemented in a hierarchy of name servers, an application level protocol for message exchange between clients and servers
- A lookup mechanism for translating objects into other objects
- A globally distributed, loosely coherent, scalable, reliable, dynamic database
- Comprised of three components
 - A “name space”
 - Servers making that name space available
 - Resolvers (clients) which query the servers about the name space
- It is easier to remember a host name than it is to remember an IP address.
- A name has more meaning to a user than a 4 byte number.
- Applications such as FTP, HTTP, email, etc., all require the user to input a destination.
- The user generally enters a host name.
- The application takes the host name supplied by the user and forwards it to DNS for translation to an IP address.
- DNS works by exchanging messages between client and server machines.
- A client application will pass the destination host name to the DNS process to get the IP address.
- The application then sits and waits for the response to return.

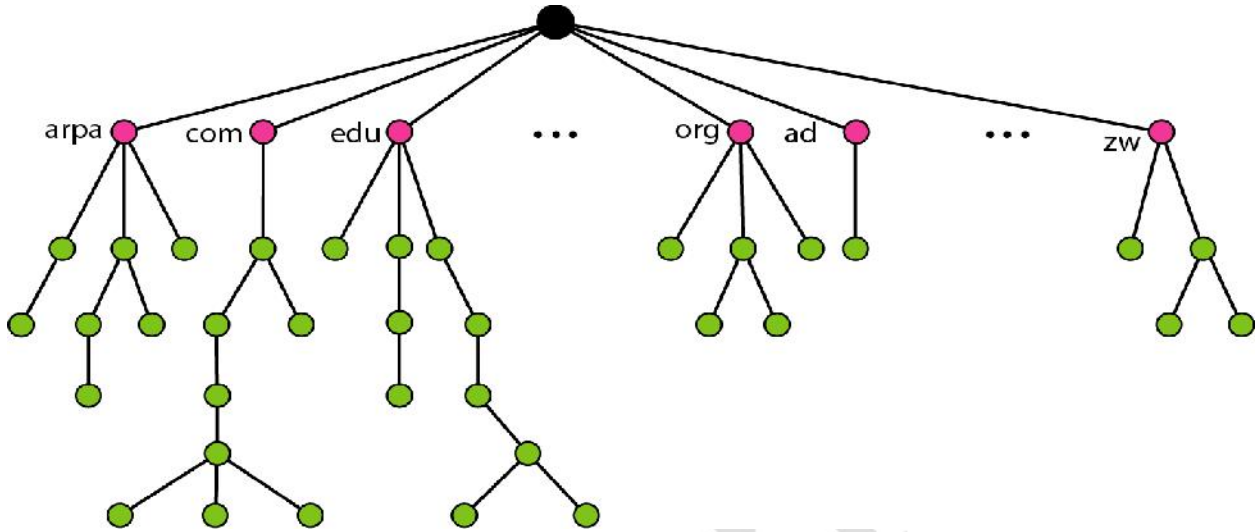
How DNS Works



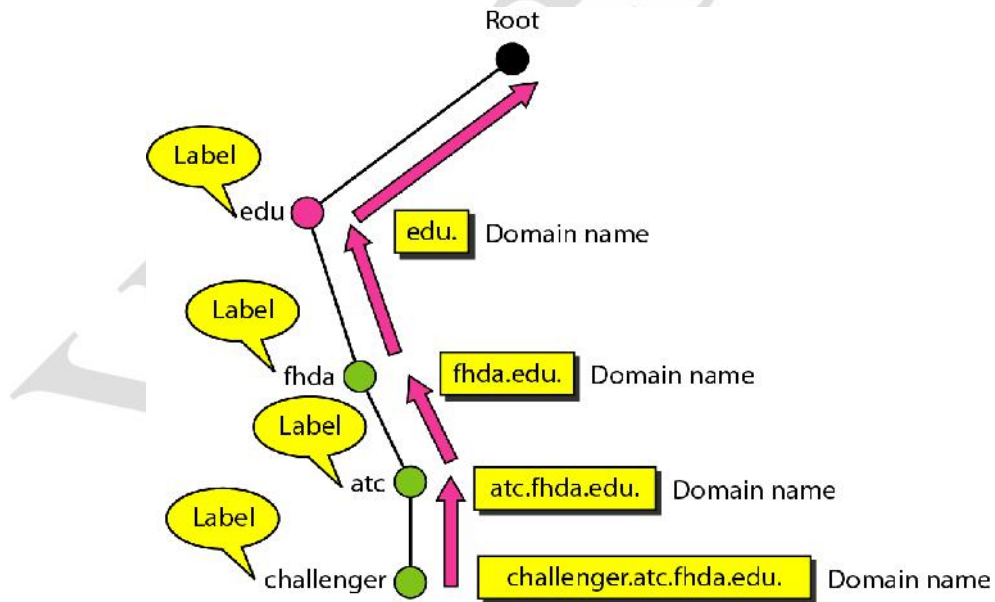
- Client wants IP for www.amazon.com; 1st approx:
- client queries a root server to find com DNS server
- client queries com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

Domain Name Space

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127.

