UNIT –
VAPPLICATION
LAYER

The Application Layer- DNS - Name Space - Resource Records - Name Servers- E-Mail - Architecture And Services – The User Agent –Message Format –Message Transfer –Final Delivery – WWW –Architecture – Static Web Pages – Dynamic Web Pages And Web Applications –HTTP –Network Security –Introduction To Cryptography –Substitution Ciphers-Transposition Ciphers-Public Key Algorithms – RSA – Authentication Protocols –Authentication Using Kerberos.

DOMAIN NAME SYSTEM

This is primarily used for mapping host and e-mail destinations to IP addresses but can also be used other purposes. DNS is defined in RFCs 1034 and1035.

Working:-

- To map a name onto an IP address, an application program calls a library procedure called Resolver, passing it the name as a parameter.
- The resolver sends a UDP packet to a local DNS server, which then looks up the name and returns the IP address to the resolver, which then returns it to the caller.
- Armed with the IP address, the program can then establish a TCP connection with the destination, or send it UDP packets.

1. **The DNS namespace.**
2. **Resource Records.**
3. **Name Servers.**

1. THE DNS NAMESPACE:

The Internet is divided into several hundred top level domains, where each domain covers many hosts. Each domain is partitioned into sub domains, and these are further partitioned as so on. All these domains can be represented by a tree, in which the leaves represent domains that have no sub domains. A leaf domain may contain a single host, or it may represent a company and contains thousands of hosts. Each domain is named by the path upward from it to the root. The components are separated by periods (pronounced “dot”)

Eg: Sun Microsystems Engg. Department = eng.sun.com.

The top domain comes in 2 flavours:-

- **Generic:** com(commercial), edu(educational instructions), mil(the U.S armed forces, government), int (certain international organizations), net(network providers), org (non profit organizations).

- **Country:**include1entryforeverycountry.Domainnamescanbeeitherabsolute(endswithaperiod e.g. eng.sum.com) or relative (doesn't end with a period). Domain names are case sensitive and the component names can be up to 63 characters long and full path names must not exceed 255 characters.

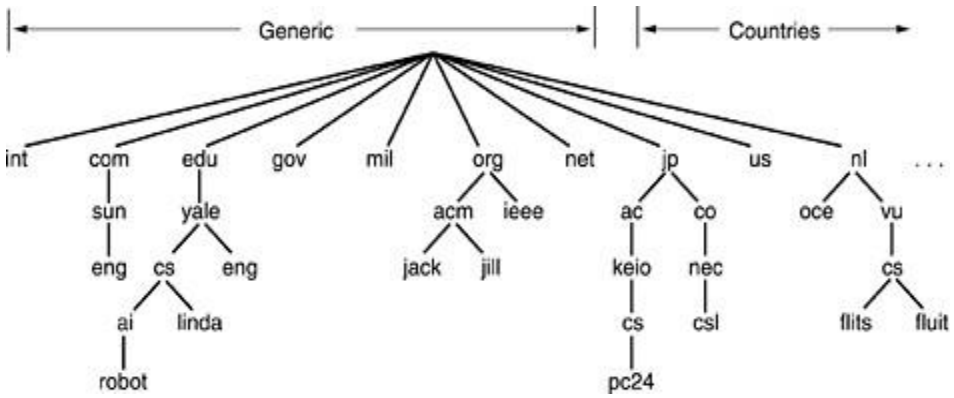


Figure 5-1. A portion of the Internet domain name space.

Insertions of a domain into the tree can be done in 2 days:-
• Under a generic domain (Eg: cs.yale.edu)
• Under the domain of their country (E.g:cs.yale.ct.us)

2. RESOURCERECORDS:

Every domain can have a sent of resource records associated with it. For a single host, the most common resource record is just its IP address. When a resolver gives a domain name to DNS, it gets both the resource records associated with that name i.e., the real function of DNS is to map domain names into resource records. A resource record is a 5-tuple and its format is as follows:

Domain	Name	Time to live	Type	Class	Value

Domain _name : Tells the domain to which this record applies.
Time- to- live : Gives an identification of how stable the record is (High Stable = 86400 i.e. no. of seconds /day) (High Volatile = 1 min)
Type: Tells what kind of record this is.
Class: It is IN for the internet information and codes for non internet information
Value: This field can be a number a domain name or an ASCII string

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

3. NAMESERVERS:

It contains the entire database and responds to all queries about it. DNS name space is divided up into non-overlapping zones, in which each zone contains some part of the tree and also contains name servers holding the authoritative information about that zone.

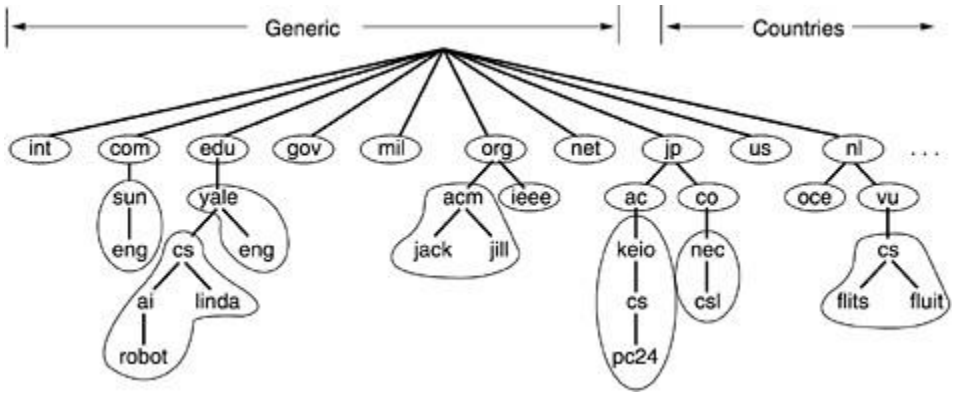


Figure 5-2. Part of the DNS name space showing the division into zones.

When a resolver has a query about a domain name, it passes the query to one of the local name servers:

1. If the domain being sought falls under the jurisdiction of name server, it returns the authoritative resource records (that comes from the authority that manages the record, and is always correct).
2. If the domain is remote and no information about the requested domain is available locally the name server sends a query message to the top level name server for the domain requested.

E.g.: A resolver of flits.cs.vle.nl wants to know the IP address of the host Linda.cs.yale.edu

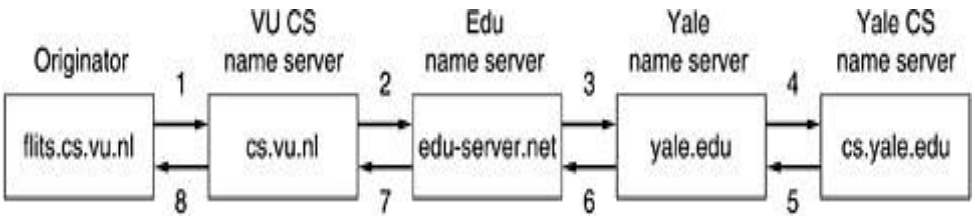


Figure 5-3. How a resolver looks up a remote name in eight steps.

Step 1: Resolver sends a query containing domain name sought the type and the class to local name server, cs.vu.nl.

Step 2: Suppose local name server knows nothing about it, it asks few others nearby name servers. If none of them know, it sends a UDP packet to the server for edu-server.net.

Step 3: This server knows nothing about Linda.cs.yale.edu or cs.yale.edu and so it forwards the request to the name server for yale.edu.

Step 4: This one forwards the request to cs.yale.edu which must have authoritative resource records.

Step 5 to 8: The resource record requested works its way back in steps 5-8 This query method is known as **Recursive Query**

3. When a query cannot be satisfied locally, the query fails but the name of the next server along the line to try is returned.

