MODULE 1
Introduction, Basics of Cryptography, Secret Key Cryptography

- Computer security is all about studying cyber attacks with a view to defending against them.
- The attacks include pharming and phishing attacks together with assorted malware and denial of service attacks.
- Understanding what makes systems vulnerable to these attacks is an important first step in avoiding or preventing them.
- There are different classes of vulnerabilities including those caused by poorly written or configured software.
- There are diverse defence strategies such as Access control, authentication, and data protection techniques are introduced.

1.1 Cyber Attacks

1.1.1Motives

- **"What are the main goals of an attacker?"**
  The sheer thrill of mounting a successful cyber attack has been motivation enough for hackers (Table 1.1).
- Most hackers were (and still are) young adults, often teens, who had dropped out of school but were otherwise intelligent and focused.
- Many of the ***"traditional" hackers*** seem to be obsessive programmers.
- Often hackers use ***scripts and attack kits*** designed by others (these can be freely downloaded from the Internet). Their activities do not require any special programming skills or advanced knowledge of computer systems.
- Other perpetrators of cyber attacks include company insiders, often employees who wish to gain illegal acces and have extra privileges
- There is also a ***serious threat from cyber terrorists***
- ***Cyber terrorism*** is one weapon which may include biological, chemical, and nuclear weapons. Their goals are to ***cripple the information/communication systems of the financial and business institutions of their "enemies."***
- The primary motivation for launching cyber attacks has shifted to ***financial gain.***

Table 1 . 1 Notable cyber attacks

Year	Event
1988	Robert Morrisa 23-year-old Cornell graduate student, released a worm that over an

	Arpanet, incapacitating almost 6000 computers, congesting government and university systems. He was fined $10,000 and sentenced to 3 years probation.
1991	31-year-old David L. Smith created the worm "Melissa," which infected thousands of computers causing damage of approximately $1.5 billion. This virus sent copies of itself to the first 50 names of the recipient's address book. He received a 20-month jail term.
2001	"Anna Kournikova" virus. Promising photos of the tennis star mailed itself to the every person in the victim's address book. Investigators were apprehensive that the virus was created with a toolkit enabling the rookies to create a virus.
2008	The headquarters of the Obama and McCain presidential campaigns were hacked.

Some of the main motives of launching cyber attacks are:

1. **Theft of sensitive information.**
2. **Disruption of service.**
3. **Illegal access to or use of resources.**

1. Theft of sensitive information.

- Many organizations store and communicate sensitive information.
- Information on new products being designed or revenue sources can be hugely advantageous to a company's competitors.
- Likewise, details of **military installations or precise military plans** can be of immense value to a nation's adversaries.
- Political spying targeted at government ministries and national intelligence can HAVE many sensitive operations planned for the future.
- Besides corporations, banks, the military, intelligence, etc., the individual too has increasingly been a target.
- Leakage of personal information such as ***credit card numbers, passwords, and even personal spending habits are common and are collectively referred to as identity theft.*** Such information is advertised on certain websites and may be purchased for a small fee.

2. Disruption of service.

- Interruption or disruption of service is launched against an organization's servers so they are made unavailable or inaccessible.
- In recent times, there have been unconfirmed reports of such attacks being launched by business rivals of e-commerce websites.

- The goal here appears to be "***my competitor's loss is my gain."*** In 2001, there were a series of such attacks that targeted the websites of Yahoo, Microsoft, etc. in a short span of time.
- They were meant to alert corporates and others of the dangers of this class of attacks.

3. Illegal access to or use of resources.

- The goal here is to obtain free access or service to paid services.
- Examples of this include free access to online digital products such as magazine or journal articles, free talk time on someone else's account, free use of computing power on a supercomputer, etc.
- In each case, the attacker is able to circumvent controls that permit access to only paid subscribers of such services.

1.1.2 Common Attacks

Some of the common attacks are :

1. **Phishing**
2. **Pharming**
3. **Dictionary attacks**
4. **Denial of Service (dos)**
5. **Trojan**
6. **Spyware**

1. Phishing:
   - One set of attacks are those that attempt to ***retrieve personal information*** from an ***individual.***
   - It ***provokes the victims to a fake website*** — an on-line bank, for example.
   - The fake site has the look and feel of the authentic bank with which the victim has an account.
   - The victim is then asked to enter sensitive information such as his/her login name and password, which are then passed on to the fake website.
   - Personal information may also be leaked out from credit cards, smart cards, and ATM cards through a variety of skimming attacks.
2. Pharming:
   - It attempts to deduce ***sensitive information from lost or stolen smart cards*** through advanced power and timing measurements conducted on them.
   - Finally***, leakage of information*** may also take place through **eavesdropping or snooping** on the link between two communicating parties.
3. Dictionary attacks :
   - One means of intruding into a computer system is through password-guessing attacks.
   - The ultimate goal of the attacker is ***to impersonate his/her victim***.
   - The attacker can then perform unauthorized logins (break-ins), make on-line purchases, initiate banking transactions, etc., all under the assumed identity of the victim.

4. Denial of Service (DoS):
   - Denial of Service (DoS) means the attacker performs a ***interruption or disruption of the computing services on a system*** .
   - These attacks exhaust the ***computing power, memory capacity, or communication bandwidth*** of their targets so they are rendered unavailable.
   - One version of this attack causes website defacement.
   - At various times, the websites of high-profile targets such as the American president or various government ministries have been targeted.
   - To prevent such attacks an ***alarm*** being raised,
   - Dos attack on a web server slows down the web server so that its response time to requests from the outside world is unacceptably high.
5. Malware.
   - Worms and viruses are malware that replicate themselves.
   - A virus typically infects a file, so a virus spreads from one file to another.
   - A worm is usually a stand-alone program that infects a computer, so a worm spreads from one computer to another.
   - Worms and viruses use various spreading techniques and media — e-mail, Internet messages, web pages, Bluetooth, and MMS are some of the propagation vectors.
   - Trojan:A trojan is a kind of malware that masquerades as a utility but has other goals such as the modification of files, data theft, etc.
   - Spyware, installed on a machine, can be used to monitor user activity and as a key logger to recover valuable information such as passwords from user keystrokes.

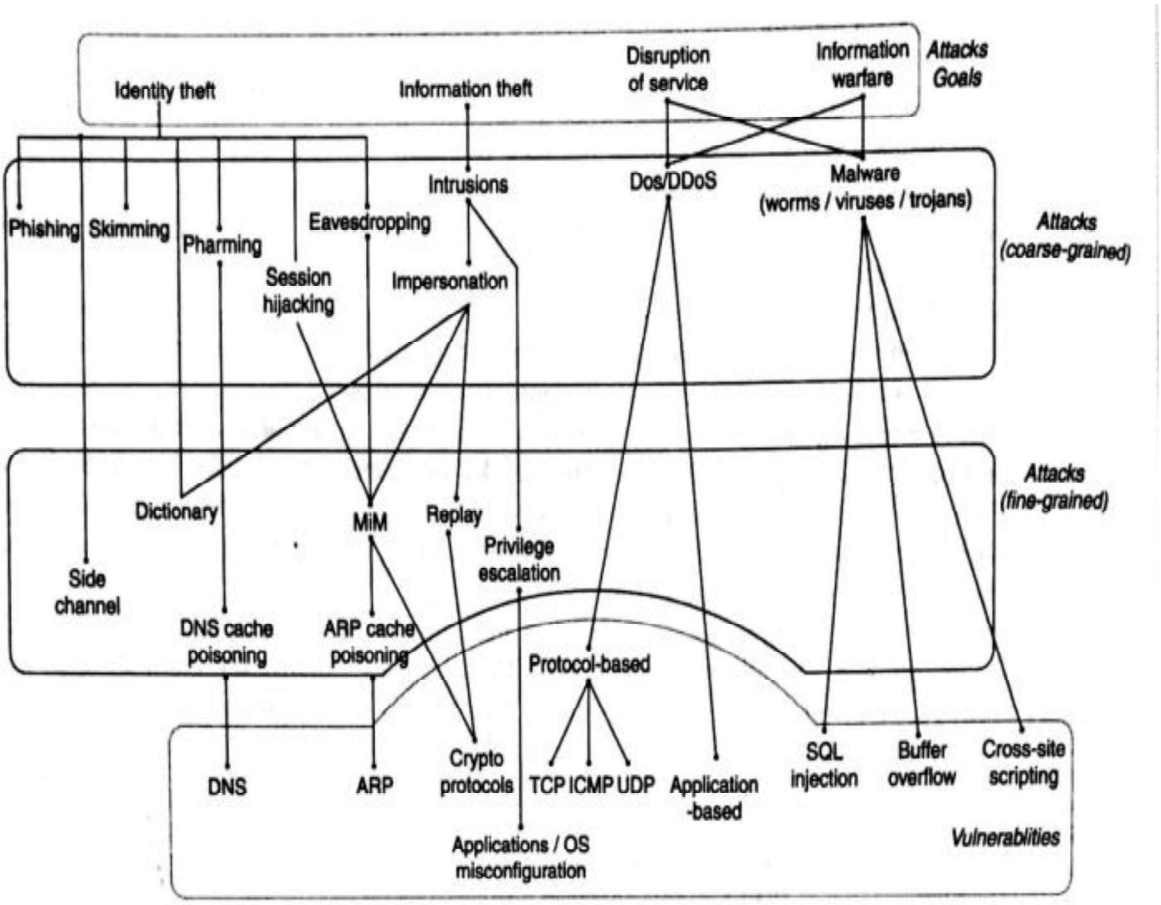


Figure 1.1 common attacks and vulnerabilities

1.1.3 Vulnerabilities

- Behind every attack is a vulnerability of some type or the other.
- Definition:A vulnerability is a ***weakness*** in a ***procedure, protocol, hardware, or software*** within an organization that has the potential to cause damage.
- There are at least **four important vulnerability classes in the domain of security:**
  1. **Human Vulnerabilities:**
     - These are vulnerabilities caused by human behaviour or action.
     - For example, the user clicks on a link in an e-mail message received from a questionable source. By so doing, the user can be directed to a site controlled by the attacker as in a **phishing attack or a cross-site scripting attack**.
     - Similarly clicking on an e-mail attachment may open up a document causing a **macro to be executed.**
     - The **macro may be designed to infect other files** on the system and/or spread the infected e-mail to other e-mail addresses harvested from the victim's inbox.

- In both these cases, the **human vulnerability consists of clicking on a link or attachment in an e-mail from a possibly unknown source.**
- The **link or attachment** may have provoked the victim by a flashy message suggesting quick money, etc., blinding him/her to the fact that the message came from an unknown source.
- It is actions like this that make a phishing attack or an e-mail virus so very successful.

2. **Protocol Vulnerabilities:**
   - A number of networking protocols including **TCP, IP, ARP, ICMP, UDP, DNS**, and various protocols used in local area networks (LANs) have features that have been used in unanticipated ways to craft assorted attacks.
   - Pharming attacks and various ***hijacking attacks*** are some examples.
   - There are tools available ***on-line to facilitate some of these attacks.***
   - One such tool subverts the normal functioning of the ARP protocol to sniff passwords from a LAN.
   - There are a number of vulnerabilities in the ***design of security protocols*** that lead to ***replay or man-in-the-middle attacks.***
   - These attacks, in turn, lead to identity theft, compromise of secret keys, etc.
   - Vulnerabilities in network protocols are often related to aspects of their design though they may also be the result of poor implementation or improper deployment.
3. **Software Vulnerabilities:**
   - This family of vulnerabilities is caused by written system or application software.
   - In many cases, the causes of the problem seems to be the code that is all too trusting of user input.
   - Ex A web server accepts input from a users browser.the web server must accept the request after typing the complete username and password.the server software should perform sufficient validation.
4. **Configuration Vulnerabilities:**
   - These relate to configuration settings on newly installed applications, files, etc.
   - ***Read-write-execute*** permissions on files may be too generous and susceptible to abuse.
   - The privilege level assigned to a process may be higher than what it should be to carry out a task. This privilege may be misused during some point in its execution leading to what are commonly called "privilege escalation" attacks.
   - Besides misconfiguration of software and services, security appliances such as firewalls may be incorrectly or incompletely configured with possibly devastating effect.