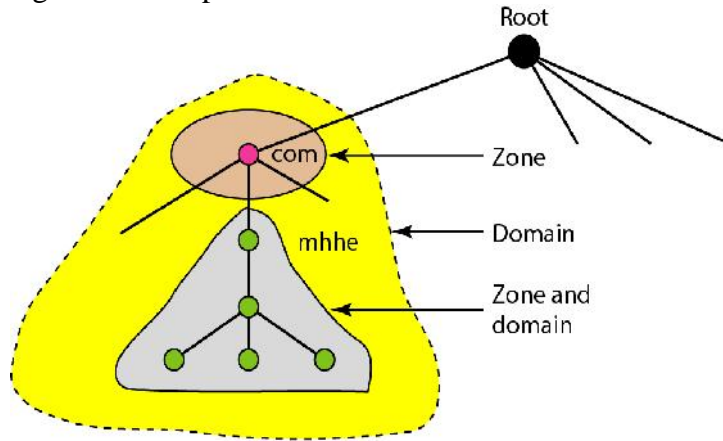


Domain names and Labels

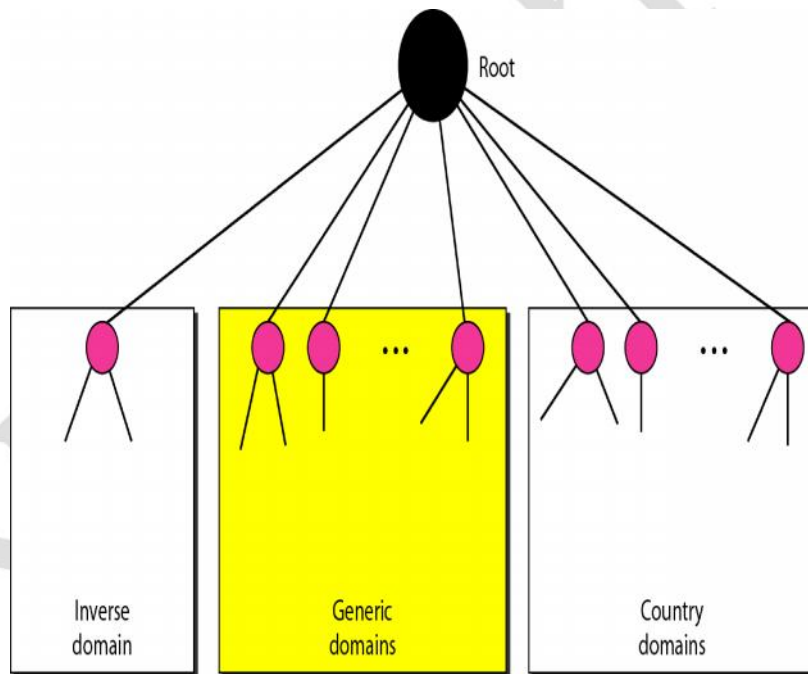


Zones and Domains

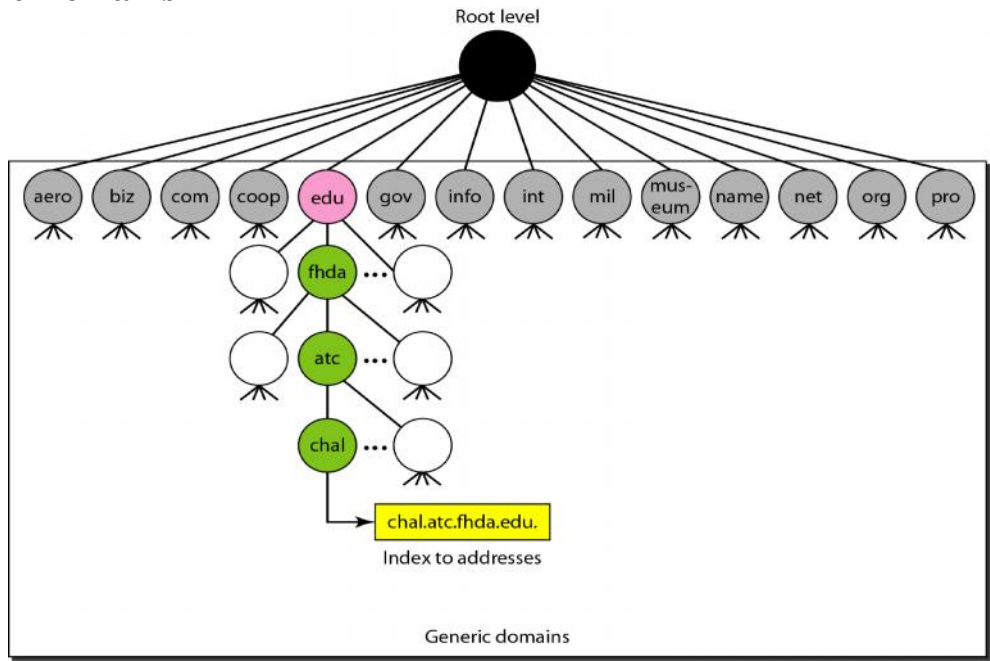
- Zones are “administrative spaces”
- Zone administrators are responsible for portion of a domain’s name space
- Authority is delegated from a parent and to a child