ELECTRONIC MAIL

1. ARCHITECTURE ANDSERVICES:

E-mail systems consist of two subsystems. They are:-

(1). User Agents, which allow people to read and send e-mail

(2). Message Transfer Agents, which move messages from source to destination

E-mail systems support 5 basic functions:-

a. Composition
b. Transfer
c. Reporting
d. Displaying
e. Disposition

(a). Composition: It refers to the process of creating messages and answers. Any text editor is used for body of the message. While the system itself can provide assistance with addressing and numerous header fields attached to each message.

(b). Reporting: It has to do with telling the originator what happened to the message that is, whether it was delivered, rejected (or) lost.

(c). Transfer: It refers to moving messages from originator to the recipient.

(d). Displaying: Incoming messages are to be displayed so that people can read their email.

(e). Disposition: It concerns what the recipient dose with the message after receiving it. Possibilities include throwing it away before reading (or) after reading, saving it and soon.

Most systems allow users to create **mailboxes** to store incoming e-mail. Commands are needed to create and destroy mailboxes, inspect the contents of mailboxes, insert and delete messages from mailboxes, and so on.

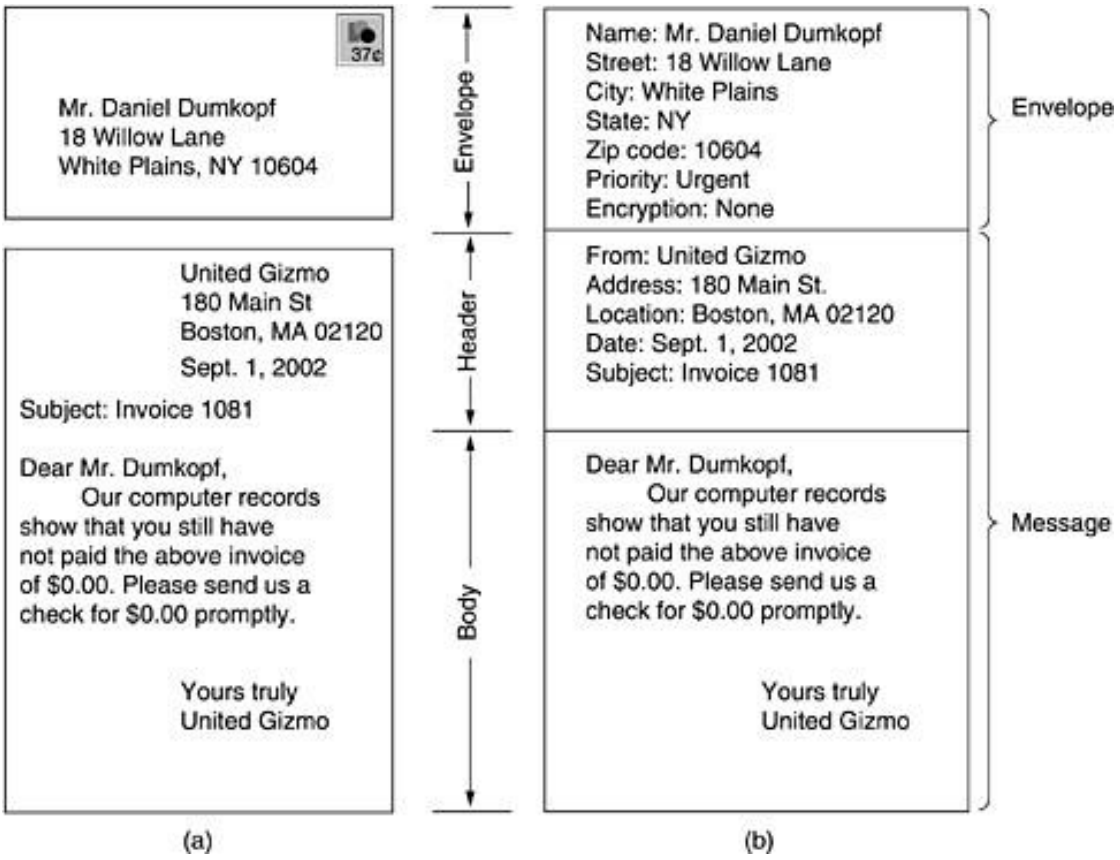


Figure 5-4: Envelopes and messages. (a) Paper mail. (b) Electronic mail.

(1) THE USERAGENT

A user agent is normally a program (sometimes called a mail reader) that accepts a variety of commands for composing, receiving, and replying to messages, as well as for manipulating mailboxes.

SENDING E-MAIL

To send an e-mail message, a user must provide the message, the destination address, and possibly some other parameters. The message can be produced with a free-standing text editor, a word processing program, or possibly with a specialized text editor built into the user agent. The destination address must be in a format that the user agent can deal with. Many user agents expect addresses of the form *user@dns-address*.

READING E-MAIL

When a user agent is started up, it looks at the user's mailbox for incoming e-mail before displaying anything on the screen. Then it may announce the number of messages in the mailbox or display a one-line summary of each one and wait for a command.

(2) MESSAGE FORMATS

RFC822

Messages consist of a primitive envelope (described in RFC 821), some number of header fields, a blank line, and then the message body. Each header field (logically) consists of a single line of ASCII text containing the field name, a colon, and, for most fields, a value.

Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

Figure 5-5: RFC 822 header fields related to message transport

MIME — The Multipurpose Internet Mail Extensions

RFC 822 specified the headers but left the content entirely up to the users. Nowadays, on the worldwide Internet, this approach is no longer adequate. The problems include sending and receiving

1. Messages in languages with accents (e.g., French and German).
2. Messages in non-Latin alphabets (e.g., Hebrew and Russian).
3. Messages in languages without alphabets (e.g., Chinese and Japanese).
4. Messages not containing text at all (e.g., audio or images).

A solution was proposed in RFC 1341 called **MIME** (**Multipurpose Internet Mail Extensions**)

The basic idea of MIME is to continue to use the RFC 822 format, but to add structure to the message body and define encoding rules for non-ASCII messages. By not deviating from RFC 822, MIME messages can be sent using the existing mail programs and protocols. All that has to be changed are the sending and receiving programs, which users can do for themselves.

Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content

Figure 5-6: RFC 822 headers added by MIME

MESSAGE TRANSFER

The message transfer system is concerned with relaying messages from the originator to the recipient. The simplest way to do this is to establish a transport connection from the source machine to the destination machine and then just transfer the message.

SMTP—THE SIMPLE MAIL TRANSFER PROTOCOL

SMTP is a simple ASCII protocol. After establishing the TCP connection to port 25, the sending machine, operating as the client, waits for the receiving machine, operating as the server, to talk first. The server starts by sending a line of text giving its identity and telling whether it is prepared to receive mail. If it is not, the client releases the connection and tries again later.

Even though the SMTP protocol is completely well defined, a **few problems** can still arise.

One problem relates to message length. Some older implementations cannot handle messages exceeding 64KB.

Another problem relates to timeouts. If the client and server have different timeouts, one of them may give up while the other is still busy, unexpectedly terminating the connection.

Finally, in rare situations, infinite mail storms can be triggered.

For example, if host 1 holds mailing list *A* and host 2 holds mailing list *B* and each list contains an entry for the other one, then a message sent to either list could generate a never-ending amount of e-mail traffic unless somebody checks for it.

FINAL DELIVERY

With the advent of people who access the Internet by calling their ISP over a modem, it breaks down.

One solution is to have a message transfer agent on an ISP machine accept e-mail for its customers and store it in their mailboxes on an ISP machine. Since this agent can be on-line all the time, e-mail can be sent to it 24 hours a day.