1.2 DEFENCE STRATEGIES AND TECHNIQUES

1.2.1 Access Control—Authentication and Authorization

- The first defence strategy to prevent intrusions is access control.
- This implies the existence of a trusted third party that mediates access to a protected system.
- The trusted third party is typically implemented in software and may be a part of the operating system and/or the application.
- The first step in access control is to ***permit or deny entry into the system***.
- This involves some form of authentication — a process whereby the subject or principal (the party attempting to login) establishes that it is indeed ***the entity it claims to be.***
- One form of authentication is the humble password.
- Example:The principal first enters his/her login name. By prompting him/her to enter his/her password, the system implicitly challenges the principal to prove his/her identity.
- In this simple case, knowledge of the secret password constitutes "proof of identity."
- After successful authentication, a subject is logged into the system. The subject may need to access several resources such as files.

1.2.2 Data Protection

- The data in transit or in storage needs to be protected.
- It implies data confidentiality – the data should not be readable by an intruder.
- Another dimension to data protection is the preservation of **data integrity**.
- This implies that the data while transmitting should not be ***tampered or modified***
- Cryptographic techniques are among the best known ways to protect both, the confidentiality and integrity of data.
- Cryptography is the science of disguising data and is the subject of the part of this book.
- The encryption operation is performed by the sender which converts the plain text to ciphertext.
- decryption operation is performed by the receiver which converts the ciphertext to plaintext.
- The encryption and decryption operations both use the same secret key known only to the sender and receiver.
- This prevents an eavesdropper from decrypting the encrypted message.
- the computation of the cryptographic checksum uses a secret shared by the sender and receiver.
- The sender computes the checksum as a "one-way function" of the message and secret. It transmits the message and checksum.
- The receiver also computes the checksum. If the computed checksum matches that received, the receiver concludes that there is no error in the received message.

1.2.3 Prevention and Detection

- Access control and message encryption are preventive strategies.
- Authentication keeps intruders out, while authorization limits what can be done by those who have been allowed in.
- Encryption prevents intruders from eavesdropping on messages.
- The cryptographic checksum, on the other hand, detects tampering of messages.
- In the important domain of software security, code testing is used to detect vulnerabilities.
- Blackbox testing is employed when the source code of a program is not available. The goal here is to determine whether the software has been carefully designed to handle unexpected or malicious input.
- For greater assurance of secure software, whitebox testing should be employed. Here, the security engineer has access to source code and can perform more elaborate testing by exercising different control paths in the source code.
- intrusion preventive techniques can be used to detect anomalous behavior, Continuous monitoring of network logs and operating system logs
- Intrusion detection systems also look for certain patterns of behaviour.
- For example, multiple instances of a given worm often exhibit a characteristic bit pattern called a worm signature.

1.2.4 Response, Recovery, and Forensics

- Once an attack or infection has been detected, response measures should be taken .
- These include shutting down all or part of the system.
- Many intrusion attempts leave information
- Cyber forensics is an emerging discipline with a set of tools that help trace back the perpetrators of cyber crime.
- Table 1.2 defines some of the most widely used terms in cyber security parlance.

Table 1.2 *Definitions of commonly used terms in security parlance*

- *Security policy* is the set of rules and practices that regulate how an organization manages and protects its computing and communication resources from unauthorized use or misuse.
- *A security mechanism* is a technique or device used to implement a security policy.
- A *vulnerability* is a weakness or flaw in the architecture, implementation, or operational procedures of a system that could be exploited to cause loss or failure.
- Exploitation of a vulnerability with malicious intent leads to a cyber *attack*.
- *Access control* is the process of preventing unauthorized access to a computing or communication resource.
- *Authorization* involves granting a specific entity or process the permission to access restricted data or perform a restricted operation.
- *Auditing* is the process of collecting and analyzing relevant information in order to ensure compliance with security policies laid out for an organization.

One or more of the following are implicit when we talk about a secure connection or session between two parties:

- *Entity authentication* is the process of verifying that the entity being communicated with is indeed the entity it claims to be.
- *Message authentication* is the process of verifying the source or origin of the received message.
- *Confidentiality* is the protection of data from disclosure to an unauthorized party or process.
- *Integrity* is the assurance that data has not been modified, tampered with, or made inconsistent in any way.
- *Non-repudiation* offers a guarantee against repudiation or denial by a party of the fact that it created or sent a particular message.

1.3GUIDING PRINCIPLES

1. **Security is as much (or more) a human problem than a technological problem and must be addressed at different levels.**
   - At the highest level, security should be addressed by top-level management in large organizations.
   - Robust security policies should be formulated and a comprehensive implementation strategy outlined by a dedicated team of security specialists, possibly headed by a Chief Information Security Officer (CISO).
   - Some of the mechanisms used to implement high-level policies are in the realm of technology.
   - Security engineers have a key role to play in designing techniques and products to protect organizations from the various cyber attacks.
   - System administrators handle day-to-day operations.
   - They should be proactive in crucial security practices such as patch application.
   - One of the key tasks of a system administrator is to configure systems and applications. Their job also involves setting user/group permissions to various system resources such as files, configuring firewalls, sifting through system logs for signs of an intrusion, and processing alerts.
   - The final link in the security chain is the rank and file within an organization.
   - The employees within an organization should be educated on various do's and don'ts through periodically updated security awareness programs.
   - In summary, a healthy combination of enlightened security policy and procedures, backed by enforcement, aided by technology, coupled with diligent

participation of administrators and employees, and presided over by an empowered CISO is the surest insurance against cyber attacks.

2. **Security should be factored in at inception, not as an afterthought.**
   - No one then had thought that those protocols would be abused by attackers in so many creative ways!
   - application software (web software, for example) developed today continues to be often vulnerable to numerous attacks such as cross-site scripting and SQL injection attacks.
   - The solution lies, at least in part, in integrating secure coding practices into the software curriculum in our colleges and universities.
   - In general, security should be factored in early on during the design phase of a new product and then carried forward right through implementation and testing.
   - The product could be a networking protocol, a new version of an operating system, a piece of application software, or the architectural layout of computing infrastructure for an enterprise.

3. Security by obscurity (or by complexity) is often bogus.

   - There have been a number of cryptographic algorithms proposed which was made mandatory in newly standardized protocols, but their details were not made public.
   - The flaws are exposed over time after the protocols have been widely deployed, attracting closer attention from the hacker community.
   - There are ethical hackers whose goal is to break software/ protocols/algorithms so that they can be fixed before things get out of hand.
   - It is the ethical hacker community at least, if not the public at large, who should be able to study new protocols and algorithms prior to widespread adoption.
   - One such example was the procedure followed for selecting an algorithm in the late 1990s for the new secret key cryptography standard — AES was finally chosen after much public scrutiny and debate. As another example, open source software is usually freely available. Public review of its security features can make or break its reputation.

4. **Always consider the "Default Deny" policy for adoption in access control.**
   - The subjects in an access control policy could be people, network packets, operating system processes or even user input.
   - One policy is the "Default Permit," i.e., grant the subject's request unless the subject is on a blacklist or it has certain blacklisted attributes.
   - The dual of this policy is the "Default Deny" policy. In this case, the subject's request is denied unless it is on a whitelist.
   - Clearly, whitelisting is the more conservative approach.

- With whitelisting, the access controller may reject a legitimate subject whose name has been mistakenly excluded from the whitelist but that is the price to be paid for greater security.
- Blacklisting, on the other hand, may accept a bad guy because his name or attributes were mistakenly excluded from the blacklist.
- The tradeoffs between blacklisting and whitelisting should be carefully examined (see Principle 8). However, in general, prudent security design should seriously consider adoption of the "Default Deny" policy.

5. An entity should be given the least amount/level of permissions/privileges to accomplish a given task.

- Role-based access control (RBAC) has influenced a variety of software platforms ranging from operating systems to database management systems.
- The principal idea in RBAC is that the mapping between roles and permissions is paramount.
- The role played by an individual at a given point in time determines the rights or privileges the individual has.
- Conferring higher privilege on an individual than what is warranted by his/her current role could compromise the system.
- Privilege escalation in its different manifestations has caused many security breaches in computer systems.
- The problem often lies in sloppy or incomplete configuration management.
- In publicly accessible servers within an organization such as the web and e-mail servers, unnecessary services hosted by them can open the door to malware, which can compromise those servers. The latter are then used as a springboard to spread to the internal machines in that organization.

6. **Use 'Defence in depth' to enhance security of an architectural design**.

- This principle is used in many high-security installations and has been recently introduced in some airports. A passenger's ticket is checked before entering the airport terminal building. This is followed by verification of travel documents and inspection of check-in baggage at the airline counter. Next comes a security check (physical) and a further check of the boarding pass, travel documents, and check-in baggage before entering the boarding area (main concourse).
- Defence in depth is applicable to cyber security as well.
- Consider designing the firewall architecture for a mid-to-large size enterprise.
- Every packet from the outside (Internet) should be intercepted by at least two firewalls.
- The firewalls may be from two different vendors and would, preferably, have been configured by two different system administrators.

- They may, and typically do, have some overlapping functionality. Because of differences in the hardware/software design and in configuration, what escapes Firewall 1 may be caught by Firewall 2 and vice versa.

7. Identify vulnerabilities and respond appropriately.

- We have already seen a large number of vulnerability types.
- Vulnerabilities in software or protocols are well researched.
- But equally important are weakness/shortcomings in policy, procedures, and operations.
- How many organizations are geared to implement policies regarding the entry of visitors' laptops and PDAs?
- Or do they even have such policies in place? Such mobile devices and Bluetooth-enabled gadgets may transmit malware to unsuspecting stations within the organization.
- Likewise, USB-enabled PCs may be victims of viruses residing on USB flash drives.
- Often, these organizations have elaborate security infrastructure in the form of firewalls and intrusion detection systems. They securely guard the high-profile main entrance but blissfully ignore the security requirements of the less conspicuous side and rear doors.
- Vulnerability detection and response brings to mind fast-spreading Internet scanning worms.

8. Carefully study the tradeoffs involving security before making any.

- Engineering design often involves making tradeoffs — cost versus performance, functionality versus chip area, etc.
- The previous principle highlighted an important tradeoff — security versus cost.
- Consider, for example, the area of electronic payment involving small purchases (say Rs. 10 or less). Such payments, called micropayments, may be made for digital goods such as on-line news-paper articles.
- Payment schemes use some form of cryptography. The cryptographic overheads of these schemes, in terms of computation cost, can be high. Can we use cheaper (lower overhead) cryptography for micropayments? The downside here is that such cryptography is not as secure. But given the transaction amount, the risk of fraud is probably acceptable.
- In this case, we may be justified in trading off increased security for lower cost. Besides security versus cost, security versus performance is a tradeoff often encountered.