DNS in the Internet



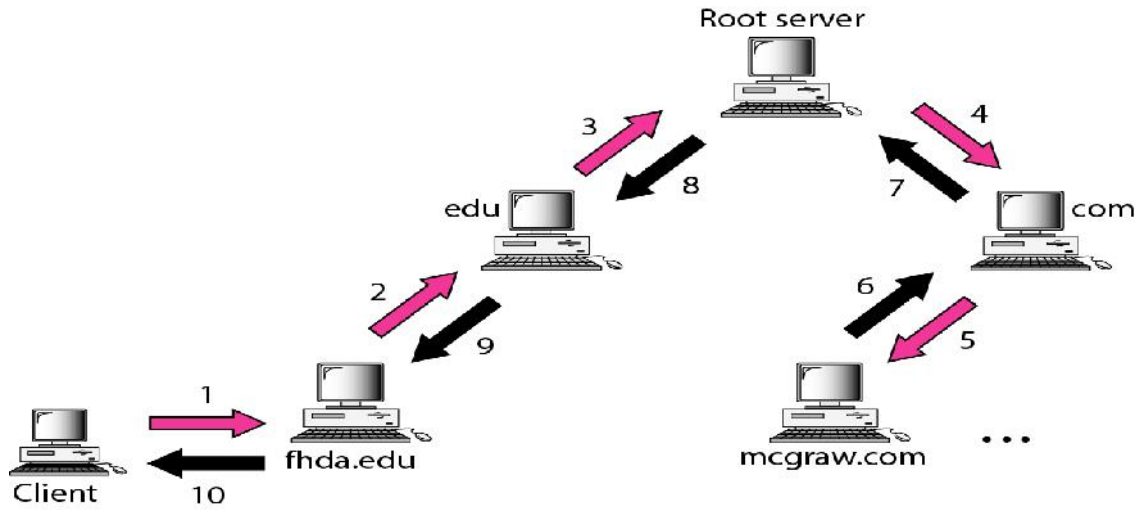
Generic Domains



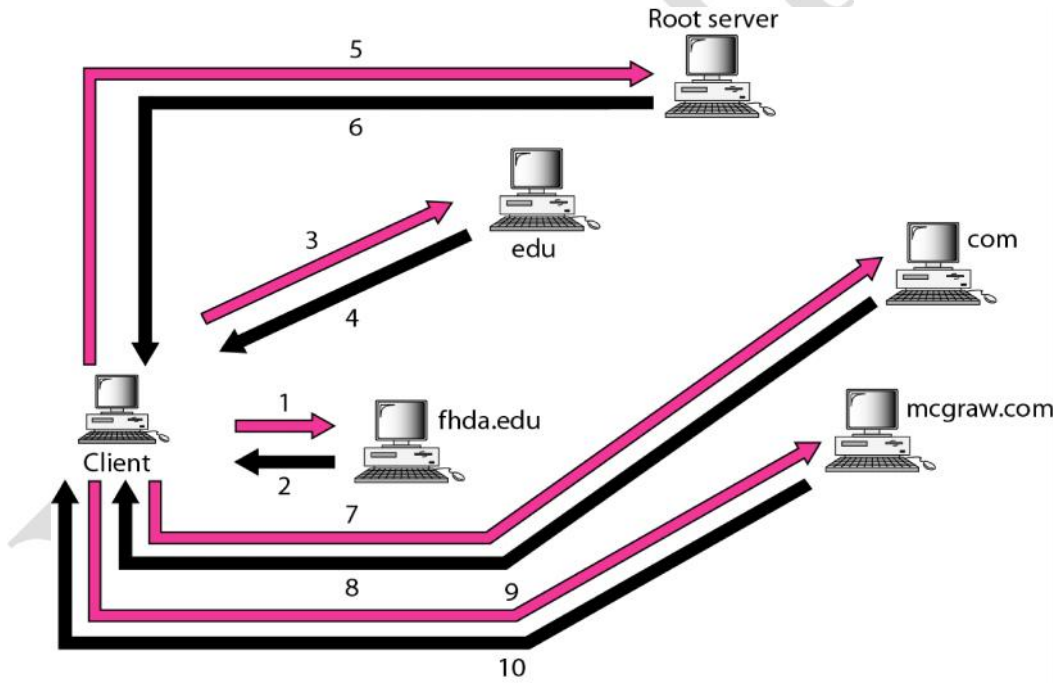
Generic Domain labels

<i>Label</i>	<i>Description</i>
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to “com”)
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

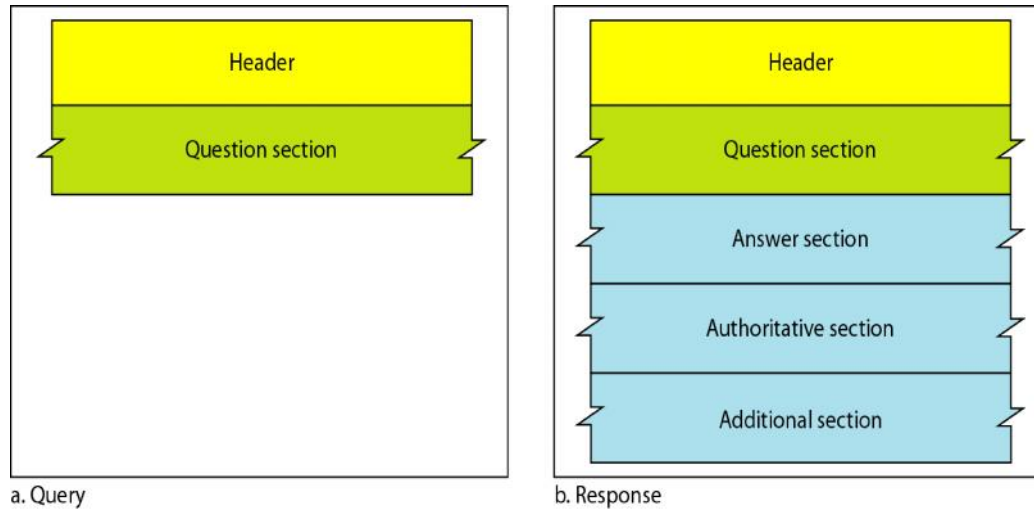
Recursive Resolution



Iterative Resolution



DNS Messages



Applications:- Remote Login

Telnet

- TELNET is a general protocol, meant to support logging in from almost any type of terminal to almost any type of computer.
- It allows a user at one site to establish a TCP connection to a login server or terminal server at another site.
- A TELNET server generally listens on TCP Port 23.

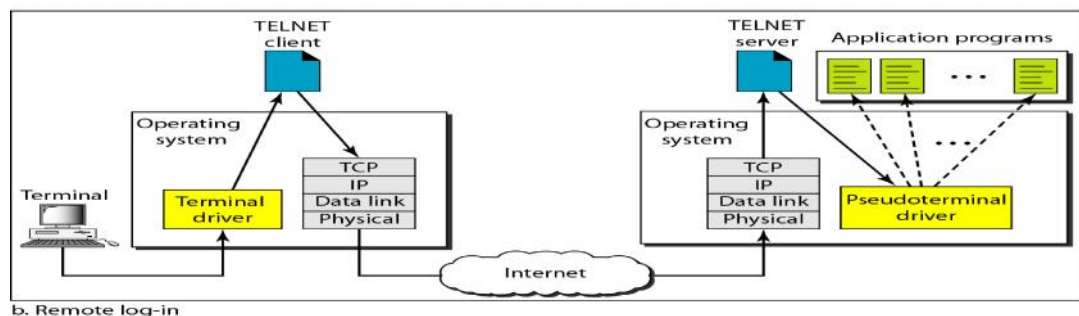
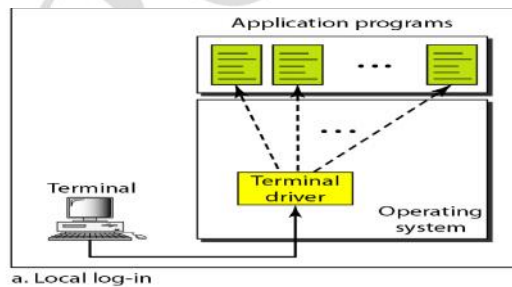
How it works