POP3

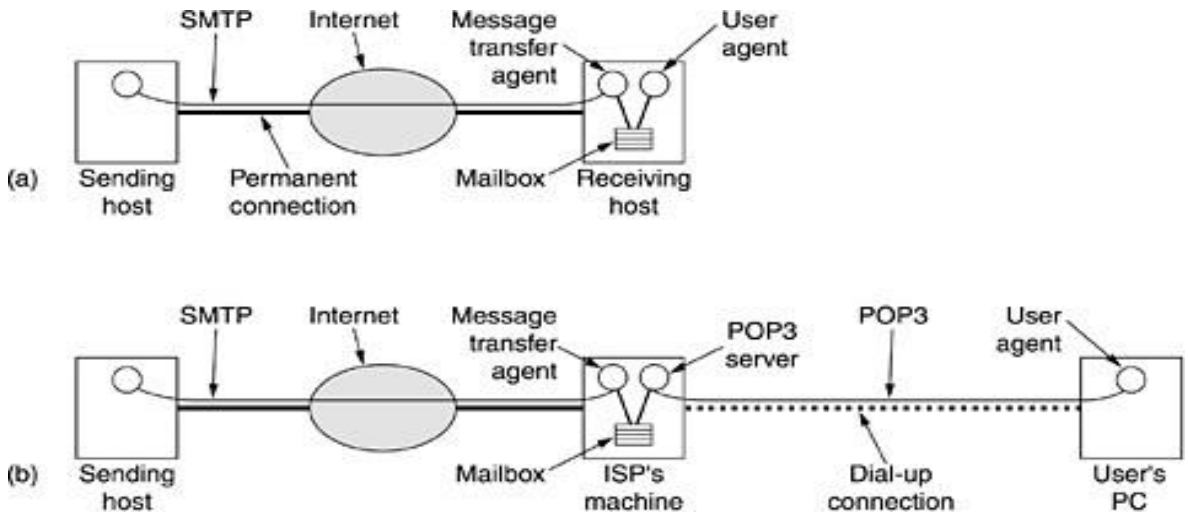


Figure:5-7

(a) Sending and reading mail when the receiver has a permanent Internet connection and the user agent runs on the same machine as the message transfer agent.

(b) Reading e-mail when the receiver has a dial-up connection to an ISP

POP3 begins when the user starts the mail reader. The mail reader calls up the ISP (unless there is already a connection) and establishes a TCP connection with the message transfer agent at port 110. Once the connection has been established, the POP3 protocol goes through three states in sequence:

1. Authorization.
2. Transactions.
3. Update.

The authorization state deals with having the user log in.

The transaction state deals with the user collecting the e-mails and marking them for deletion from the mailbox.

The update state actually causes the e-mails to be deleted.

IMAP(Internet Message Access Protocol).

POP3 normally downloads all stored messages at each contact, the result is that the user's e-mail quickly gets spread over multiple machines, more or less at random; some of them not even the user's.

This disadvantage gave rise to an alternative final delivery protocol, **IMAP** (**Internet Message Access Protocol**).

IMAP assumes that all the e-mail will remain on the server indefinitely in multiple mailboxes. IMAP provides extensive mechanisms for reading messages or even parts of messages, a feature useful when using a slow modem to read the text part of a multipart message with large audio and video attachments.

WORLD WIDE WEB

WORLD WIDE WEB

The World Wide Web is an architectural framework for accessing linked documents spread out over millions of machines all over the Internet. The initial proposal for a web of linked documents came from CERN physicist Tim Berners-Lee in 1989.

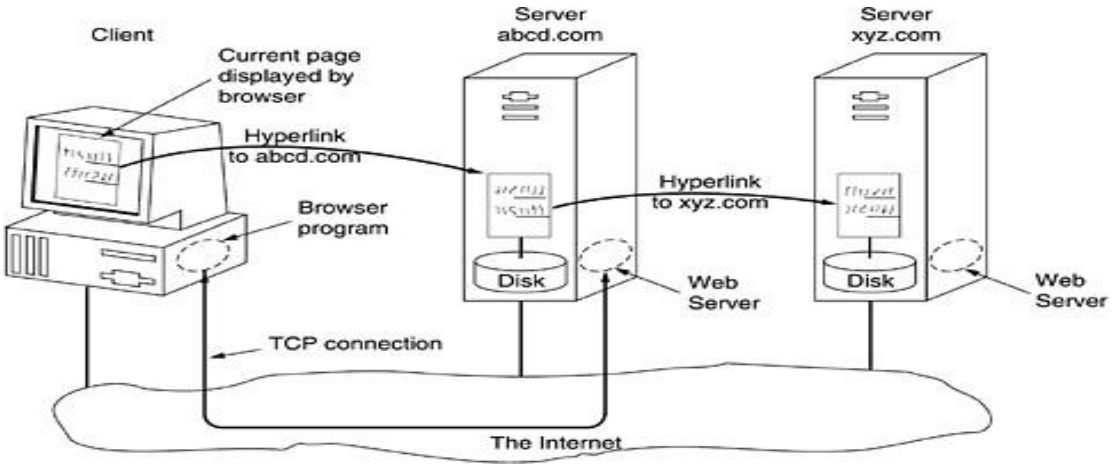
ARCHITECTURAL OVERVIEW

From the users' point of view, the Web consists of a vast, worldwide collection of documents or **Web pages.** Each page may contain links to other pages anywhere in the world. Users can follow a link by clicking on it, which then takes them to the page pointed to. This process can be repeated indefinitely.

Pages are viewed with a program called a **browser**, of which Internet Explorer and Netscape Navigator are two popular ones. The browser fetches the page requested, interprets the text and formatting commands on it, and displays the page, properly formatted, on the screen.

Strings of text that are links to other pages, called **hyperlinks**, are often highlighted, by underlining, displaying them in a special color, or both.

THE PARTS OF THE WEB MODEL



Here the browser is displaying a Web page on the client machine. When the user clicks on a line of text that is linked to a page on the *abcd.com* server, the browser follows the hyperlink by sending a message to the *abcd.com* server asking it for the page. When the page arrives, it is displayed. If this page contains a hyperlink to a page on the *xyz.com* server that is clicked on, the browser then sends a request to that machine for the page.

CLIENT SIDE

When an item is selected, the browser follows the hyperlink and fetches the page selected. Therefore, the embedded hyperlink needs a way to name any other page on the Web. Pages are named using **URLs** (**Uniform Resource Locators**).

The steps that occur at the client side are:

- The browser determines the URL
- The browser asks DNS for the IP address
- DNS replies with the IP address
- The browser makes a TCP connection to port 80 on the IP address
- It sends a request asking for file
- The *site* server sends the file
- The TCP connection is released.
- The browser fetches and displays all the text and images in the file.
- Web pages are written in standard HTML language to make it understandable by all browsers.

There are two possibilities: plug-ins and helper applications. A plug-in is a code module that the browser fetches from a special directory on the disk and installs as an extension to itself.

The other way to extend a browser is to use a helper application. This is a complete program, running as a separate process.

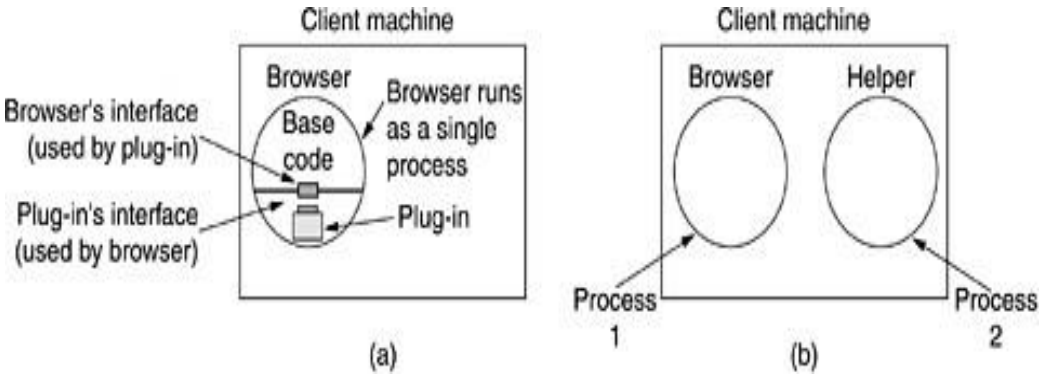


Figure 5-8. (a) A browser plug-in. (b) A helper application.