Basics of Cryptography

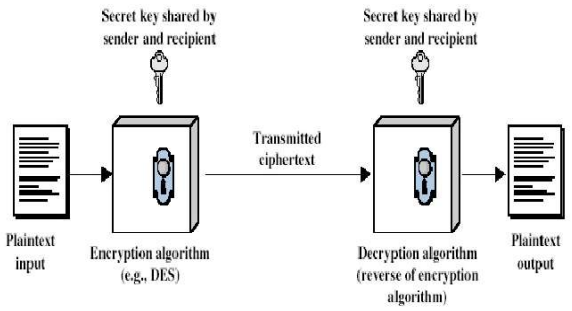
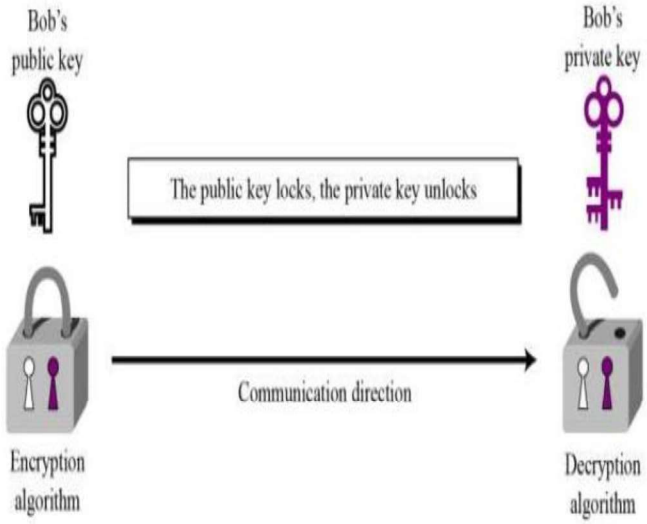
PRELIMINARIES

- Cryptography is the science of disguising messages so that only the intended recipient can decipher the received message.
- Cryptography is the lynchpin of data security — besides providing for message confidentiality, it also helps in providing message integrity, authentication, and digital signatures.
- The original message or document to be transferred is called plaintext
- The plaintext which is encrypted is called ciphertext.
- The process of converting the original plaintext to ciphertext is called encryption
- The process of recovering the original plaintext from the ciphertext is called decryption.
- Encryption involves the use of an encryption function or algorithm, denoted by E, and an encryption key, e.
- Decryption involves the use of a decryption function denoted by D, and a decryption key, d.
- These operations are summarized below.
- $c = E_e(p)$
- $p = D_d(c)$
- Here, p denotes a block of plaintext. It is encrypted by the sender to produce ciphertext denoted by c.
- Decryption operation is performed by the receiver on the ciphertext to recover the plaintext.
- **Kerckhoff's Principle: The secrecy should be in the key used for decryption, not in the decryption or encryption algorithms.**

4.1.1 Secret versus "Public" Key Cryptography

- There **are two types of cryptography** in widespread use –

1. Secret key cryptography	**2. Public key cryptography.**
➢ In secret key cryptography, **both sender and receiver share a common secret - the same secret key** is used for encryption as well as decryption. So e = d , this form of cryptography is also referred to as ***symmetric key cryptography.***	➢ In public key cryptography, two distinct keys forming a key pair are used – I. the encryption key or public key and II. the decryption key or private key. ➢ The public key of a user(receiver) is used to encrypt messages to that user. ➢ It is the private key of the recipient that is used to decrypt the message. ➢ Because the public and private keys are distinct, this form of cryptography is also referred to ***as asymmetric key cryptography.***
➢ If Alka and Brijesh share a secret key, k,	➢ Assuming that Brijesh has a public key-private

then she encrypts the message using the common secret. ➢ The encrypted message received by Brijesh is decrypted using the same secret. ➢ The secret key operations are summarized below. ➢ **Operation performed by Alka** ➢ **$c = E_k(p)$** ➢ **Operation performed by Brijesh:** ➢ **$p = D_k(c)$**	key pair, she would encrypt her message using his public key **, $B._{pu}$.** ➢ Brijesh then decrypts the message using the corresponding private key, **$B._{pr}$.** ➢ Assuming that Brijesh keeps his private key securely, he and only he can decrypt the message received from Alka. ➢ The public key-private key operations are summarized below. ➢ Operation performed by Alka: ➢ **$c = E_{B.pu}(p)$** ➢ Operation performed by Brijesh: ➢ **$p=D_{B.pr}(c)$** ➢
➢ EX:Data Encryption Standard, Advanced Encryption Standard (AES)	RSA, Elliptic Curve Cryptography (ECC).
	

4.1.2 Types of Attacks

- At a very high level, a cryptographic algorithm is secure if a cryptanalyst (a person with expertise in breaking ciphers) is unable to
- **(a) obtain the corresponding plaintext from a given ciphertext.**
- **(b) deduce the secret key or the private key**
- How would the attacker proceed to realize the above objectives? He could accumulate copious amounts of ciphertext.

- He would then look for patterns in the ciphertext in an attempt to reconstruct some plaintext and/or deduce the key. Such an attack which exclusively uses ciphertext is referred to as a **"known ciphertext" attack.**
- Occasionally, all or part of some plaintext blocks are predictable or may be guessed.
- A cryptanalyst may then build a list of corresponding plaintext, ciphertext pairs with the intention of deducing the key. Such an attack is referred to as a **"known plaintext" attack.**
- It may even be possible for a shrewd attacker to carefully choose pieces of plaintext and then induce the sender to encrypt such text.
- An attack on a cryptographic scheme which makes use of pairs of attacker-chosen plaintext and the corresponding ciphertext is referred to as a **"chosen plaintext" attack.**
- The most obvious, though compute-intensive, attack with known plaintext is a **brute force** attempt at obtaining the key by trying all possible key values.
- Let (p_1,c_1) , (p_2,c_2) , (p_3,c_3) be plaintext—ciphertext pairs.

```
for (each potential key value, k in the key space)
{
        proceed = true;
         i= 1;
        while (proceed = = true && i < m)
        {
                if (ci, ≠ Ek( pi))
                {
                        proceed = false;
                         i ++ ;
                }
                if (i = m+1)
                 print (" Key Value is k");
}
```

4.2 ELEMENTARY SUBSTITUTION CIPHERS

4.2.1 Monoalphabetic Ciphers

- The most basic cipher is a substitution cipher.
- For ease of understanding, we consider English text in all the examples in this chapter.
- Let E denote the set of alphabets, (A, B, . . . Z).
- A monoalphabetic substitution cipher defines a permutation of the elements in ∑.
- There are 26! permutations; so, there are 26! possible monoalphabetic substitution ciphers.
- The simplest substitution cipher is one that replaces each alphabet in a text by the alphabet k positions away (in the modulo 26 sense).
- For k = 3, the substitutions are

  D for A,

E for B,
A for X,
B for Y, etc.

- Such a scheme is referred to as a **Caesar cipher**.
- A sample plaintext and the corresponding ciphertext for k= 3 is

Plaintext: WHAT IS THE POPULATION OF MARS

Ciphertext: ZKDW LV WKH SRSXODWLRQ RI PDUV

- In substitution ciphers, like the Caesar cipher, each letter is always substituted for another unique letter. Such ciphers are said to be monoalphabetic.

4.2.2 Polyalphabetic Ciphers

- In a polyalphabetic cipher, the ciphertext corresponding to a particular character in the plaintext is not fixed. It may depend on, for example, its position in the block.
- We next study two examples of such ciphers.

a. The Vigenere Cipher

- The Vigenere cipher is a polyalphabetic cipher that uses a multi-digit key $\mathbf{k_1}$, $\mathbf{k_2}$, $\mathbf{k_3}$, $\mathbf{k_4}$... $\mathbf{k_m}$
- Here , $\mathbf{k_1}$, $\mathbf{k_2}$, $\mathbf{k_3}$, $\mathbf{k_4}$... $\mathbf{k_m}$ are each integers.
- The plaintext is split into non-overlapping blocks, each containing **m consecutive characters.**
- Then the first letter of each block is replaced by the letter $\mathbf{k_1}$ positions to its right the second letter of each block is replaced by the letter $\mathbf{k_2}$ positions to its right, and soon.

Plain text	W	I	S	H	I	N	G		Y	O	U		S	U	C	C	E	S	S

- Key: **04 19 03 22 07 12 05 11 04 19 03 22 07 12 05 11 4**
- Ciphertext: **A B V D P Y L J S N P Q J T X F G V H O Z**
- The first letter in the above text is W. The corresponding key value is 04.
- This means that the ciphertext is the letter 4 positions ahead (in the modulo 26 sense).
- The key length = 8, i.e., the keystring repeats after every **8 characters.**
- There are **four occurrences** of the letter "s" in the above text
- However, each occurrence of "s" is encrypted as a different character in the ciphertext - " V", "X", "O," and "Z".

b. The Hill Cipher

- The Hill cipher is another **polyalphabetic cipher** proposed by **Lester Hill.**
- As in the Vigenere cipher, the plaintext is broken into blocks of size m. However, the key in the Hill cipher is an in m x m matrix of integers between 0 and 25.

- Unlike the Caesar and Vigenere ciphers, each character in the ciphertext is a function of all the characters in that block.
- Let p1,, p2, . . . pm,,, be the numeric representation of the characters in the plaintext and
- let c1,c2,c3,…cm represent the corresponding characters in the ciphertext.
- To compute the ciphertext, we map each alphabet to an integer.
- We use the mapping,

A	0
B	1
C	2
D	3
E	4
F	5

- The relationship between a block of plaintext and its ciphertext is expressed by

$$
\begin{aligned}
c_1 &= p_1 k_{11} + p_2 k_{21} + \ldots + p_m k_{m1} \bmod 26 \\
c_2 &= p_1 k_{12} + p_2 k_{22} + \ldots + p_m k_{m2} \bmod 26 \\
&\ldots \\
&\ldots \\
c_m &= p_1 k_{1m} + p_2 k_{2m} + \ldots + p_m k_{mm} \bmod 26
\end{aligned}
$$

- This can be conveniently written as
- C= p K
- Here, C and p are row vectors corresponding to the plaintext and ciphertext, respectively, and K is the m x m in matrix comprising the key.
- At the receiver end, the plaintext can be recovered from the ciphertext by using
- $p = cK^{-1}$

(refer problem solved in class)

c.One—time Pad

- To perform theone time pad cipher encryption operation, the pad values are added to numeric values that represent the plaintext that needs to be encrypted.
- Each character of the plaintext is turned into a number and a pad value for that position is added to it.
- The resulting sum for that character is then converted back to a ciphertext letter for transmission.
-

Plaintext:	S	A	C	K	G	A	U	L	S	P	A	R	E	N	O	O	N	E
Plaintext value:	19	01	03	11	07	01	21	12	19	16	01	18	05	14	15	15	14	05
One-time pad text:	F	P	Q	R	N	S	B	I	E	H	T	Z	L	A	C	D	G	J
One time pad value:	06	16	17	18	14	19	02	09	05	08	20	26	12	01	03	04	07	10
Sum of plaintext and pad:	25	17	20	29	21	20	23	21	24	24	21	44	17	15	18	19	21	15
After modulo Subtraction:				03								18						
Ciphertext:	Y	Q	T	C	U	T	W	U	X	X	U	R	Q	O	R	S	U	O