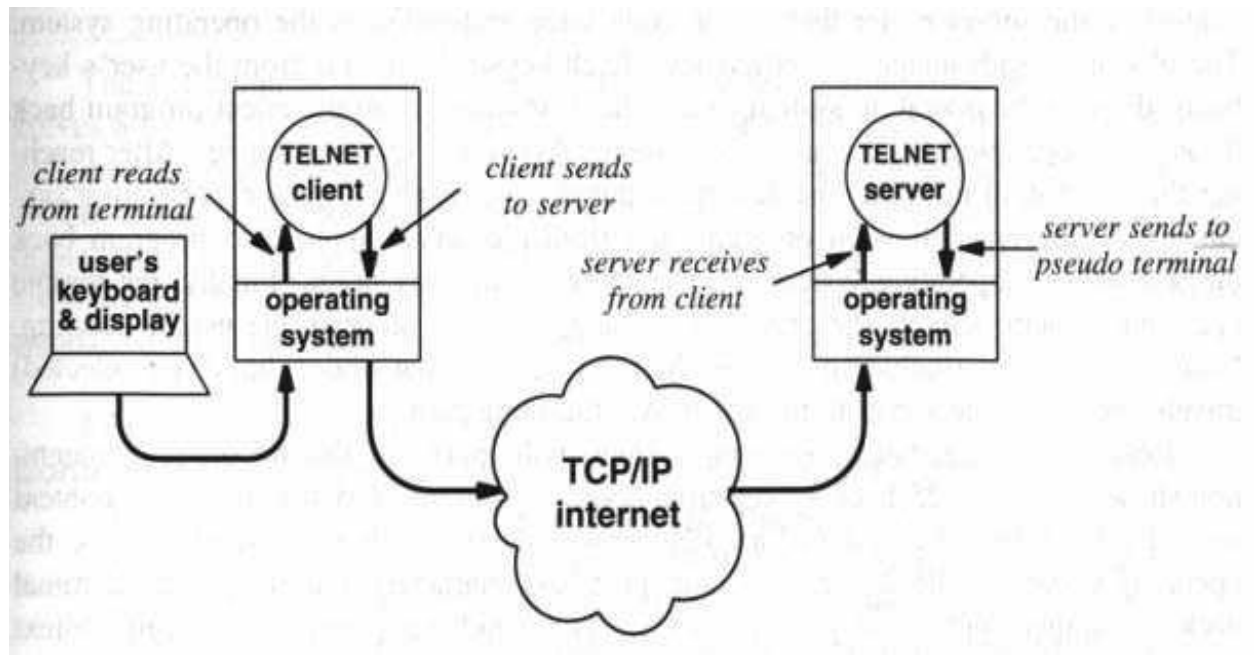
- A user is logged in to the local system, and invokes a TELNET program (the TELNET client) by typing
- `telnet xxx.xxx.xxx`

where xxx.xxx.xxx is either a host name or an IP address.

- The TELNET client is started on the local machine (if it isn't already running). That client establishes a TCP connection with the TELNET server on the destination system.
- Once the connection has been established, the client program accepts keystrokes from the user and relays them, generally **one character at a time**, to the TELNET server.
- The server on the destination machine accepts the characters sent to it by the client, and passes them to a terminal server.

- A "terminal server" is just some facility provided by the operating system for entering keystrokes from a user's keyboard.
 - The terminal server treats the remote user as it would any other user logged in to the system, including relaying commands to other applications.
 - The terminal server passes outputs back to the TELNET server, which relays them to the client, which displays them on the user's screen.
- In general, a TELNET server is implemented as a master server with some number of slave servers. The master server listens for service requests from clients. When it hears one, it spawns a slave server to handle that specific request, while the master goes back to listening for more requests.
- The only thing that makes TELNET hard to implement is the heterogeneity of the terminals and operating systems that must be supported. Not all of them use the same control characters for the same purposes.
- To accommodate this heterogeneity, TELNET defines a Network Virtual Terminal (NVT). Any user TELNETting in to a remote site is deemed to be on an NVT, regardless of the actual terminal type being used.
- It is the responsibility of the client program to translate user keystrokes from the actual terminal type into NVT format, and of the server program to translate NVT characters into the format needed by the destination host. For data sent back from the destination host, the translation is the reverse.
- NVT format defines all characters to be 8 bits (one byte) long. At startup, 7 bit US ASCII is used for data; bytes with the high order bit = 1 are command sequences.
- The 128 7-bit long US ASCII characters are divided into 95 printable characters and 33 control codes. NVT maps the 95 printable characters into their defined values - decimal 65 = "A", decimal 97 = "a", etc.
- The 33 control codes are defined for NVT as:





Secure Shell [SSH]

Secure Shell (SSH) Protocol is a protocol for secure network communications designed to be relatively simple and inexpensive to implement. The initial version, SSH1, focused on providing a secure remote logon facility to replace Telnet and other remote logon schemes that provided no security.

SSH also provides a more general client-server capability and can be used to secure such network functions as file transfer and e-mail.

A new version, SSH2, provides a standardized definition of SSH and improves on SSH1 in numerous ways. SSH2 is documented as a proposed standard in RFCs 4250 through.

SSH client and server applications are widely available for most operating systems. It has become the method of choice for remote login and X tunneling and is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems. SSH is organized as three protocols that typically run on top of TCP (Figure 1):

- *Transport Layer Protocol*: Provides server authentication, data confidentiality, and data integrity with forward secrecy (that is, if a key is compromised during one session, the knowledge does not affect the security of earlier sessions); the transport layer may optionally provide compression
- *User Authentication Protocol*: Authenticates the user to the server
- *Connection Protocol*: Multiplexes multiple logical communications channels over a single underlying SSH connection