SERVER SIDE

The steps to be followed by the server side are:

1. Accept a TCP connection from a client (a browser).
2. Get the name of the file requested.
3. Get the file (from disk).
4. Return the file to the client.
5. Release the TCP connection.

PROCESSING OF REQUEST

The processing of request on the web is as follows:

1. Resolve the name of the Web page requested.
2. Authenticate the client.
3. Perform access control on the client.
4. Perform access control on the Web page.
5. Check the cache.
6. Fetch the requested page from disk.
7. Determine the MIME type to include in the response.
8. Take care of miscellaneous odds and ends.
9. Return the reply to the client.
10. Make an entry in the server log.

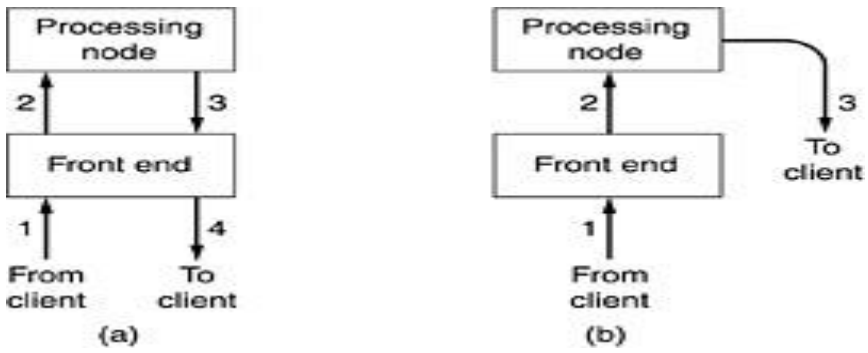


Figure 5-9. (a) Normal request-reply message sequence. (b) Sequence when TCP handoff is used

Sometimes a trick, called TCP handoff, is used to get around this problem. With this trick, the TCP end point is passed to the processing node so it can reply directly to the client.

URLs— UNIFORM RESOURCE LOCATORS

When the Web was first created, it was immediately apparent that having one page point to another Web page required mechanisms for naming and locating pages. In particular, three questions had to be answered before a selected page could be displayed:

1. What is the page called?
2. Where is the page located?
3. How can the page be accessed?

If every page were somehow assigned a unique name, there would not be any ambiguity in identifying pages. Nevertheless, the problem would not be solved.

Consider a parallel between people and pages. In the United States, almost everyone has a social security number, which is a unique identifier, as no two people are supposed to have the same one. Nevertheless, if you are armed only with a social security number, there is no way to find the owner's address, and certainly no way to tell whether you should write to the person in English, Spanish, or Chinese. The Web has basically the same problems.

The solution chosen identifies pages in a way that solves all three problems at once. Each page is assigned a **URL** (**Uniform Resource Locator**) that effectively serves as the page's worldwide name.

URLs have three parts: the protocol (also known as the **scheme**), the DNS name of the machine on which the page is located, and a local name uniquely indicating the specific page (usually just a file name on the machine where it resides). As an example, the Web site for the author's department contains several videos about the university and the city of Amsterdam. The URL for the video page is

http://www.cs.vu.nl/video/index-en.html

This URL consists of three parts: the protocol (http), the DNS name of the host (www.cs.vu.nl), and the file name (video/index-en.html), with certain punctuation separating the pieces. The file name is a path relative to the default Web directory at cs.vu.nl.

STATIC WEB DOCUMENTS

- The basis of the Web is transferring Web pages from server to client. In the simplest form, Web pages are static. They are just files sitting on some server waiting to be retrived.
- In this context, even a video is a static web page because it is just a file.
- In this section we will look at static web page in details. In the next one, we will examine dynamic content.

HTML—The Hyper Text Markup Language:

- HTML allows users to produce Web pages that include text, graphics, video, pointers to other Web pages, and more.
- HTML is a markup language, or language for describing how documents are to be formatted.
- Markup languages thus contain explicit commands for formatting. For example, in HTML, means start boldface mode, and means leave bold face mode.
- Writing a browser is then straightforward: the browser simply has to understand the mark up commands.
- Embedding all the markup commands within each HTML file and standardizing them makes it possible for any Web browser to read and reformat any Webpage.
- While it is certainly possible to write documents like this with any plain text editor, and many people do, it is also possible to use word processors or special HTML editors that do most of the work.
- A Web page consists of a head and a body, each enclosed by<html>and</html>tags.
- The head is bracketed by the <head>and </head>tags and the body is bracketed by the<body>and </body>tags. The strings inside the tags are called directives.
- Most, but not all, HTML tags have this format. That is, they use <something>to mark the beginning of something and </something>to mark it send.
- Tags can be in either lowercase or uppercase. Thus, <head>and <HEAD>mean the same thing, but lower case is best for compatibility.
- Some tags have (named) parameters, called attributes. For example, the<img>tag is used for including an image inline with the text. It has two attributes, src and alt. The first attribute gives the URL for the image.
- The list of special characters is given in the standard. All of them begin with an ampersand and end with a semicolon.
    - Forexample, producesaspace,èproducese`andéproducese´.Since<, >, and & have special meanings, they can be expressed only with their escape sequences, <, >, and &.
- The main item in the head is the title, delimited by <title>and </title>.The title itself is not displayed on the page. Some browsers use it to label the page's window.
- Several headings are used in each heading is generated by an <h*n*>tag, where *n* is a digit in the range 1 to 6. Thus, <h1>is the most important heading ;<h6>is the least important one.

- <h1>headingsarelargeandboldfacewithatleastoneblanklineaboveandbelow.Incontrast, <h2>headings are in a smaller font with less space above and below.
- Thetagsandareusedtoenterboldfaceanditalicsmode.The tagstartsaparagraph.the tag that exists to mark the end of a paragraph

Tag	Description
<html>…</html> <head>…</head> <title>…</title> <body>…</body> <h n>…</h n> … … … … … … <hr> <img src=””> ….	Declares the web page to be written in HTML. Delimits the page’s head. Defines the title. Delimits the page’s body. Delimits a level n heading. Set … in boldface. Set … in italic. Center … on the page horizontally. Brackets an unordered list. Brackets a numbered list. Brackets an item in an ordered or numbered list. Forces a line break here. Starts a paragraph. Inserts a horizontal rule. Displays an image here. Defines a hyperlink.

XML and XSL:

- XML and XSL is (eXtensible Markup Language) and (eXtensible Style Language).
- HTML, with or without forms, does not provide any structure to web pages.
- It also mixes the content with the formatting, as e-commerce and other applications become more common, there is an increasing need for structuring pages and separating the content from the formatting.
- The W#C has developed an enhancement to HTML to allow web pages to be structured for automated processing.
- It defines a structure called book_list, which is a list of books. Each book has three fields, the title, author, and year of publication.
- In this example, each of the three fields is an indivisible entity, but it is also permitted to futher subdivide the fields.
- The author fields could have been done as follows to give a finer-grained control over searching and formatting.

- Example:
  <author>
  <first_name>Andrew</first_name>
  <last_name>Tanenbaum</last_name>
  </author>
- Each field can be subdivided into subfields and sub subfields arbitrarily deep.
- The file is a style sheet that tells how to display the page, it is design view in the xml file.