4.3 ELEMENTARY TRANSPOSITION CIPHERS

- A transposition cipher shuffles, rearranges, or permutes the bits in a block of plaintext.
- Unlike a substitution cipher, the number of 0's and 1's in a block does not change after the shuffling.
- For simplicity, we work with characters (letters) rather than bits.
- Imagine a block of plaintext arranged in a matrix row by row as below.
- **Plaintext: Begin Operation at Noon (any case)**

$$\begin{bmatrix} b & e & g & i \\ n & o & p & e \\ r & a & t & i \\ o & n & a & t \\ n & o & o & n \end{bmatrix}$$

- rearrange the rows as follows
- ROW 1 to row 3
- Row 2 to row 5
- ROW 3 to row 2,
- Row 4 to row 1
- Row 5 row 4.
- The resulting matrix is
- $$\begin{bmatrix} o & n & a & t \\ r & a & t & i \\ b & e & g & i \\ n & o & o & n \\ n & o & p & e \end{bmatrix}$$
- now rearrange the columns as follows
- Column 1 to column 4 ,
- Column 2 to column 3,
- Column 3 to column 1,
- Column 4 to column 2.
- The resulting matrix is

- $\begin{bmatrix} a & t & n & o \\ t & i & a & r \\ g & i & e & b \\ o & n & o & n \\ p & e & o & n \end{bmatrix}$

- The ciphertext thus generated is

- **A T N O T I A R G I E B O N O N P E O N**

- To decrypt the message, the recipient would have to cast the cipher text in a 5 x 4 matrix, reverse the column shuffles, and then reverse the row shuffles.
- For example, with a combination of guesswork, luck, and limited prior information, a spy might be able to deduce that the planned start time of an attack is 11:15 pm upon receiving the following ciphertext.
- **1 1 K C T A T A M M O C P M 5 1 C E N E**
- This is the ciphertext using the row and column shuffling as in the example above.
- The corresponding plaintext is

Commence Attack 11 15 pm

4.4 OTHER CIPHER PROPERTIES

4.4.1 Confusion and Diffusion

- In 1949, Claude Shannon first proposed the ideas of confusion and diffusion in the operation of a cipher.
- Confusion is the property of a cipher whereby it provides no clue regarding the relationship between the ciphertext and the key.
- Given plaintext p, a sequence of keys k1, k2, . . . ki and the corresponding ciphertexts are obtained using this encryption $\mathbf{E_{k1}(p), E_{k2}(p), E_{k3}(p),\ldots\ldots\ldots E_{ki}(p),}$
- It is nearly impossible to deduce the value of a new, arbitrarily chosen key kj used to create the ciphertext,Ekj(p).
- Confusion reigns supreme with a cipher if , for any plaintext,p if even a single bit in a key k is changed to produce k', then roughly half the bits in the ciphertexts Ek(p) and Ek'(p) are different.
- While confusion is concerned with the relationship between the key and the ciphertext,
- Diffusion is concerned with the relationship between the plaintext and the corresponding ciphertext.

4.4.2 Block Ciphers and Stream Ciphers

Block Ciphers

- With block ciphers, the plaintext is split into fixed size chunks called blocks, and each block is encrypted separately.
- Typically all blocks in the plaintext are encrypted using the same key.
- Block ciphers include DES, AES, RSA, and ECC.
- Block sizes used in secret key cryptography are usually smaller — 64 bits in DES and 128 bits in AES.
- The block size in RSA is much larger — 768 or more bits, while the block size in ECC is about 200 bits.
- If two blocks of plaintext within a message are identical, their corresponding ciphertexts are identical. This statement, however, is only partially true.

Stream cipher

- Stream ciphers typically operate on bits.
- The one-time pad is an example of a stream cipher.
- Practical stream ciphers typically generate a pseudo-random keystream which is a function of a fixed length key and a per-message bit string.
- The key is known to both the sender and the receiver.
- The per-message string could be a message sequence number.
- Alternatively, it could be a random number generated by the sender and transmitted to the receiver along with the encrypted message.
- The ciphertext is itself obtained by performing an $\oplus$ operation between the plaintext and the keystream.
- An example of a stream cipher is RC4 used in the wireless LAN protocol, IEEE 802.11.
- Stream ciphers are usually faster than block ciphers and use less complicated circuits. However, RC4 and some other stream ciphers have been shown to be vulnerable to attack.

Secret key cryptography

5.1 PRODUCT CIPHERS

- Modern day secret-key ciphers are typically synthesized using the Substitution Box (S-Box) and the Permutation Box (P-Box).

- **Substitution Box (S-Box)**
  - An S-box is a device that takes as **input a (binary) string of length m** and **returns a (binary) string 1 of length n.** While it is often the case that m = n, this need not always be so.
  - An S-box is implemented using a table (or array) of $\mathbf{2^m}$ rows with each row containing an **n-bit value.**
  - The **input to the S-**box is used to index the table which returns the **n-bit** output of the **S-Box.**
- **Permutation Box (P-Box).**
  - A P-Box performs a permutation or re-arrangement of the bits in the input.
  - A permutation is more restrictive than a substitution.
  - For example, the number of zeros in the output of the P-Box is equal to the number of zeros in its input while an S-box imposes no such restriction.
  - A P-Box or S-box by itself is not sufficiently powerful to create a secure cipher. However, cascading P-Boxes and S-Boxes alternately, the strength of a cipher can be greatly increased. Such a cipher is referred to as a **product cipher.**

- The three operations that take place in sequence as shown in Fig. 5.1:

  (1) An Operation Involving A Function Of The Encryption Key
  (2) A Substitution
  (3) A Permutation

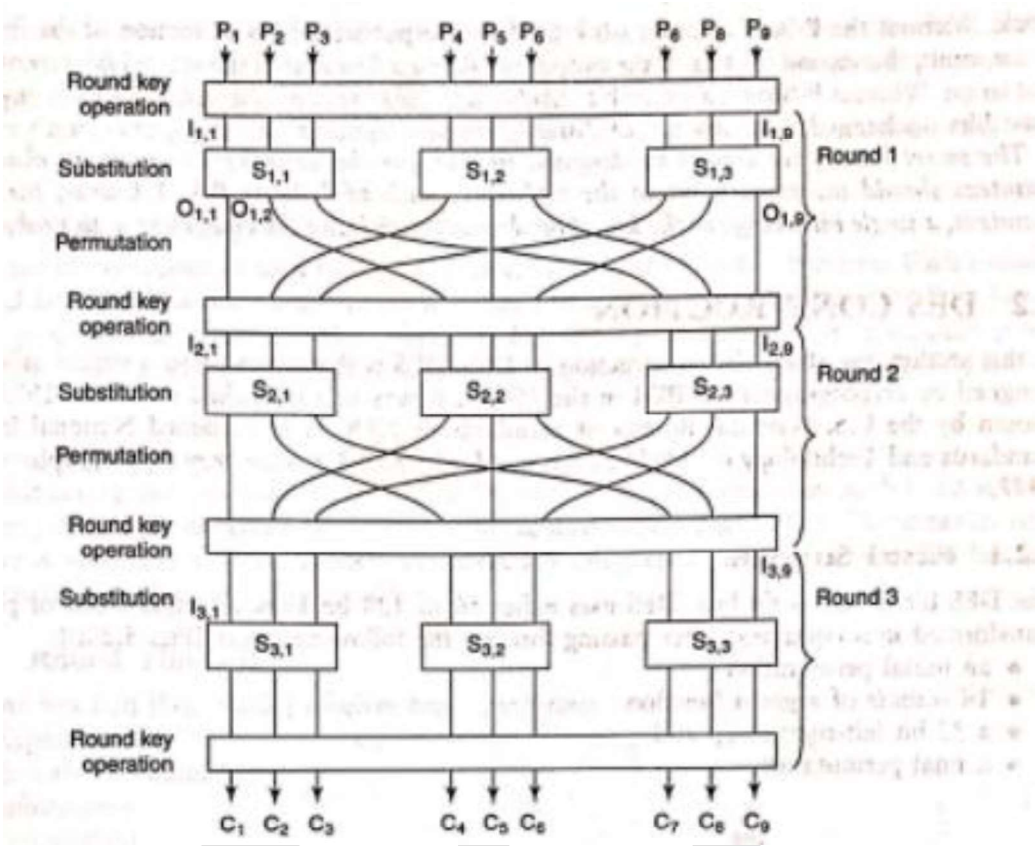


Figure: Three-round SPN network

- These operations are repeated over many rounds or iterations.
- Of the three operations, the first is the only one that involves the e**ncryption key.**
- It is usually an ⊕(ex or) of the input with the "round" key.
- Each round key is a function of the bits in the encryption key.
- the S-box is usually implemented as a table.
- If the block size of the cipher is b, the size of the table that implements a b x b
- S-box is b x 2^b bits.
- Thus, the table size increases exponentially with the number of inputs.
- As an example, for b = 64, the size of the table is 270 bits which is a thousand billion billion bits!
- To save table space, a single S-box is broken into multiple S-boxes as shown in each round of Fig. 5.1.
- If s is the number of S-boxes, the number of inputs to each S-box is **b/s.**
- Each S-box is now implemented using a table of **size $(b/s)2^{b/s}$ bits.**
- Thus, the total size of all the **S-boxes is b x $2^{b/s}$ bits.**

- For a block size of 64, the use of eight S-boxes (each with 8 inputs) would bring down the storage requirements to about 16,000 bits.
- Usage of s box injects non-linearity into the design of the cipher.
- Non-linearity implies the absence of a linear relationship between any subset of bits in the plaintext, cipher text, and key.
- Finally, the third step in each round or iteration is a permutation.
- A P-Box re-orders the inputs that it receives. it diffuses or spreads contiguous bits of the input across the entire block.
- Without the P-Box, the first b/s bits of the output would be a function of the first b/s bits of the input, the second b/s bits of the output would be a function of the second b/s bits of the input and so on.

5.2 DES CONSTRUCTION

- DES is the successor to a cipher called Lucifer designed by cryptographers at IBM in the 1960's.
- It was first published in March 1975 and was chosen by the U.S. National Bureau of Standards or NBS (later re-named National Institute of Standards and Technology or NIST) as the standard cipher for secret key cryptography in January 1977.