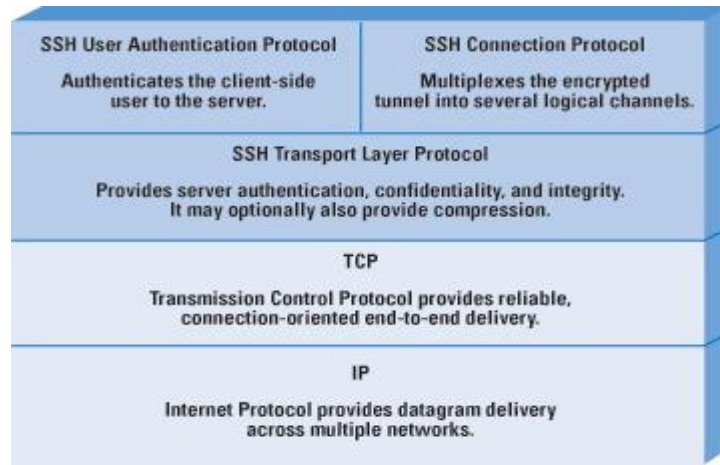
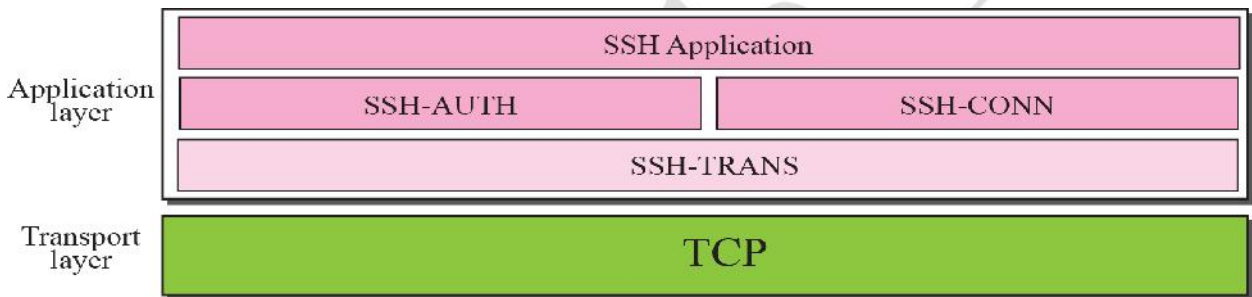
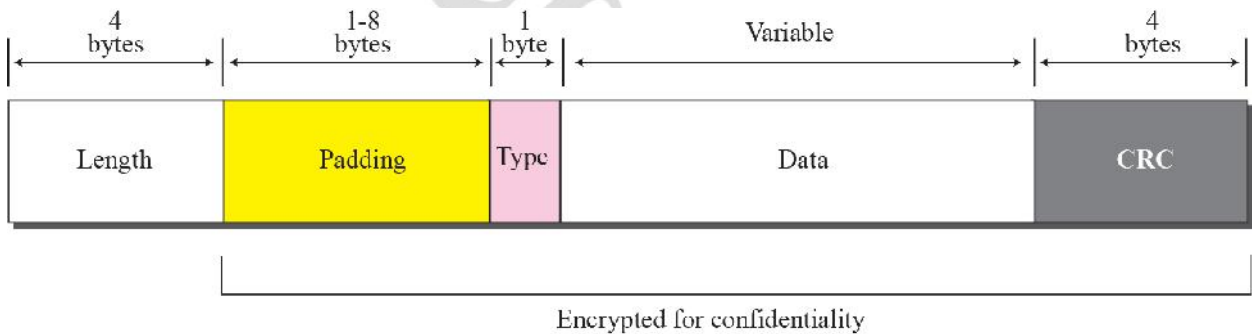


Figure : SSH Protocol Stack

Components of SSH



SSH Packet Format



Applications:- SMTP [Simple Mail Transfer Protocol]

SMTP was developed to send e-mail messages across the [Internet](#).

In the [OSI model](#), SMTP is an [application](#) layer protocol that utilizes [TCP](#) as the transport protocol to transmit mail to a destination mail exchanger, in other words, SMTP is used to transmit mail to a mail server. Mail can be transmitted by a client to the mail exchanger [server](#), or from mail exchanger to mail exchanger.

Mail sent via SMTP is usually sent from one mail exchanger to another, directly. E-mail was never designed to be instantaneous, but that is often how it appears to us.

Mail Exchangers (MX)

Mail Exchangers are the name given to the applications that support the SMTP protocol. Mail Exchangers such as sendmail or Microsoft Exchange should listen for [IP datagrams](#) that arrive on the [network interface](#) with a [TCP](#) port number of 25 and on . This port is one of the 'well known ports' defined in [RFC 1700](#). When a message is received, the mail exchanger should check to see if it is for one of its users, then move the mail to the user's mailbox.

To identify the mail exchangers for a [domain name](#), [DNS zone files](#) for the [domain](#) contain an [MX resource record](#) identifying the host name and [IP address](#) of the mail exchangers.

Simple mail transfer protocol differs from [Post Office Protocol version 3 \(POP3\)](#). [POP3](#) is used by [e-mail](#) client applications such as Microsoft Outlook, Mozilla Thunderbird, Eudora and other e-mail applications to retrieve mail stored in personal mailboxes at the mail [server](#).

E-Mail Clients

E-mail is a client-server protocol that allows the exchange of messages and attachments in various formats. An e-mail client is a [software](#) application which seamlessly handles all the technical communications tasks to connect to and find the e-mail at the [server](#), download the e-mail messages, organizes them and presents them to the user in a usable format. An e-mail client also provides the means to compose new messages, reply to and forward received messages, and to organize the messages for later review.

E-mail clients use [POP3](#) or [IMAP](#) instead of SMTP. SMTP is used strictly between mail servers.

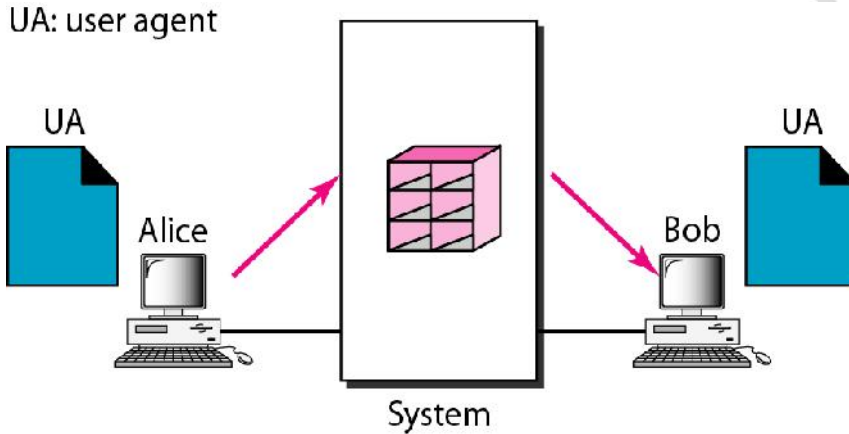
POP3 :- Post Office Protocol version 3 (POP3) is a standard mail protocol used to receive emails from a remote server to a local email client. POP3 allows you to download email messages on your local computer and read them even when you are offline.

IMAP:- The Internet Message Access Protocol (IMAP) is a mail protocol used for accessing email on a remote web server from a local client. IMAP and POP3 are the two most commonly used Internet mail protocols for retrieving emails. Both protocols are supported by all modern email clients and web servers.