XHTML: (The eXtended Hyper Text Markup Language)

HTML keeps evolving to meet new demands. Many people in the industry feel that in the future, the majority of web-enabled device will not be PCs, but wireless, handheld PDA-type device.

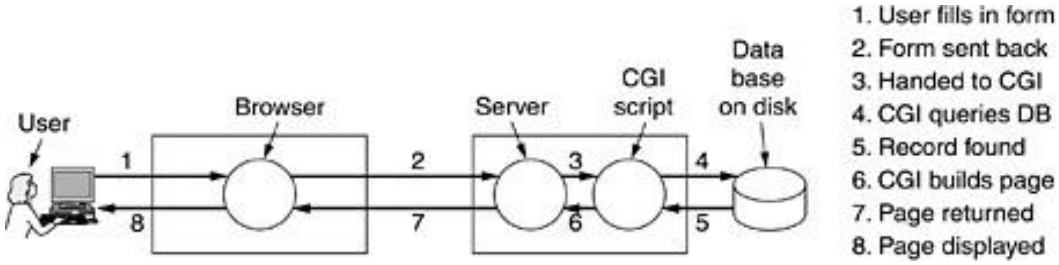
- These devices have limited memory for large browser full of heuristics that try to somehow deal with syntactically incorrect web pages.
- There are 6 major difference in HTML and XHML:

  - XHTML pages and browser must strictly conform to the standard. No more shodly web pages.
  - All tags and attributes must be in lower case, tags like <HTML> are not valid in XHTML.
  - Closing tags are required, even for </p>. for tags that have no natural closing tag. Such as
,<hr> and <img>,a slash must preced the closing”>”.
    Eg:
  - Attribute must be contained within quotation marks.
    **Eg: **
    The 500 has to be enclosed in quotation marks, just like the name of the JPEG file, even though 500 is just a number.
  - Tags must be nest properly. In the past, proper nesting was not required as long as final state achieved was correct. Tags closed in the inverse order that they were opened.
    Eg: vacation pictures .
  - Every document must specify its document type. For a discussion of all the changes, major and minor, seewww.w3.org.

DYNAMIC WEB DOCUMENTS

Dynamic web documents are created at both client and server sides.

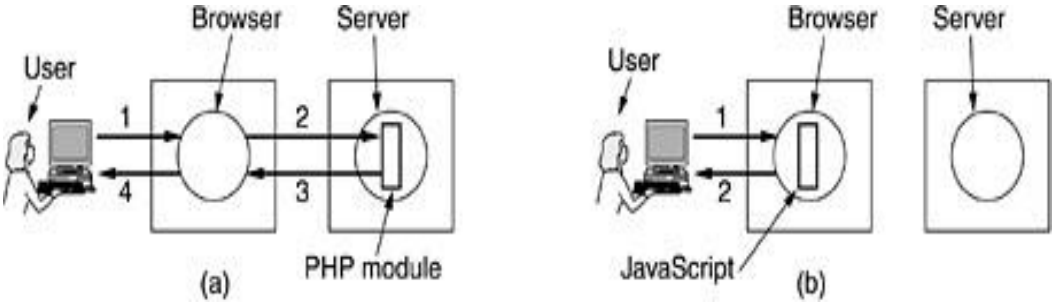
SERVER-SIDE GENERATION



The server side generation involves the following steps:

- User fills inform.
- Form sent back
- Handed to CGI
- CGI queries database.
- Record found
- CGI build spage
- Page returned
- Page displayed

CLIENT-SIDE GENERATION



CGI, PHP, JSP, and ASP scripts solve the problem of handling forms and interactions with databases on the server. They can all accept incoming information from forms, look up information in one or more databases, and generate HTML pages with the results.

Usually the server side scripting is done with PHP and client side scripting is java script. Complete web pages can be generated on the fly by various scripts on the server machine. Once they are received by the browser, they are treated as normal HTML pages and displayed.

Dynamic content generation is also possible on the client side. Web pages can be written in XML and then converted to HTML according to XSL file. Java script programs can perform arbitrary computations.

Finally plug ins and helper applications can be used to display content in a variety of formats.

NETWORK SECURITY

- The requirements of information security within an organization have undergone two major changes in the last decades.
- The generic name for the collection of tools designed **to protect data and to prevent hackers is computer security.**
- Another important thing is that affected security is the introduction of distributed systems and the use of networks and communication facilities for carrying data between terminal user and computer and between computer and compiler.
- Network security measures are needed to protect data during their transmission and to guarantee that data transmissions are authentic.

SECURITY ATTACKS:

Any action that compromises the security of information owned by an organization.

- Two types of security attacks are

  1)Passive Attack-Just listens the message.

  2) Active Attack-Alter message.

SECURITY MECHANISMS:

- A process that designed to detect, prevent, recover from security attack.

SECURITY SERVICE:

- Use of one or more security mechanisms to provide service.

IMPORTANT FEATURES OF SECURITY:

1) CONFIDENTIALITY-It is the protection of transmitted data from passive attacks.

2)AUTHENTICATION-The receiptient that the message is from source that it claim to be from.

3)NON REPUDIATION-It prevent either sender or receiver.

- When a message is sent ,the receiver can prove that the alleged sender in fact sent the message similarly when a message is received ,the sender can prove that alleged receiver in fact received the message.

SECURITY REQUIREMENTS AND ATTACKS

Computer and network security address three requirements.

- Secrecy
- Integrity
- Availability

SECRECY

Secrecy requires that the information in a computer system only be accessible for reading by authorized parties. This type of access includes printing, displaying and other forms of disclosure, including simply revealing the existence of an object.

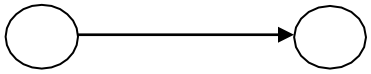
INTEGRITY

Integrity requires that computer system can be modified only by authorized parties. Modification includes writing, changing status, deleting and creating.

AVAILABILITY

- Availability requires that computer systems are available to authorized parties.
- The types of attacks on the security of a computer system or network can be viewed by the function of the computer system.

In general, there is a flow of information from a source to destination. This is called normal flow.



Information source destination source

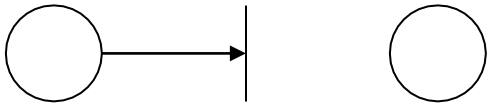
The following are the four categories of attack.

They are:

- Interruption
- Interception
- Modification
- Fabrication

INTERRUPTION

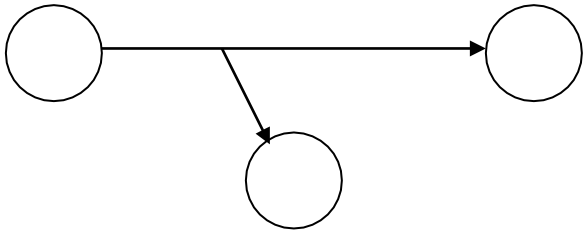
An asset of the system is destroyed or it becomes unavailable or unusable. This is an attack on availability. Examples: destruction of a piece of hardware such as hard disk, the cutting of a communication line, or the disabling of the file management system.



INTERCEPTION

An authorized party gains access to an asset. This is an attack on confidentiality. The authorized party could be a person or program or a computer.

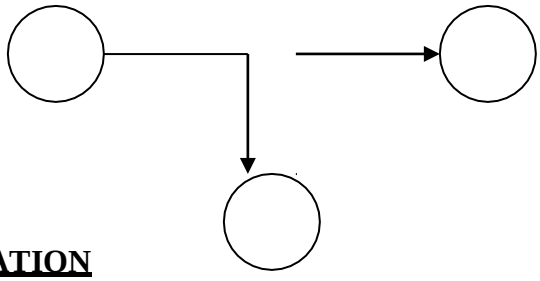
E.g.: wiretapping to capture data in a network and the illicit copying of files or programs.