5.2.1 Fiestel Structure

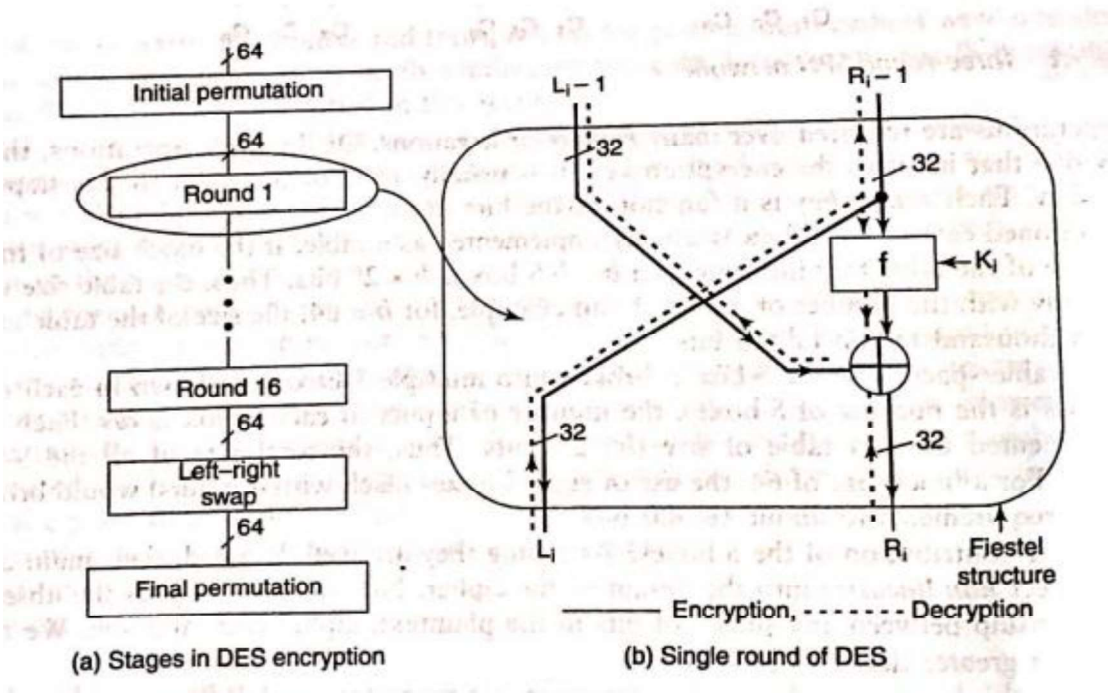


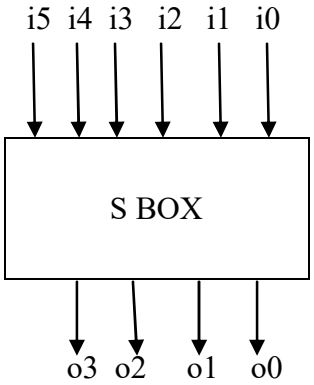
Figure:DES encryption

- The DES data block size is **64 bits**.
- DES uses either **56 or 128 bit keys**.
- A single block of plaintext is transformed into ciphertext after passing through the following stages as shown in above figure:
    1. **An initial permutation**
    2. **16 rounds of a given function**
    3. **a 32-bit left-right swap and**
    4. **a final permutation**
- Each of the 16 rounds is functionally identical.
- The structure of each DES round is explained below.
- Let **L_{i-1} and R_{i-1} be the left and right halves of the input to round i.**
- As shown in above figure:
- **$L_i = R_{i-1}$**
- **$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$**
- The function f is applied at each round and is referred to as the **"round" function.**
- Each round uses a round key, which is one of the inputs to f.
- Each round key is derived from the DES key.
- The process of decryption involves obtaining Li-1 and Ri-1 from Li and Ri.Execution proceeds from bottom to top and is summarized by the following equations derived from above Eqs :
- **$R_{i-1} = L_i$**
- **$L_{i-1} = R_i \oplus f(L_i, K_i)$**
- The structure of such a cipher is attributed to Horst Feistel (one of the key designers of DES). A cipher that has such a structure is referred to as a **Feistel cipher.**

5.2.2 Round Function

- A round function [above figure (b)] involves four operations:
    1) **Expansion**
    2) **⊕ with the round key**
    3) **Substitution**
    4) **Permutation**
- The input to the round function is Ri-1, a 32-bit quantity [Fig.(b)].
- This is first expanded into **48 bits** by repeating some bits and interchanging their positions.
- The 48-bit quantity is then **⊕ed with the round key, Ki**. (which is different for each round).
- The bits in a round key are a function of the bits in the original 56-bit key.
- The result of the ⊕ operation is divided into **eight 6-bit chunks.**
- Each chunk is substituted by a **4-bit chunk**
- A total of 8 different S-boxes provide the eight substitutions.
- An S-box is implemented using a **4 x 16 array**.
- Each row of the array is a permutation of the numbers 0 through 15.

- Two bits of the i-th chunk serve as a row index (i5,i0)into the i-th table (Fig. 5.3) and the remaining four bits serve as a column index(i4,i3,i2,i1).
- The output of the S-box is simply the 4-bit string pointed to by the row and column indices.



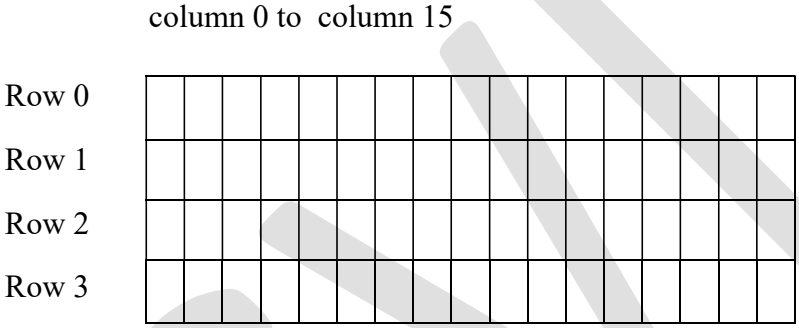


Figure s box implementation using array size 4X16

Module 2

Cryptographic Hash

2.1 INTRODUCTION

- ***Definition:*** A hash function is a deterministic function that maps an input element from a larger (possibly infinite) set to an output element in a much smaller set.
- The input element is mapped to a ***hash value.***
- For example, in a district-level database of residents of that district, an individual's record may be mapped to one of 26 hash buckets.
- Each hash bucket is labelled by a distinct alphabet corresponding to the first alphabet of a person's name.
- Given a person's name (the input), the output or hash value is simply the first letter of that name (Fig. 7.1).
- Hashes are often used to speed up insertion, deletion, and querying of databases.

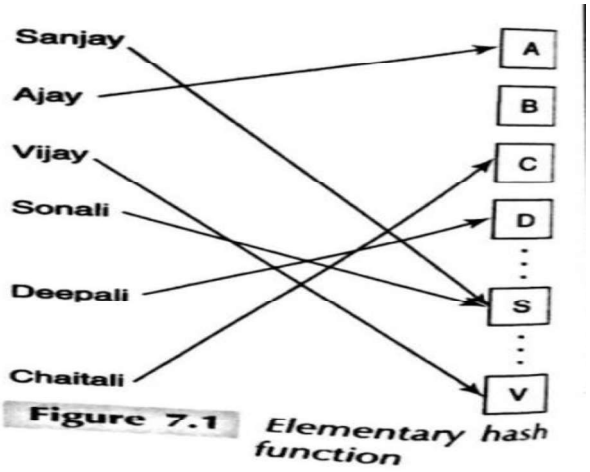


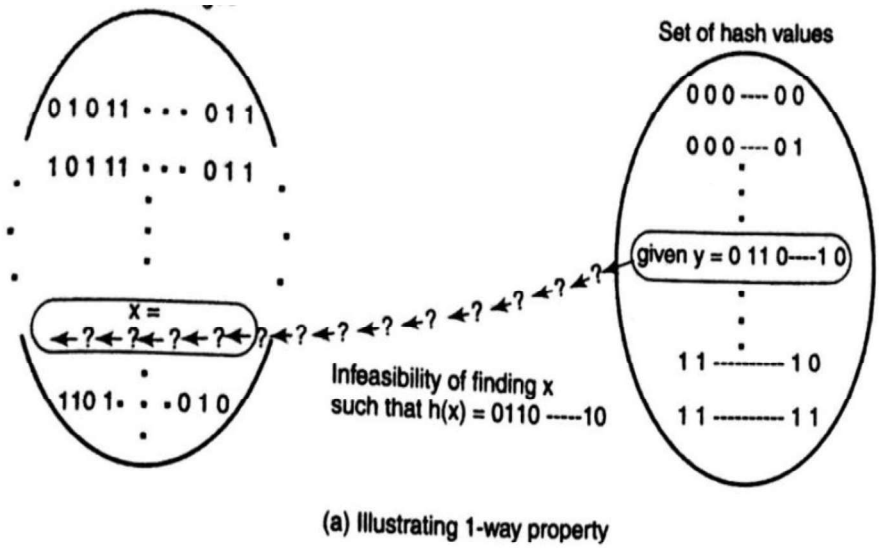
Figure 7.1 Elementary hash function

- In the example above, two names beginning with the same alphabet map to the same hash bucket and result in a collision.

2.2 PROPERTIES

7.2.1 Basics

- A cryptographic hash function, ***h(x),*** maps a binary string of arbitrary length to a fixed length binary string.
- The properties of *h* are as follows:
    1. **One-way property.** Given a hash value, ***y*** (belonging to the range of the hash function), it is computationally infeasible to find an input x such that ***b(x)* = y**
    2. ***Weak collision resistance.*** Given an input value x1, it is computationally infeasible to find another input value x2 such that ***h(x1) = h(x2)***
    3. **Strong collision *resistance.*** It is computationally infeasible to find two input values x1 and no x2 such that h(x1)=h(x2)
    4. ***Confusion + diffusion.*** If a single bit in the input string is flipped, then each bit of the hash value is flipped with probability roughly equal to 0.5.



(a) Illustrating 1-way property

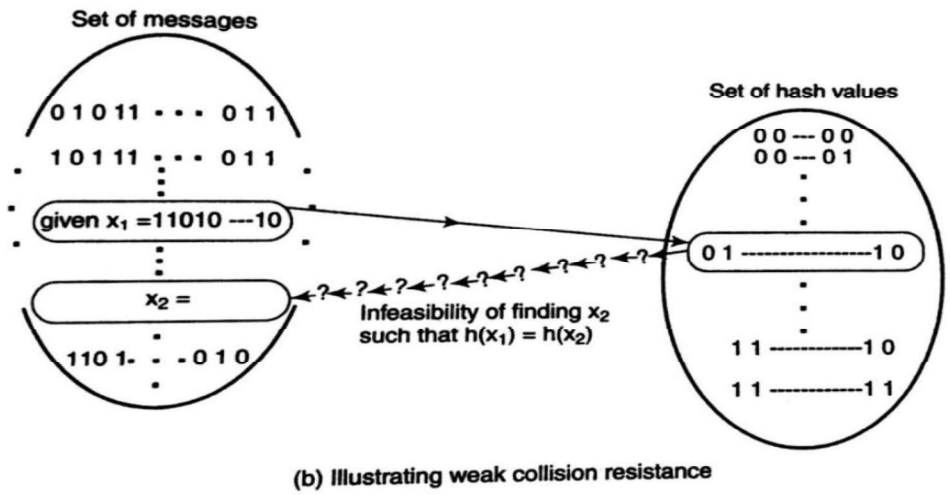


Figure 7.2 Properties of the cryptographic hash

- There is a subtle difference between the two collision resistance properties.
- In the first, the hash designer chooses x1 and challenges anyone to find an x2, which maps to the same hash value as of x1. This is a more specific challenge compared to the one in which the attacker tries to find and x2 such that h(x1)= h(x2).
- In the second challenge, the attacker has the liberty to choose x1.

2.2.2 Attack Complexity

Weak Collision Resistance

- How low long would it take to find an input, x, that hashes to a given value y?
- Assume that the hash value is w bits long. So, the total number of possible hash values is 2^w
- brute force attempt to obtain x would be to loop through the following operations

```
do
{
    generate a random string, x'
    compute h(x')
}
while (h(x') != y)
return (x')
```

- assuming that any given string is equally likely to map to any one of the 2^W hash values, it follows that the above loop would have to run, on the average, 2^{w-1} times before finding an x' such that h(x') = y.

- A similar loop could be used to find a string, x2, that has the same hash value as a given string x1.