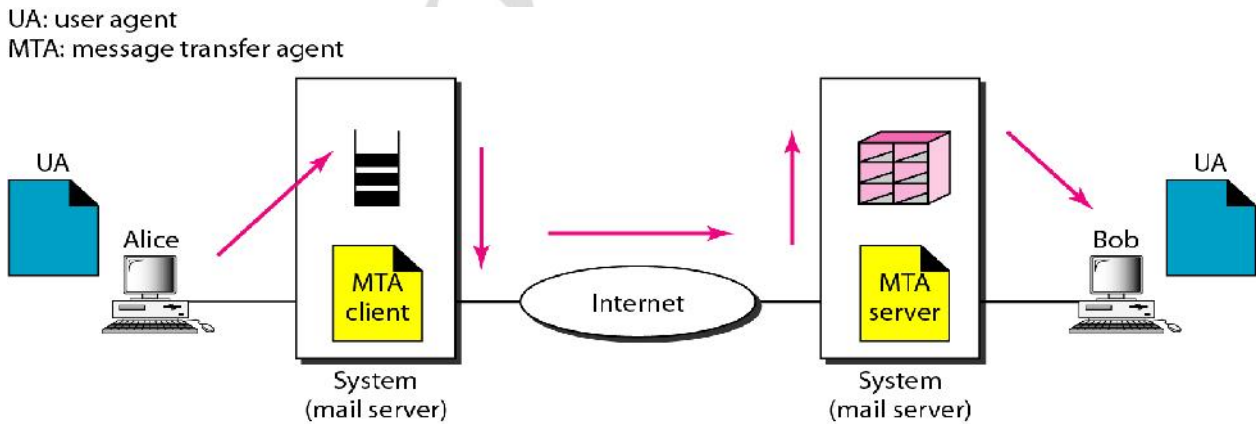
Main difference between IMAP and POP3:

The POP3 protocol assumes that there is only one client connected to the mailbox. In contrast, the IMAP protocol allows simultaneous access by multiple clients. IMAP is suitable for you if your mailbox is about to be managed by multiple users.