MODIFICATION

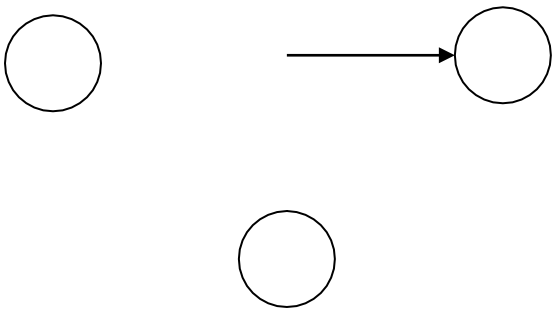
An authorized party gains access to an asset. This is an attack on confidentiality. This is an attack on integrity.

E.g: changing the values in a data file, altering a program so that it performs differently and modifying the content of messages being transmitted in a network.



FABRICATION

An authorized party inserts counterfeit objects into the system. This is an attack on authenticity. Examples include the insertion of spurious messages in a network or the addition of records to a file.



PASSIVE ATTACKS

Passive attacks means the eavesdropping on or monitoring of, transmissions. The goal of the component is to obtain information that is being transmitted. Two types of attacks are involved here.

- Release of message contents
- Traffic analysis

RELEASE OF MESSAGE CONTENTS

In this, a telephone conversation, an e-mail message, a transferred file may contain sensitive or confidential information. This helps to prevent the opponent from learning the content of these transmissions.

TRAFFIC ANALYSIS

In this encryption is used for masking the contents which helps to observe the pattern of the messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged.

Passive attacks are difficult to detect because they do not involve any alteration of data.

ACTIVE ATTACKS

In this, the attacks involve some modification of the data stream or creation of false stream. This is divided into four categories.

They are:

- Masquerade
- Replay
- Modification of messages
- Denial of service

- **Masquerade**

A masquerade takes place when one entity pretends to be a different entity. A masquerade attack usually includes one of the other forms of active attacks.

- **Replay**

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

- **Modification**

This means that some portion of a legitimate message is altered or that messages are delayed or reordered to produce an unauthorized effect.

- **Denial of service**

The denial of service prevents or inhibits the normal size or management of communication facilities.

Active attacks present the opposite characteristics of passive attacks.

It is quite difficult to prevent active attacks absolutely and would require physical protections of all communications facilities and paths at all times.

CRYPTOGRAPHY:

- Cryptography is the study of secret(crypto)writing(graphy)
- The art of science encompassing the principles and methods of transforming an intelligible message into one that is intelligible ,and then retransforming that message back to its original form.

Some of the encryption scheme are.

PLAINTEXT:This is the original intelligible message.

CIPHERTEXT:Transformed message.

ENCRYPTION ALGORITHMS:The encryption algorithm performs various substitutions and transformations on the plaintext.

SECRET KEY: Some critical information used by the cipher knows only to sender and receiver.

ENCIPHER(ENCODE): The process of converting plaintext to ciphertext.

DECIPHER(DECODE): The process of converting ciphertext back into plaintext.

CRYPTOGRAPHY SYSTEMS ARE CHARACTERISED INTO THREE:

1) The type of operation used for transforming plaintext to cipher text.

Two general principles

- Substitution.
- Tranposition.

SUBSTITUTION: In which each element in the plaintext(bit,letter,group or letter is mapped into another element.

TRANSPOSITION:In which elements in plaintext are rearranged.

2)The numbers of keys used.

Both sender and receiver use same key as

- Symmetric
- Single key
- Secret key or conventional encryption.

Both sender and receiver use differentkey

- Asymmetric ,two key or public keyencryption.

3)The way in which plaintext isprocessed.

- **A Block cipher** processes the input one block of elements at a time ,producing an output block foreach input block.(ie.Encrypt one bit/character at atime)

  Eg:THISISEASY KEY(3)

  WKLV LV HDVB.

- **A stream cipher** processs the input elements continuosly,producing output one element at atime.(ie. Break plaintext message in equal size blocks and encrypt each block as aunit).
  Eg:THIS IS EASY

  THI ISISEASY….. KEY(135)
  UKNJWN……

INTRODUCTION TO CRYPTOGRAPHY

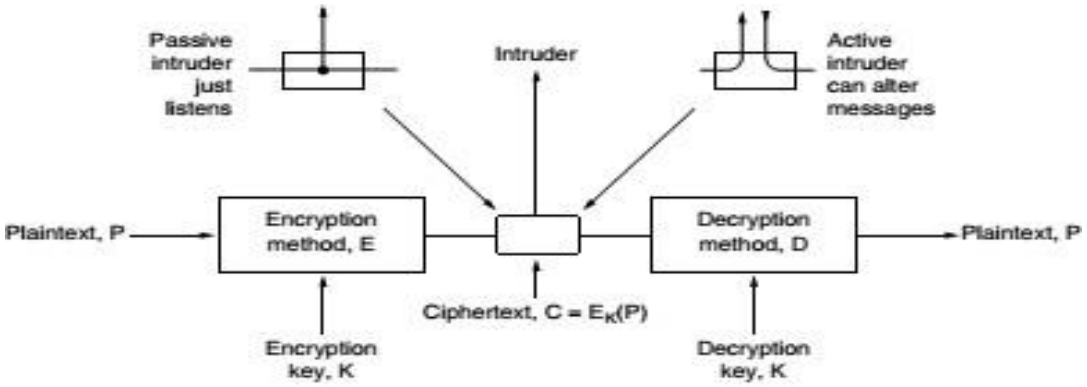


Figure 8-2. The encryption model (for a symmetric-key cipher).

- The messages to be encrypted, known as the plaintext, are transformed by a function that is parameterized by akey.
- The output of the encryption process, known as the ciphertext, is then transmitted, often by messenger or radio.
- We assume that the enemy, or intruder, hears and accurately copies down the completeciphertext.
- However, unlike the intended recipient, he does not know what the decryption key is and so cannot decrypt the ciphertexteasily.
- Sometimes the intruder can not only listen to the communication channel (passive intruder) but can also record messages and play them back later, inject his own messages, or modify legitimate messages before they get to the receiver(activeintruder)

SUBSTITUTION TECHNIQUES:

- The two basic building block of all encryption techniques are substitution andtransposition.
- A substitution techniques is one in which the letters of plaintext are replaced by other letters or by number orsymbols.

CAESAR CIPHER:

- The earliest known use of a substitution cipher and the simplest,was by JuliusCaesar.
- The Caesar cipher involves replacing each letter of alphabets with the letter standing three places further down thealphabets.

Eg:plaintext: meet me after the toga party.

Ciphertext:PHHW PH DIWHU WKH WRJD SDUWB.

-The letter following Z to A.

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

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

MONOALPHABETIC CIPHER:

- The general system of symbol for symbol substitutioncipher.
- With the key being 26 letter string corresponding to the fullalphabet.

Eg:A->X,B->Y,C->Z,D->A…….Z->W

Eg) starbucks at three

PQXOYRZHP XQ QEOBB

- The basic attack takes advantage of the statistical properties of naturallanguages.
- In English, for example ,e is the most common letter, followed by t, o, a, n, i,etc.

- The most common two-letter combinations, or digrams, are th, in, er, re, andan.
- The most common three-letter combinations, or trigrams,are the, ing, and, andion.
- A cryptanalyst trying to break a monoalphabetic cipher would start outby
- counting the relative frequencies of all letters in theciphertext.
- He would then look at trigrams to find a common one of the form tXe, which strongly suggeststhat
- X is h. Similarly, if the pattern thYt occurs frequently, they probably stands fora.

TRANSPOSITION TECHNIQUES:

- A very different kind of mapping is achieved by performing some sort of permutation on plaintext letters.
- This technique is referred as transpositioncipher.
- In which letters of the plaintext are written alternating between rows and the rows and thenread sequentially to givecipher.