Strong Collision Resistance

- A Brute-force attack on strong collision-resistance of a hash function involves looping through the program in Fig. 7.4.
- Unlike the program that attacks weak collision resistance, this program terminates when the hash of a newly chosen random string collides with any of the previously computed hash values.

```
// S is the set of (input string, hash value) pairs
// encountered so far

notFound = true
while ( notFound )
{
    generate a random string, x'
    search for a pair ( x, y) in S where x = x'
     if ( no such pair exists in S )
    {
        compute y' = h(x')
        search for a pair (x, y) in S where y = y'
        if ( no such pair exists in S )
            insert (x', y') into S
         else
             notFound = false
    }
}
return  ( x and x' )  // these are two strings that have
                      // the same hash value
```

Figure 7.4:program to attack strong collision resistance.

THE BIRTHDAY ANALOGY

- Attacking strong collision resistance is analogous to answering the following:
- "What is the minimum number of persons required so that the probability of two or more in the, group having the same birthday is greater than 1/2 ?"
- It is known that in a class of only 23 random individuals, there is a greater than 50% chance that: the birthdays of at least two persons coincide (a "Birthday Collision").
- This statement is referred, to as the Birthday Paradox.

THE BIRTHDAY ATTACK

- The following idea, first proposed by Yuval illustrates the danger in choosing hash lengths less than 128 bits.
- A malicious individual, Malloc, wishes to forge the signature of his victim, Alka, on a fake document, F.
- F could, for example, assert that Alka owes Malloc several million rupees.
- Malloc does the following:

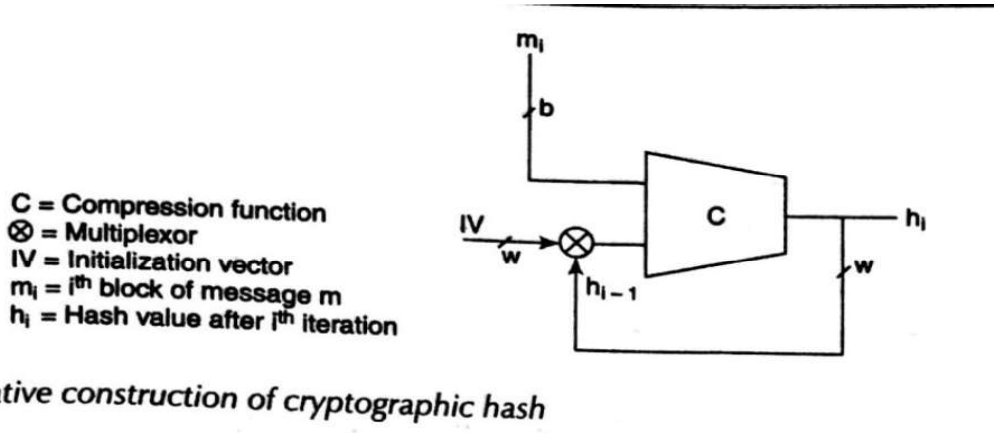
1. He creates millions of documents, Fl, F2,……Fm, etc. that are, for all practical purposes, "clones" of F.
2. This is accomplished by leaving an extra space between two words, etc.
3. If there are 300 words in F, there are 2300 ways in which extra spaces may be left between words.
4. He computes the hashes, h(F1), h(F2), . . . h(Fm) of each of these documents.
5. He creates an innocuous document, D — one that most people would not hesitate to sign. (For example, it could espouse an environmental cause relating to conservation of forests.)
6. He creates millions of "clones" of D in the same way he cloned F above.
7. Let D1, D2, ... be the cloned documents of D.
8. He computes the hashes, h(D1), h(D2), . . . h(Dm) of each of the cloned documents.
9. Malloc asks Alka to sign the document D, and Alka obliges.
10. Later Malloc accuses Alka of signing the fraudulent document
11. the digital signature is obtained by encrypting the hash value of the document using the private key of the signer.
12. Thus, Alka's signature on Dj, is the same as that on Fi,.
13. Hence, at a later point in time, Malloc can use Alka's signature on Dj), to claim that she signed the fraudulent document, F.,.

2.3 CONSTRUCTION

2.3.1 Generic Cryptographic Hash

- The input to a cryptographic hash function is often a message or document.
- To accommodate inputs of arbitrary length, most hash functions (including the commonly used MD-5 and SHA-1) use iterative construction as shown in Fig. 7.5.
- **C is a compression box.**
- It accepts two binary strings of lengths **b and w** and produces an output string of **length w.**
- Here, **b is the block size and w is the width of the digest**.
- During the first iteration, it accepts a pre-defined initialization vector (IV), while the top input is the first block of the message.
- In subsequent iterations, the ***"partial hash output" is fed back*** as the second input to the C-box.
- The top input is derived from successive blocks of the message.
- This is repeated until all the blocks of the message have been processed.
- The above operation is summarized below:
- **$h_1 = C (IV, m_1)$** for first block of message
- **$h_i = C (h_{i-1}.m_i)$** for all subsequent blocks of the message

\



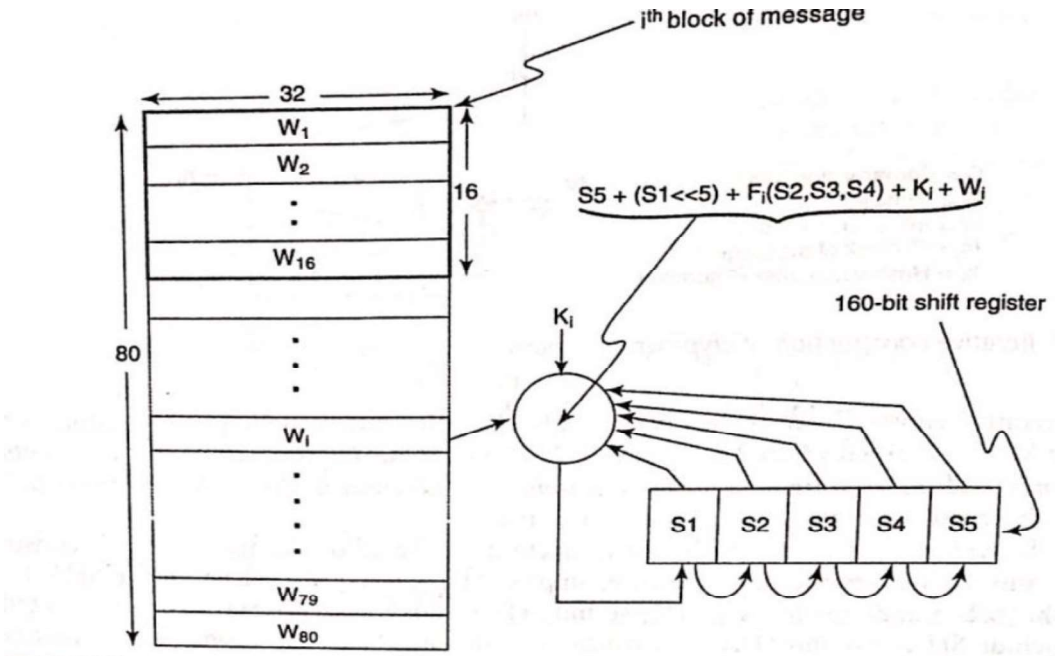
7.5 *Iterative construction of cryptographic hash*

Figure 7.5 Iterative construction of cryptographic hash

- The above iterative construction of the cryptographic hash function is a simplified version of that proposed by **Merkle and Damgard.**
- It has the property that if the compression function is collision-resultant, then the resulting hash function is also collision-resultant.
- MD-5 and SHA-1 are the best known examples. MD-5 is a 128-bit hash, while SHA-1 is a 160-bit hash.

2.3.2 Case Study: SHA-1

- SHA-1 uses the iterative hash construction of Fig. 7.5.



gure 7.6 *Computation of SHA-1*

```
initialize the shift register, S1 S2 S3 S4 S5
for each block of the (message + pad + length field) {
        create the 80-word array [using Eq. (7.2)]
        for i = 1 to 80 {
                temp ← S5 + (S1 << 5) + Fi(S2, S3, S4) + Ki + Wi
                S5 ← S4
                S4 ← S3
                S3 ← S2 >> 2
                S2 ← S1
                S1 ← temp
        }
}
```

F_i is defined b

$$F_i(S2, S3, S4) = (S2 \wedge S3) \vee (\sim S2 \wedge S4), \quad 1 \le i \le 20$$
$$F_i(S2, S3, S4) = S2 \oplus S3 \oplus S4, \quad 21 \le i \le 40$$
$$F_i(S2, S3, S4) = (S2 \wedge S3) \vee (S2 \wedge S4) \vee (S3 \wedge S4), \quad 41 \le i \le 60$$
$$F_i(S2, S3, S4) = S2 \oplus S3 \oplus S4 \quad 61 \le i \le 80$$

- The message is split into blocks of ***size 512 bits***.
- The length of the message, expressed in binary as a 64 bit number, is appended to the message.
- Between the end of the message and the length field, a pad is inserted so that the length of the **(message + pad + 64)** is a ***multiple of 512,*** the block size.
- The pad has the form: 1 followed by the required number of 0's.

Array Initialization

- Each block is split into 16 words, each 32 bits wide.
- These **16 words** populate the first 16 positions, W1, W2 ……W16, of an array of **80 words.**

- The remaining **64 words** are obtained from :

$$W_i = W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16} \quad 16 < i \leq 80$$

- This array of words is shown in Fig. 7.6.

Hash Computation in SHA 1

- A 160-bit shift register is used to compute the intermediate hash values (Fig. 7.6).
- It is initialized to a fixed pre-determined value at the start of the hash computation.
- We use the notation S1, S2, S3, S4, and S5 to denote the five 32-bit words making up the shift register.
- The bits of the shift register are then mangled together with each of the words of the array in turn.
- The mangling is achieved using a combination of the following Boolean operations: **+, v, ~, ^, XOR ROTATE.**

2.4 APPLICATIONS AND PERFORMANCE

2.4.1 Hash-based MAC

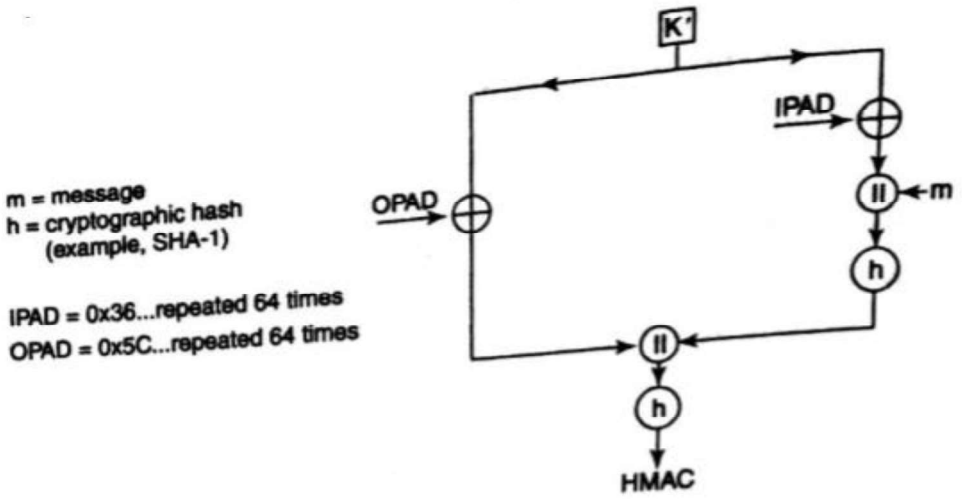
MAC

- MAC is used as a message integrity check as well as to provide message authentication.
- It makes use of a common shared secret, k, between two communicating parties.
- The hash-based MAC that we now introduce is an alternative to the CBC-MAC.
- The cryptographic hash applied on a message creates a digest or digital fingerprint of that message.
- Suppose that a sender and receiver share a secret, k.
- If the message and secret are concatenated and a hash taken on this string, then the hash value becomes a fingerprint of the combination of the message, m and the secret, k.
- ***MAC = h (m|| k)***
- The MAC is much more than just a ***checksum*** on a message.
- It is computed by the sender, appended to the message, and sent across to the receiver.