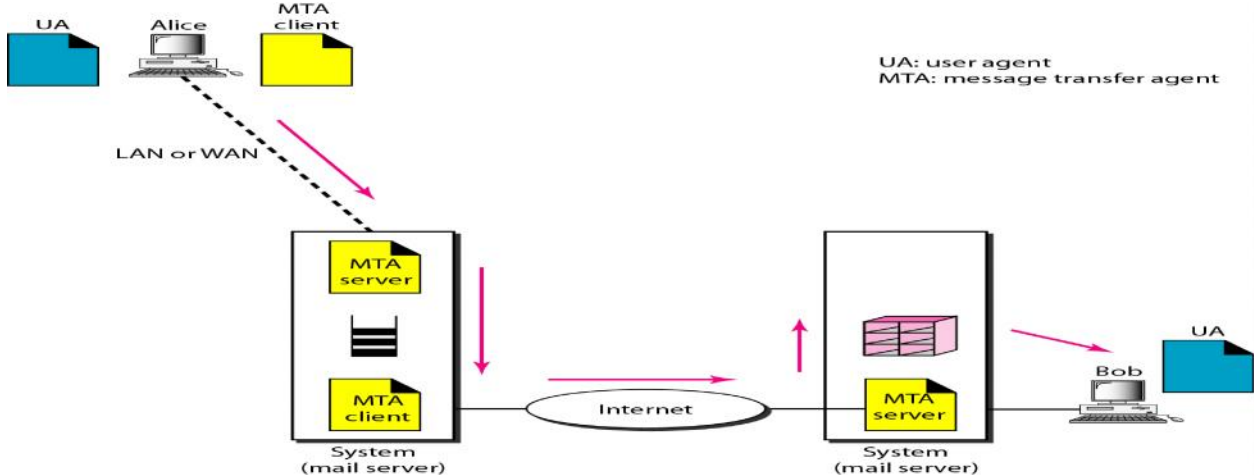
First scenario in electronic mail



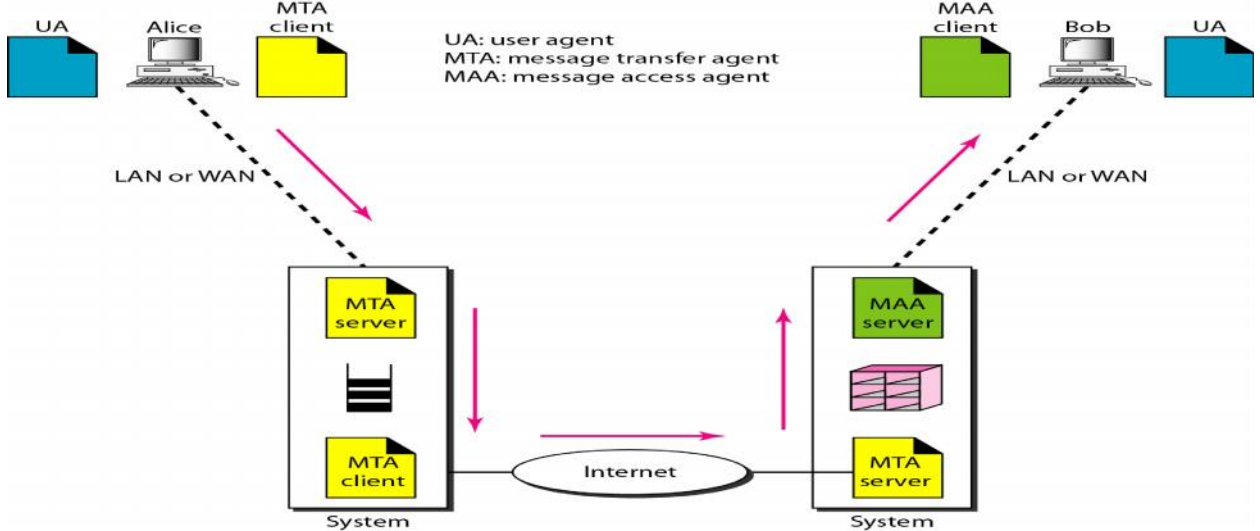
Second scenario in electronic mail



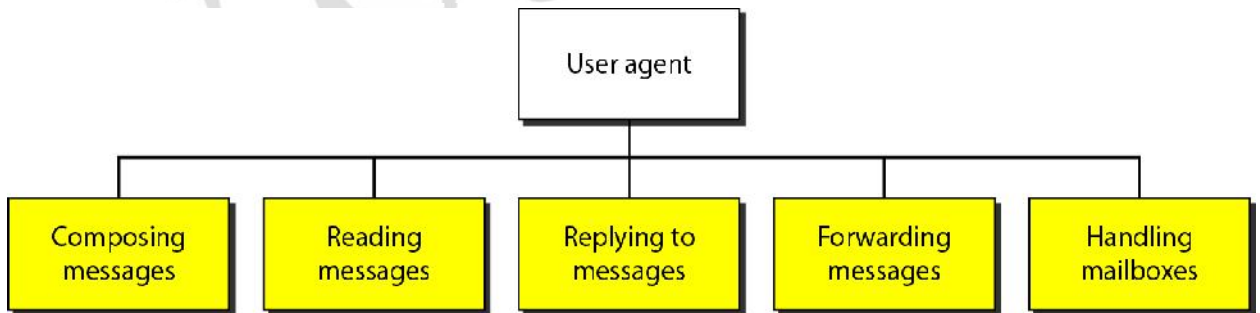
Third scenario in electronic mail



Fourth scenario in electronic mail



Services of user agent

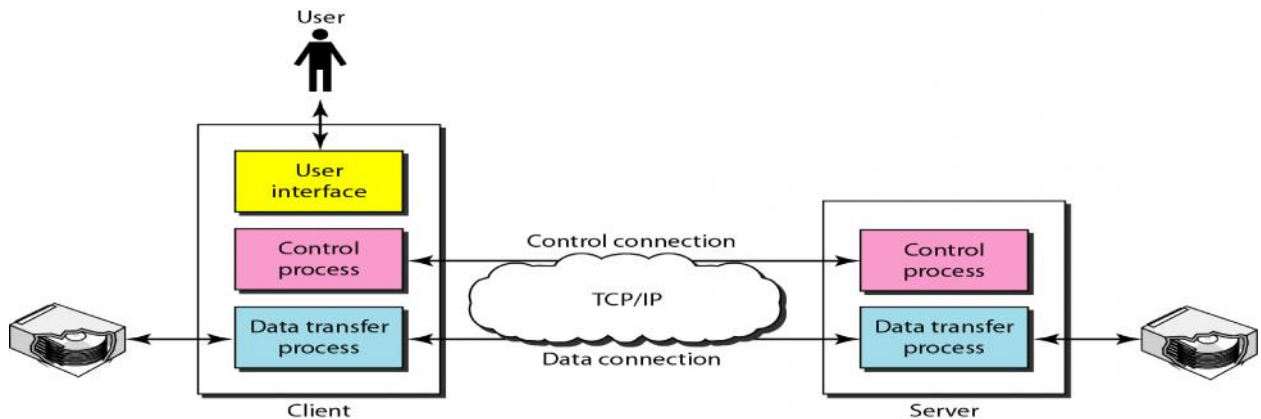


Applications:- File Transfer Protocol [FTP]

- FTP is an Internet (or more properly a TCP/IP) communications protocol to allow you to upload and download files from a machine connected to your local machine via the Internet. FTP is composed of two parts; an FTP client and an FTP server. The FTP client is the software you execute on your local machine to send or receive files. The FTP server is software which executes on a server machine on which the files are to be saved or retrieved. Most (if not all) machines which provide Web serving capability also serve as FTP servers to allow you to upload your web pages. Web pages are usually composed of HTML source files and images (either JPEG or GIF format).
- You should run FTP client software to connect to a FTP server on the Net.
- You must use an account to login.
 - Username / password
 - Anonymous / email address
- HTTP can be used for downloading file on the Web, but not as efficient as FTP.
- There are two modes of transfer in FTP:
 - ascii
 - binary.
- ASCII is used only for files saved in ASCII format (this includes Postscript files)
Binary is used for files that are formatted and saved using a wordprocessing software like WordPerfect (.txt), spreadsheets (.xls), images (.jpg, .gif), and many executable programs (.exe) and videos (.avi).

File	Extensions	Mode
Text file	.txt	ASCII
Spreadsheet, Excel	.xls, .xlw	Binary
Database file	.dbf	Binary
Word processing file	.doc	Binary
Program source code	.c, .java	ASCII
E-mails	N/A	ASCII
Unix tar file	.tar	Binary
Compressed file	.Z, .zip, .gz, .ar	Binary
Executable	.exe	Binary
Multimedia file	.gif, .mov, .wav	Binary
HTML file	.html, .htm	ASCII

FTP Example



There are many commands that can be used at the ftp prompt and users should refer to the manual page for full descriptions of all ftp interpreter commands.

Listing a Directory. Usually, the first command after making a connection is to list the contents of the directory. At the ftp prompt, type ``ls'` for a short-format listing or ``dir'` for a long-format listing.

Changing Directories. The next action you may want to take is to change to a subdirectory, which can be accomplished by typing ``cd remote-directory'` where ``remote-directory'` is the desired directory.

Transferring Files. Other important commands are the `get` and `put` commands, for example ``get remote-filename local-filename'` gets a file named ``remote-filename'` from the remote machine and stores it as ``local-filename'` on the local machine. If the file is not just plain text (an executable for example), you may want to change to binary mode by entering the ``binary'` command before your file transfer. To transfer multiple files, type ``prompt off'` and use the ``mput'` or ``mget'` commands with a file specification to match files, for example ``mget *.c'` to get all files with a ``.c'` extension. To abort a file transfer, use the terminal interrupt key (usually CTRL-C). Once you have ftp'ed files from remote system(s), you can view or edit text file(s) using your editor, run binaries for Sun Sparcstations from the Unix shell, or download file(s) to your home computer.

Exiting. To terminate the ftp connection with the remote server and exit ftp, type ``bye'` or ``quit'`.

Applications:- World Wide Web [WWW]

- The Internet has much to offer in terms of information on almost any subject matter imaginable and interaction with people and organizations from all over the world. Much of this access and interaction make use of the environment which is popularly known as the World Wide Web (WWW) or web. The WWW is an interlinked network of systems, called web servers, offering multimedia services and information. A user can access these using what is known as web browser software.
- The world wide web is a system of Internet servers that supports hypertext and multimedia to access several Internet protocols on a single interface. The World Wide Web is often abbreviated as the web or www.
- The World Wide Web was developed in 1989 by Tim Berners-Lee of the European Particle Physics Lab (CERN) in Switzerland. The initial purpose of the Web was to use networked hypertext to facilitate communication among its members, who were located in several countries.

Protocols of the Web

- The surface simplicity of the Web comes from the fact that many individual protocols can be contained within a single Web site. Internet protocols are sets of rules that allow for intermachine communication on the Internet. These are a few of the protocols you can experience on the Web:
- **HTTP** (HyperText Transfer Protocol): transmits hypertext over networks. This is the protocol of the Web.
- **E-mail** (Simple Mail Transport Protocol or SMTP): distributes e-mail messages and attached files to one or more electronic mailboxes.
- **FTP** (File Transfer Protocol): transfers files between an FTP server and a computer, for example, to download software.
- **VoIP** (Voice over Internet Protocol): allows delivery of voice communications over IP networks, for example, phone calls.