WE ARE DISCOVERED SAVE YOURSELF would be written

```
W A E I C V R D A E O R E F
E R D S O E E S V Y U S L
```

or

```
W A E I C V R D A E O R E F E R D S O E E S V Y U S L .
```

- To write the message in rectangle row by row and read the message off,column by column but permute the order ofcolumn.
- The order of column then becomes keyword AUTHOR and order the column by lexicographic orderof the letters in thekeyword.

A	U	T	H	O	R
1	6	5	2	3	4
W	E	A	R	E	D
I	S	C	O	V	E
R	E	D	S	A	V
E	Y	O	U	R	S
E	L	F	A	B	C

yields the cipher

```
W I R E E R O S U A E V A R B D E V S C A C D O F E S E Y L .
```

One-Time pad:

Introduction:

- First described by Frank Miller in1882.
- The one-time pad was re-invented in 1917 and patented a couple of yearslater.

- It is derived from the *Vernam cipher*, named after Gilbert Vernam, one of itsinventors.
- Vernam's system was a cipher that combined a message with a key read from a punchedtape.
- In its original form, Vernam's system was vulnerable because the key tape was a loop, which was reused whenever the loop made a fullcycle.
- One-time use came later, when Joseph Mauborgne recognized that if the key tape were totally random, then cryptanalysis would be impossible.

Or

- The One-Time Pad is an evolution of the Vernham cipher, which was invented by Gilbert Vernham in 1918, and used a long tape of random letters to encrypt themessage.
- AnArmySignalCorpofficer,JosephMauborgne,proposedanimprovementusingarandomkeythat was truly as long as the message, with no repetitions, which thus totally obscures the original message.
- Since any plaintext can be mapped to any ciphertext given some key, there is simply no way to determine which plaintext corresponds to a specific instance ofciphertext.

Define:

- Each new message requires a new key of the same length as the new message. Such a scheme, known as a one-timepad.
- It isunbreakable.
- It produces a random output that bears no statistical relationship to theplaintext.
- Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code.

Example:

Suppose Alice wishes to send the message "HELLO" to Bob. Assume two pads of paper containing identical random sequences of letters were somehow previously produced and securely issued to both. Alice chooses the appropriate unused page from the pad.

The way to do this is normally arranged for in advance, as for instance 'use the 12th sheet on 1 May', or 'use the next available sheet for the next message'. The material on the selected sheet is the *key* for this message. Each letter from the pad will be combined in a predetermined way with one letter of the message. It is common, but not required, to assign each letter a numerical value, e.g., "A" is 0, "B" is 1, and so on. In this example, the technique is to combine the key and the message using modular addition. The numerical values of correspondingmessageandkeylettersareaddedtogether,modulo26.Ifkeymaterialbeginswith"XMCKL" and the message is "HELLO", then the coding would be done asfollows:

```
   H      E      L      L      Omessage
 7 (H) 4 (E) 11 (L) 11 (L) 14 (O) message
+ 23 (X) 12 (M) 2 (C) 10 (K) 11 (L) key
=30     16     13     21     25     message +key
= 4 (E) 16 (Q) 13 (N) 21 (V) 25 (Z) message + key (mod 26)
```

```
E   Q   N   V   Z →ciphertext
```

If a number is larger than 25, then the remainder after subtraction of 26 is taken in modular arithmetic fashion. This simply means that if the computations "go past" Z, the sequence starts again at A.

The ciphertext to be sent to Bob is thus "EQNVZ". Bob uses the matching key page and the same process, but in reverse, to obtain the plaintext. Here the key is *subtracted* from the ciphertext, again using modular arithmetic:

```
   E    Q    N    V    Zciphertext
 4 (E) 16 (Q) 13 (N) 21 (V) 25 (Z) ciphertext
- 23 (X) 12 (M) 2 (C) 10 (K) 11 (L) key
= -19    4    11   11   14   ciphertext – key
= 7 (H)  4 (E) 11 (L) 11 (L) 14 (O) ciphertext – key (mod 26)
   H    E    L    L    O → message
```

Similar to the above, if a number is negative then 26 is added to make the number zero or higher.

- The Security of the one-time pas is entirely due to the randomness of thekey.
- If the Stream of characters that constitute the key is truly random, then the stream of characters that constitute the ciphertext will be trulyrandom.
- Thus, there are no patterns or regularities that a cryptanalyst can use to attack theciphertext.

The one-time pad offers complete security but, in practice, has two fundamentaldifficulties:

1. There is the practical problem of making large quantitied of random keys. Any heavily used system might requires millions of random character on a regular basis. Supply truly random characters in this volume is a significanttask.
2. Even more daunting is the problem of key distribution and protection. For every message to be sent, a key of equal length is needed by both sender and receiver. Thus, a mammoth key distribution problem exists.

Cryptographic Principles:

- Redundancy
  - All encrypted messages must contain some redundancy, that is, information not needed to understand themessage.
- Freshness
  - Some measures must be taken to ensure that each message received can be verified as being fresh, that is, sent veryrecently.

Redundancy Motivation:

- Consider a mail-order company, The Couch Potato (TCP), with 60,000products.
- Ordering messages consist of a 16-byte customer name followed by a 3-byte datafield.

- The last 3 bytes are to be encrypted using a very long key known only by the customer andTCP.
- This might seem secure since passive intruders cannot decrypt themessages.
- Suppose that a recently-fired employee wants to punishTCP.
- Just before leaving, he takes the customer list withhim.
- He writes a program to generate fictitious orders using real customernames.
- Since he does not have the list of keys, he just puts random numbers in the last 3 bytes, and sends hundreds oforders.
- When these messages arrive, TCP's computer uses the customer's name to locate the key and decrypt the message.
- Unfortunately for TCP, almost every 3-byte message is valid, so the computer begins printing out shippinginstructions.
- In this way an active intruder can cause a massive amount of trouble, even though he cannot understand the messages his computer isgenerating.
- This problem can be solved by the addition of redundancy to allmessages.
- For example, if order messages are extended to 12 bytes, the first 9 of which must be zeros, then this attack no longer works because the ex-employee can no longer generate a large stream of valid messages.

All messages must contain considerable redundancy so that active intruders cannot send random junk and have it be interpreted as a valid message

Freshness

- This measure is needed to prevent active intruders from playing back oldmessages.
- If no such measures were taken, our ex-employee could keep repeating previously sent validmessages.
- Some method is needed to foil replayattacks
- A solution is to include in every message a timestamp valid only for, say, 10seconds.
- The receiver can then just keep messages around for 10 seconds. Messages older than 10 seconds can be thrown out.

RSA

- RSA is a public keyalgorithm.
- It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public keycryptography.
- RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-dateimplementations.

Much practical security is based on it. Its major disadvantage is that it requires keys of at least 1024 bits for good security (versus 128 bits for symmetric-key algorithms), which makes it quite slow.

The RSA algorithm involves three steps:

- keygeneration
- encryption
- decryption

RSA involves a **public** key and a **private key.** The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key.

The RSA method is based on some principles from number theory. We will now summarize how to use the method.

1. Choose two large primes, *p* and *q* (typically 1024bits).
2. Compute $n = p \times q$ and $z = (p - 1) \times (q - 1)$.
3. Choose a number relatively prime to *z* and call it *d*.
4. Find *e* such that $e \times d = 1 \ mod z$.

The security of the method is based on the difficulty of factoring large numbers. If the cryptanalyst could factor the (publicly known) *n*, he could then find *p* and *q*, and from these *z*.

With these parameters computed in advance, we are ready to begin encryption. Divide the plaintext (regarded as a bit string) into blocks, so that each plaintext message, *P*, falls in the interval $0 \ P < n$. Do that by grouping the plaintext into blocks of *k* bits, where *k* is the largest integer for which $2^k < n$ is true.