- On receipt of the **message + MAC,** the receiver performs the computation using the common secret and the received message.
- It checks to see whether the MAC computed by it matches the received MAC.
- A change of even a single bit in the message or MAC will result in a mismatch between the computed MAC and the received MAC.
- In the event of a match, the receiver concludes the following:
- ***(a) The sender of the message is the same entity it shares the secret with — thus the MAC provides source authentication.***
- ***(b) The message has not been corrupted or tampered with in transit — thus the MAC provides verification of message integrity***.
- ***Drawbacks:***
- An attacker might obtain one or more message—MAC pairs in an attempt to determine the MAC secret.
- First, if the hash function is one-way, then it is not feasible for an attacker to deduce the input to the hash function that generated the MAC and thus recover the secret.
- If the hash function is collision-resistant, then it is virtually impossible for an attacker to suitably modify a message so that the modified message and the original both map to the same MAC value.

HMAC

- There are other ways of computing the hash MAC other than this method using HMAC .
- Another possibility is to use key itself as the Initialization Vector (IV) instead of concatenating it with the message.
- Bellare, Canetti, and Krawczyk proposed the HMAC and showed that their scheme is re against a number of subtle attacks on the simple hash-based MAC.
- Figure 7.7 shows how an HMAC is computed given a key and a message.



7.7 Computation of an HMAC

- The ***key is padded with O's*** (if necessary) to form a ***64-byte string*** denoted ***K'*** and ***XORed with a constant*** (denoted IPAD).
- It is then concatenated with the message and a hash is performed on the result.
- ***K' is also XORed*** with **another constant (denoted OPAD)** after which it is prepended to the output of the first hash.
- Once again hash is then computed to yield the HMAC.
- As shown in Fig. 7.7, HMAC performs an extra hash computation but provides greatly enhanced security.

2.4.2 Digital Signatures

- The same secret that is used to generate a MAC on a message is the one that is used to verify the MAC.
- Thus the MAC secret should be known by both parties - the party that generates the MAC and the party that verifies it.
- A digital signature, on the other hand, uses a secret that only the signer is privy to.
- An example of such a secret is the signer's private key.
- A crude example of an RSA signature by A on message, m, is $\mathbf{E_{A.pr}(m)}$
- where A.pr is A's private key.
- The use of the signer's private key is a fundamental aspect of signature generation.
- Hence, a message sent together with the sender's signature guarantees not just integrity and authentication but also non-repudiation, i.e., the signer of a document

cannot later deny having signed it since she alone has knowledge or access to her private key used for signing.

- The verifier needs to perform only a public key operation on the digital signature (using the signer's public key) and a hash on the message.
- The verifier concludes that the signature is authentic if the results of these two operations tally,

$$E_{A.pu}\left(E_{A.pr}(h(m))\right) \stackrel{?}{=} h(m)$$

Question Bank (module 2-chapter 2)

1. Explain generic hash computation and HMAC .
2. Define hashing Explain the properties of hashing with a neat figure.
3. Explain SHA-1 computation with a neat illustration.
4. Explain weak and strong collision resistance.
5. Explain digital signature.
6. Explain birthday analogy and birthday attack

MODULE 2 - [Chapter 3]

DISCRETE LOGARITHM AND ITS APPLICATIONS.

INTRODUCTION.

- Consider the finite, multiplicative group $(Z_p^*, *_p)$ where p is prime.
- Let g be the generator of the group.
- $g^1 \bmod p, g^2 \bmod p, \ldots g^{p-1} \bmod p$.
- Let x be an element in $\{0, 1, \ldots p-1\}$.
- The function:

$$y = g^x \pmod p$$

→ Modular exponentiation with Base g and modulus p.

- The Inverse operation is:

$$x = \log_g y \pmod p$$

→ Discrete logarithm

Example.

→ Let $p = 131$

$g = 2$

* DIFFIE - HELLMAN KEY EXCHANGE PROTOCOL.

→ Consider two parties, A & B that need to agree upon a shared secret for the duration of their current session.

→ In 1976, Diffie and Hellman proposed the idea of a private key and corresponding public key,

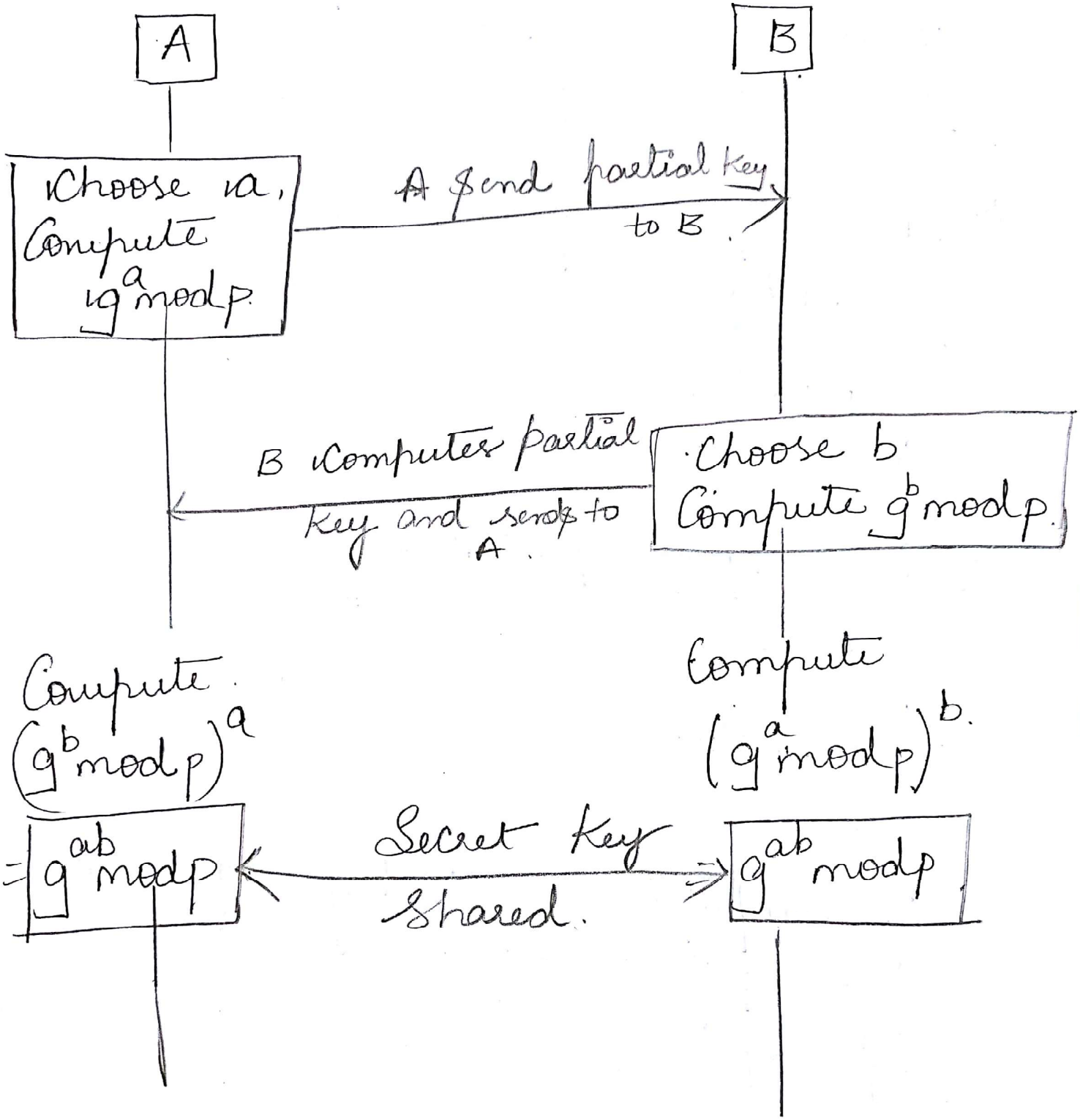
1) A chooses a random integer a, $1 < a < p-1$, computes the partial key $g^a \bmod p$ and sends to B.

2) B chooses a random integer b, $1 < b < p-1$, computes the partial key $g^b \bmod p$ and sends to A.

3) On the receipt of A's msg, B computes $(g^a \bmod p)^b \bmod p = g^{ab} \bmod p$

4) On the receipt of B's msg, A computes $(g^b \bmod p)^b \bmod p = g^{ab} \bmod p$.

DIFFIE HELLMAN KEY EXCHANGE



Nagashree. C
Asst Professor, Department of CSE,SVIT

Example:

Compute Diffie Hellman partial Keys and Secret Keys.

where $a = 24$, $b = 17$,

$g = 2$ and $p = 131$.

1) A computes partial key:

$= g^a \bmod p$.

$= 2^{24} \bmod 131$

$= 46$

2). B Computes partial key:

$= g^b \bmod p$.

$= 2^{17} \bmod 131$

$= 72$

3) A Computes Secret key after receiving B's partial key.

$= (g^b \bmod p)^a$ → B's partial key.

$= (72)^{24} \bmod 131$

$= \boxed{13}$

4) B Computes Secret key: $(g^a \bmod p)^b$

$= 46^{17} \bmod 131$

$= \boxed{13}$.

ATTACKS

- The partial keys, $g^a \bmod p$ and $g^b \bmod p$ are sent in clear
- An Eavesdropper with the knowledge of the partial keys and public parameters (p and g) deduce the common secret $g^{ab} \bmod p$, derived by A & B.
- This problem is referred to as Computational Diffie Hellman problem.

MAN IN THE MIDDLE ATTACK ON DIFFIE-HELLMAN KEY EXCHANGE.

- An attacker, C chooses an integer c and computes $g^c \bmod p$.
- C then interrupts A's message to B, substitutes it with $g^c \bmod p$ and sends this instead to B.
- C also intercepts B's message to A sending $g^c \bmod p$ instead.
- After the message transfer

B computes $\rightarrow (g^c \bmod p)^b \bmod p$

$\rightarrow \boxed{g^{bc} \bmod p.}$

- While A Computes,

$$(g^c \bmod p)^a \bmod p = \boxed{g^{ac} \bmod p.}$$

- C also Computes the two Secrets
  - → $g^{ac} \bmod p$ and
  - → $g^{bc} \bmod p$.
- A and B might think that they have a secure channel for Communication by encrypting all messages.
- But A Shares the Secret $g^{ac} \bmod p$ with C,
- B shares the Secret $g^{bc} \bmod p$ with C.
- Every Subsequent message encrypted by A and intended for B can be decrypted by C.
- Similarly Every message from B to A can be decrypted by C.
- This is a classic Example of an active "Man in the Middle Attack".