Hypertext and links: the motion of the Web

- The operation of the Web relies primarily on hypertext as its means of information retrieval. HyperText is a document containing words that connect to other documents. These words are called links and are selectable by the user.
- A single hypertext document can contain links to many documents. In the context of the Web, words or graphics may serve as links to other documents, images, video, and sound. Links may or may not follow a logical path, as each connection is created by the author of the source document. Overall, the Web contains a complex virtual web of connections among a vast number of documents, images, videos, and sounds.
- Producing hypertext for the Web is accomplished by creating documents with a language called hypertext markup language, or html. With HTML, tags are placed

within the text to accomplish document formatting, visual features such as font size, italics and bold, and the creation of hypertext links.

```
<p> This is a paragraph that shows the underlying HTML code. <strong>This sentence is rendered in bold text</strong>. <em>This sentence is rendered in italic text.</em> </p>
```

- HTML is an evolving language, with new tags being added as each upgrade of the language is developed and released. Nowadays, design features are often separated from the content of the HTML page and placed into cascading style sheets (css). This practice has several advantages, including the fact that an external style sheet can centrally control the design of multiple pages. The World Wide Web Consortium (W3C), led by Web founder Tim Berners-Lee, coordinates the efforts of standardizing HTML. The W3C now calls the language XHTML and considers it to be an application of the XML language standard.
- **Pages on the Web**
- The backbone of the World Wide Web are its files, called pages or Web pages, containing information and links to resources - both text and multimedia - throughout the Internet.
- Web pages can be created by user activity. For example, if you visit a Web search engine and enter keywords on the topic of your choice, a page will be created containing the results of your search.
- Access to Web pages can be accomplished in all sorts of ways, including:
 - Entering a Web address into your browser and retrieving a page directly
 - Browsing through sites and selecting links to move from one page to another both within and beyond the site
 - Doing a search on a search engine to retrieve pages on the topic of your choice (See: [The World of Search Engines](#))
 - Searching through directories containing links to organized collections of Web pages (See: [The World of Subject Directories](#))
 - Clicking on links within e-mail messages
 - Using apps on social networking sites or your mobile phone to access Web and other online content
 - Retrieving updates via RSS feeds and clicking on links within these feeds (See: [RSS Basics](#))

Retrieving documents on the Web: the URL and Domain Name System

- url stands for uniform resource locator. The URL specifies the Internet address of a file stored on a host computer, or server, connected to the Internet. Web browsers use the URL to retrieve the file from the server. This file is downloaded to the user's computer, or client, and displayed on the monitor connected to the machine. Because of this relationship between clients and servers, the Web is a client-server network.

Programming languages and environments

- The use of programming languages beyond HTML extend the capabilities of the Web. They are used to write software, process Web forms, fetch and display data, and perform all kinds of advanced functions. It is difficult to talk about these languages without getting into too much technical jargon, but here is an attempt. What follows is a brief guide to some of the more common languages in use on the Web today.
- **CGI (Common Gateway Interface)** refers to a specification by which programs can communicate with a Web server. A CGI program, or script, is any program designed to process data that conforms to the CGI specification. The program can be written in any programming language, including C, Perl, and Visual Basic Script (VBScript). In the early days of the Web, CGI scripts were commonly used to process a form on a Web page. Perl is popular with Google, and is also the language of the [Movable Type](#) blog platform.
- **Active Server Pages (ASP):** Developed by Microsoft, ASP is a programming environment that processes scripts on a Web server. The programming language VBScript is often used for the scripting. Lightweight programs can be written with this language. Active Server Pages end in the file extension .asp. For an example, check out [Databases and Indexes](#) at the University at Albany Libraries.
- **.NET framework:** Also developed by Microsoft, this development framework is a more powerful one than ASP for writing applications for the Web. Programming languages include C+ and VB.Net. ASP.Net is a related environment, producing pages with the file extensions .aspx. The [Microsoft](#) site is a good example of a site created with the .NET framework.
- **PHP:** This is another server-based language. It is frequently the language used to write open source (e.g., nonprofit, community-created) programs found on the Web, including [MediaWiki](#) (the software that runs the [Wikipedia](#)), and the popular blog software [WordPress](#). While PHP functionality can be installed on Windows servers, it is native to the Linux server environment and commonly used there.
- **Java/Java Applets:** Java is a programming language similar to C++. Developed by Sun Microsystems, the aim of Java is to create programs that will be platform independent. The Java motto is, "Write once, run anywhere." A perfect Java program should work equally well on a Windows, Apple, Unix, or Linux server, and so on, without any additional programming. This goal has yet to be realized. Java can be used to write applications for both Web and non-Web use.
- Web-based Java applications are usually in the form of **Java servlets**. These are small Java programs fetched from within a Web page that can be downloaded from a server and run on a Java-compatible Web browser. A Web page that links to a Java servlet has the file extension .jsp.
- **JavaScript** is a very popular programming language created by Netscape Communications. Small programs written in this language are embedded within a Web page, or fetched externally from within the page, to enhance the page's functionality. Examples of JavaScript include drop-down menus, image displays, and mouse-over interactions. The drop-down menus on the site of the UCLA Library shown below are a good example: when you hover your mouse over the menu item, a set of sub-menus opens up below.
- **XML:** XML (eXtensible Markup Language) is a mark-up language that enables Web designers to create customized tags to provide functionality not available with HTML

alone. XML is a language of data structure and exchange, and allows developers to separate form from content. With XML, the same content can be formatted for multiple applications. In May 1999, the W3 Consortium announced that HTML 4.0 has been recast as an XML application called XHTML.

- **AJAX** stands for Asynchronous JavaScript and XML. This language is used to create interactive Web applications. Its premise is that it sends data to the browser behind the scenes, so that when it is time to view the information, it is already "there." [Google Maps](#) is a well-known example of AJAX. A different kind of example can be found with [SurfWax LookAhead](#), an RSS search tool that retrieves feeds as you type your search.
- **SQL (Structured Query Language):** This is a language that focuses on extracting data from databases. Programmers write statements called queries that retrieve data from the tables in the database. Some Web sites are created extensively or entirely from data stored in database tables. You can often tell that a SQL query has produced data on a page by the presence of a question mark (?) and a record number in the URL, as the example below illustrates.
- **Mashups**
- Programs on the Web can be flexible. Sometimes they are combined with each other to form enhanced presentations. These are known as mashups.
- A mashup is a Web application or Web page that combines data from two or more external sources. Mashups give you access in one place to information available in multiple places.