To encrypt a message, *P*, compute $C = P^e \pmod n$. To decrypt *C*, compute $P = C^d \pmod n$. It can be proven that for all *P* in the specified range, the encryption and decryption functions are inverses. To perform the encryption, you need *e* and *n*. To perform the decryption, you need *d* and *n*. Therefore, the public key consists of the pair (*e*, *n*), and the private key consists of (*d*, *n*).

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	$P^3 \pmod{33}$	C^7	$C^7 \pmod{33}$	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E
		Sender's computation		Receiver's computation		

In the above example, the encryption of the plain text “SUZANNE” is shown:

p = 3, q = 11, n = 33, z = 20

- d = 7 (since 7 and 20 have no commonfactors)
- 7e = 1 (mod20)
- e =3
- C = P3 (mod33)
- P = C7 (mod33)

Encryption:

C = Me modn
= 887 mod 187
= [(884 mod 187) * (882 mod 187) * (881 mod 187] mod 187
= [(59,969,536 mod 187)(7744 mod 187)(88 mod 187)] mod187
= (132 * 77 * 88) mod 187
= 894,432 mod 187
= 11

Decryption:

M =Cd modn
=1123 mod 187
=[(111 mod 187)* (112 mod 187)* (114 mod 187)*(118 mod 187)* (118mod 187)] mod 187
=[(111mod187)*(121mod187)*(14,641mod187)*(214,358,881mod187)*(214,358,881mod187)]mod187
=(11 * 121 * 55 * 33 * 33) mod 187
=79,720,245 mod 187
=88.

If p = 3, q = 11, n = 33, Φ(n) = 20,
d = 7 (because 7, 20 have no common factors)
=>7e = 1 mod 20
=3

Some form of chaining is needed for data encryption. However, in practice, most RSA-based systems use public-key cryptography primarily for distributing one-time session keys for use with some symmetric-key algorithm such as AES or triple DES. RSA is too slow for actually encrypting large volumes of data but is widely used for keydistribution.

8.7.4 Authentication Using Kerberos

An authentication protocol used in many real systems (including Windows 2000 and later versions) is **Kerberos**, which is based on a variant of Needham-Schroeder. It is named for a multiheaded dog in Greek mythology that used to guard the entrance to Hades (presumably to keep undesirables out). Kerberos was designed at M.I.T. to allow workstation users to access network resources in a secure way. Its biggest difference from Needham-Schroeder is its assumption that all clocks are fairly well synchronized. The protocol has gone through several iterations. V5 is the one that is widely used in industry and defined in RFC 4120. The earlier version, V4, was finally retired after serious flaws were found (Yu et al., 2004). V5 improves on V4 with many small changes to the protocol and some improved features, such as the fact that it no longer relies on the now-dated DES. For more information, see Neuman and Ts'o (1994).

Kerberos involves three servers in addition to Alice (a client workstation):

1. Authentication Server (AS): Verifies users during login.
2. Ticket-Granting Server (TGS): Issues "proof of identity tickets."
3. Bob the server: Actually does the work Alice wants performed.

AS is similar to a KDC in that it shares a secret password with every user. The TGS's job is to issue tickets that can convince the real servers that the bearer of a TGS ticket really is who he or she claims to be.

To start a session, Alice sits down at an arbitrary public workstation and types her name. The workstation sends her name and the name of the TGS to the AS in plaintext, as shown in message 1 of Fig. 8-42. What comes back is a session key and a ticket, $K_{TGS}(A, K_{S,t})$, intended for the TGS. The session key is encrypted using Alice's secret key, so that only Alice can decrypt it. Only when message 2 arrives does the workstation ask for Alice's password—not before then. The password is then used to generate K_A in order to decrypt message 2 and obtain the session key.

At this point, the workstation overwrites Alice's password to make sure that it is only inside the workstation for a few milliseconds at most. If Trudy tries logging in as Alice, the password she types will be wrong and the workstation will detect this because the standard part of message 2 will be incorrect.

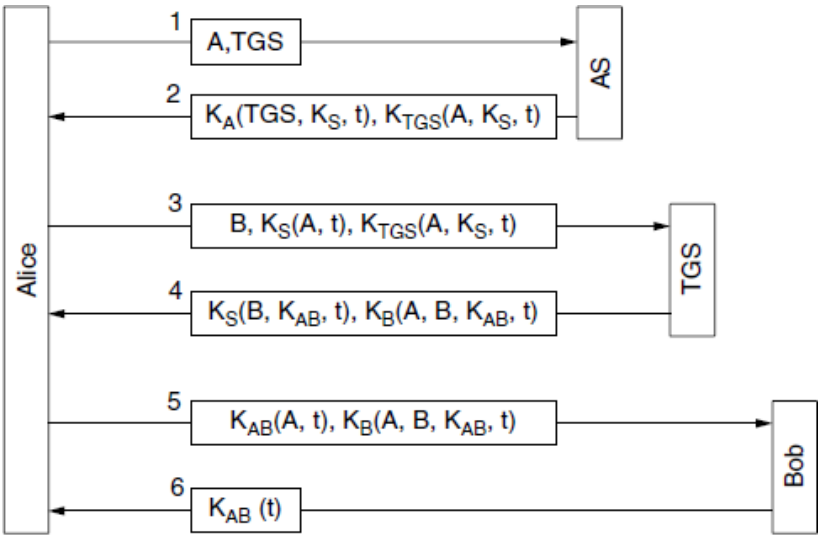


Figure 8-42. The operation of Kerberos V5.

After she logs in, Alice may tell the workstation that she wants to contact Bob the file server. The workstation then sends message 3 to the TGS asking for a ticket to use with Bob. The key element in this request is the ticket $K_{TGS}(A, K_{S,}t)$, which is encrypted with the TGS's secret key and used as proof that the sender really is Alice. The TGS responds in message 4 by creating a session key, K_{AB}, for Alice to use with Bob. Two versions of it are sent back. The first is encrypted with only K_S, so Alice can read it. The second is another ticket, encrypted with Bob's key, K_B, so Bob can read it.

Trudy can copy message 3 and try to use it again, but she will be foiled by the encrypted timestamp, t, sent along with it. Trudy cannot replace the timestamp with a more recent one, because she does not know K_S, the session key Alice uses to talk to the TGS. Even if Trudy replays message 3 quickly, all she will get is another copy of message 4, which she could not decrypt the first time and will not be able to decrypt the second time either.

Now Alice can send K_{AB} to Bob via the new ticket to establish a session with him (message 5). This exchange is also timestamped. The optional response (message 6) is proof to Alice that she is actually talking to Bob, not to Trudy.

After this series of exchanges, Alice can communicate with Bob under cover of K_{AB}. If she later decides she needs to talk to another server, Carol, she just repeats message 3 to the TGS, only now specifying C instead of B. The TGS will promptly respond with a ticket encrypted with K_C that Alice can send to Carol and that Carol will accept as proof that it came from Alice.

The point of all this work is that now Alice can access servers all over the network in a secure way and her password never has to go over the network. In fact, it only had to be in her own workstation for a few milliseconds. However, note that each server does its own authorization. When Alice presents her ticket to Bob, this merely proves to Bob who sent it. Precisely what Alice is allowed to do is up to Bob.

Since the Kerberos designers did not expect the entire world to trust a single authentication server, they made provision for having multiple **realms**, each with its own AS and TGS. To get a ticket for a server in a distant realm, Alice would ask her own TGS for a ticket accepted by the TGS in the distant realm. If the distant TGS has registered with the local TGS (the same way local servers do), the local TGS will give Alice a ticket valid at the distant TGS. She can then do business over there, such as getting tickets for servers in that realm. Note, however, that for parties in two realms to do business, each one must trust the other's TGS. Otherwise, they cannot do business.