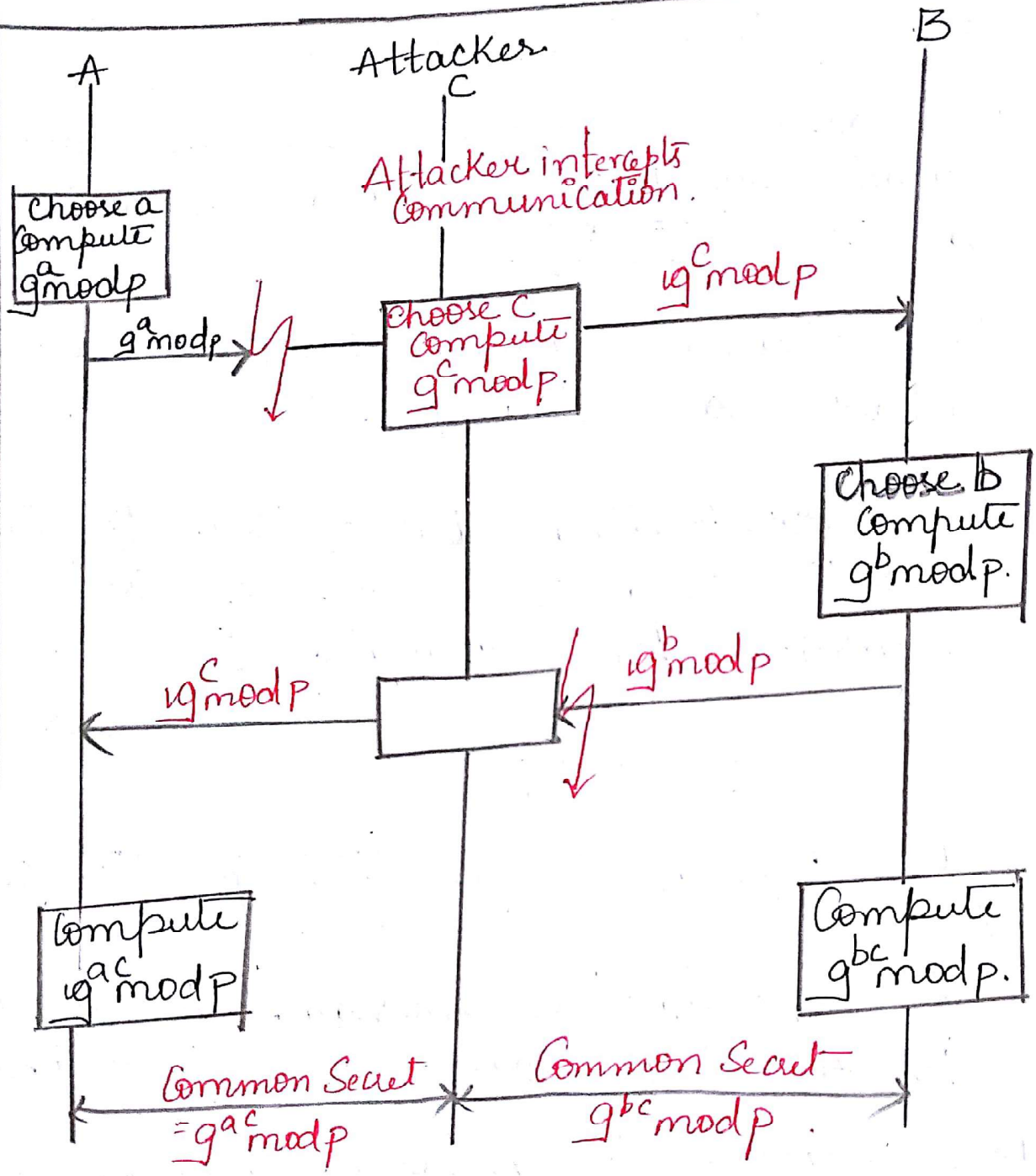


Fig: Man in the Middle Attack on Diffie Hellman Key Exchange.

EL GAMAL ENCRYPTION.

- El gamal encryption uses a large prime number p and generator g in $(Z_p^*, *_p)$.
- An Elgamal private key, is an integer a, $1 < a < p-1$.
- The Corresponding public Key is the triplet (p, g, α) where α is the encryption key calculated as:

$$\alpha = g^a \bmod p.$$

- Let (p, g, α) be the public Key of A.
- To Encrypt a message to be sent to A, B does the following:

1) B chooses a random number r, $1 < r < p-1$ such that r is relatively prime to $p-1$

2) B Computes:

$$C_1 = g^r \bmod p \qquad C_2 = (m * \alpha^r) \bmod p$$

3) B sends the Ciphertext
$C = [C_1, C_2]$ to A.

Decryption At A's side

* A uses its private key a to decrypt and obtain plaintext m.:

$$(C_1^{-a}) * C_2 \bmod p$$

* ELGAMAL SIGNATURES.

→ Let a be the private key of A.
→ Let (p, g, x) be the public key of A.
→ To sign a message m, A does the following:

1) She computes the hash $h(m)$ of the message.

2) She chooses a random number r, $1 < r < p-1$, such that r is relatively prime to $p-1$.

3) She Computes

$$x = g^{r} \bmod p$$

4) She Computes

$$y = (h(m) - ax) r^{-1} \bmod (p-1)$$

5) The Signature is the pair (x, y).

* Signature verification user x,

To prove Elgamal Signature:

Consider step 4 Eqn

$$y = (h(m) - ax) r^{-1} \bmod p-1$$

$$y = (h(m) - ax) \frac{1}{r} \bmod p-1$$

$$ry = (h(m) - ax) + K(p-1)$$ where K is an integer

→ Raising Both Sides to power of g and reducing modulo P.

$$g^{ry} = g^{h(m) - ax}$$

$$g^{ry} = g^{h(m)} \frac{1}{g^{ax}} \bmod p.$$

g^{p-1} is Equal to 1 [Fermat's Theorem]

$$g^{ax} \cdot g^{ry} = g^{h(m)} \bmod p.$$

$$\alpha^x * x^y = g^{h(m)} \bmod p.$$ [since $\alpha = g^a \bmod p$. $x = g^r \bmod p$]

* SCHNORR SIGNATURE

→ Schnorr signature is the pair (x, y) where

$$x = h(m \,||\, g^r \bmod p) \text{ and}$$

$$y = (r + ax) \bmod q.$$

where $\alpha = g^a \bmod p$.

r be random number

$$1 \leq r \leq q-1.$$

PROBLEMS ON ELGAMAL ENCRYPTION.

Q. A Block of plaintext message $m = 3$, that to be encrypted,
Assume $P = 11$, $g = 2$, recipients private Key $= 5$,
Sender chooses random integer $r = 7$.
Perform Encryption & Decryption.

Step 1: $p = 11$, $g = 2$
Recipients private key, $a = 5$.
Compute public key of receiver:

$$K = g^a \bmod p.$$

$$K = 2^5 \bmod 11$$

$$K = 32 \bmod 11$$

$$\boxed{K = 10}$$

step 2: Compute C_1 and C_2 [Sender has to compute]

$$C_1 = g^r \bmod p$$

$C_1 = g^r \bmod p \qquad [r = 7]$

$= 2^7 \bmod 11$

$= 128 \bmod 11$

$\boxed{C_1 = 7}$

$C_2 = m * \alpha^r \bmod p \qquad [m = 3]$

$= 3 * 10^7 \bmod 11$

$\boxed{C_2 = 8}$

$C = [7, 8]$

Step 3: Decrypt

$m = C_1^{-a} * C_2 \bmod p$

$= 7^{-5} * 8 \bmod 11$

$= (7^{-1})^5 * 8 \bmod 11$

$= 8^5 * 8 \bmod 11$

$\boxed{m = 3}$

$7 \times 3 = 21 \bmod 11$ ✗

$7 \times 5 = 35 \bmod 11$ ✗

not Equal to 1 hence Continue.

Substitute $7^{-1} = 8$ (Inverse.)

$\therefore 7 \times 8 = 56$

Take $56 \bmod 11$

$= 1$

[Equivalent to 1]

Q. $p = 23,\ g = 11,\ a = 6,\ r = 3,\ m = 10.$

step1: $K = g^a \bmod p$

$= 11^6 \bmod p$

$\boxed{K = 9}$

step2: Compute C_1, C_2

$C_1 = g^r \bmod p$

$= 11^3 \bmod 23$

$\boxed{C_1 = 20}$

$C_2 = (m * K^r \bmod p)$

$= (10 * 9^3 \bmod 23)$

$\boxed{C_2 = 22}$

step3: Decrypt:

$m = C_1^{-a} * C_2 \bmod p.$

$= 20^{-6} * 22 \bmod 23$

$= (20^{-1})^6 * 22 \bmod 23.$

$= (15)^6 * 22 \bmod 23$

[Not Equal to 1]

$20 * 1 = 20 \bmod 23$ ✗

$20 * 2 = 40 \bmod 23$ ✗

⋮

$20 * 15 = 300 \bmod 23$

$= 1$ ✓

$\boxed{m = 10}$

MODULE-2

Public Key Cryptography and RSA

RSA

Step1: Choose two large prime numbers p and q

Step2: Compute the modulus n,

$$n = p * q$$

Step3: Compute Euler totient function

$$\phi(n) = (p-1) * (q-1)$$

Step4: Choose the encryption key e such that

$$\gcd(e, \phi(n)) = 1$$

Step5: Compute decryption key d

$$de \bmod \phi(n) = 1$$

$$d = e^{-1} \bmod \phi(n)$$

e is called public key

d is called private key.

Nagashree. C
Asst Professor, Department of CSE,SVIT

Step 6: Encryption:

$$C_i = m_i^e \bmod n$$

Step 7: Decryption:

$$m_i = C_i^d \bmod n$$

Example:

Suppose RSA prime numbers are $p=3$, $q=11$, $e=3$., $m=00111011$

Solution:

Step 1: Compute modulus n

$$n = p*q$$
$$n = 3*11$$
$$n = 33$$

Step 2: Compute $\phi(n)$

$$\phi(n) = (p-1)*(q-1)$$
$$= (3-1)*(11-1)$$
$$= 2*10$$
$$\phi(n) = 20$$

Nagashree. C
Asst Professor, Department of CSE,SVIT

> Step 3: Compute encryption key e.

$\gcd(e, \phi(n)) = 1$

$\gcd(3, 20) = 1$

$e = 3$ = public key.

> Step 4: Compute Decryption key

$d = e^{-1} \bmod \phi(n)$

$= 3^{-1} \bmod 20$

$\boxed{d = 7}$

> Step 5: Encryption: $\boxed{C_i = M_i^e \bmod n}$

Step 6: Decryption $\boxed{M_i = C_i^d \bmod n}$

M = 0 0 1 1 1 0 | 1 1

Block 1 ↓

Block 2

0 0 0 1 1 1 — append zeros

NOTE: plain text M is divided into Block size of 6 bits or number of bits required to represent M = 33 requires 6 bits in Binary

Nagashree. C
Asst Professor, Department of CSE, SVIT

Encryption

$C_i = m_i^e \bmod n$

$m_1 = 001110,$

$m_1 = 14.$

$C_1 = 14^3 \bmod 33$

$\boxed{C_1 = 5}$

$C_2 = m_2^e \bmod n$

$m = 000011.$

$m_2 = 3$

$C_2 = m_2^e \bmod n$

$= 3^3 \bmod n$

$\boxed{C_2 = 27}$

Decryption.

$m_i = C_i^d \bmod n$

$d = 7$

Replace C Value computed

$m_1 = 5^7 \bmod 33.$

$\boxed{m_1 = 14}$

$m_2 = C_2^d \bmod n.$

C_2 Computed is 27 substitute C, d & n value

$m_2 = C_2^d \bmod n$

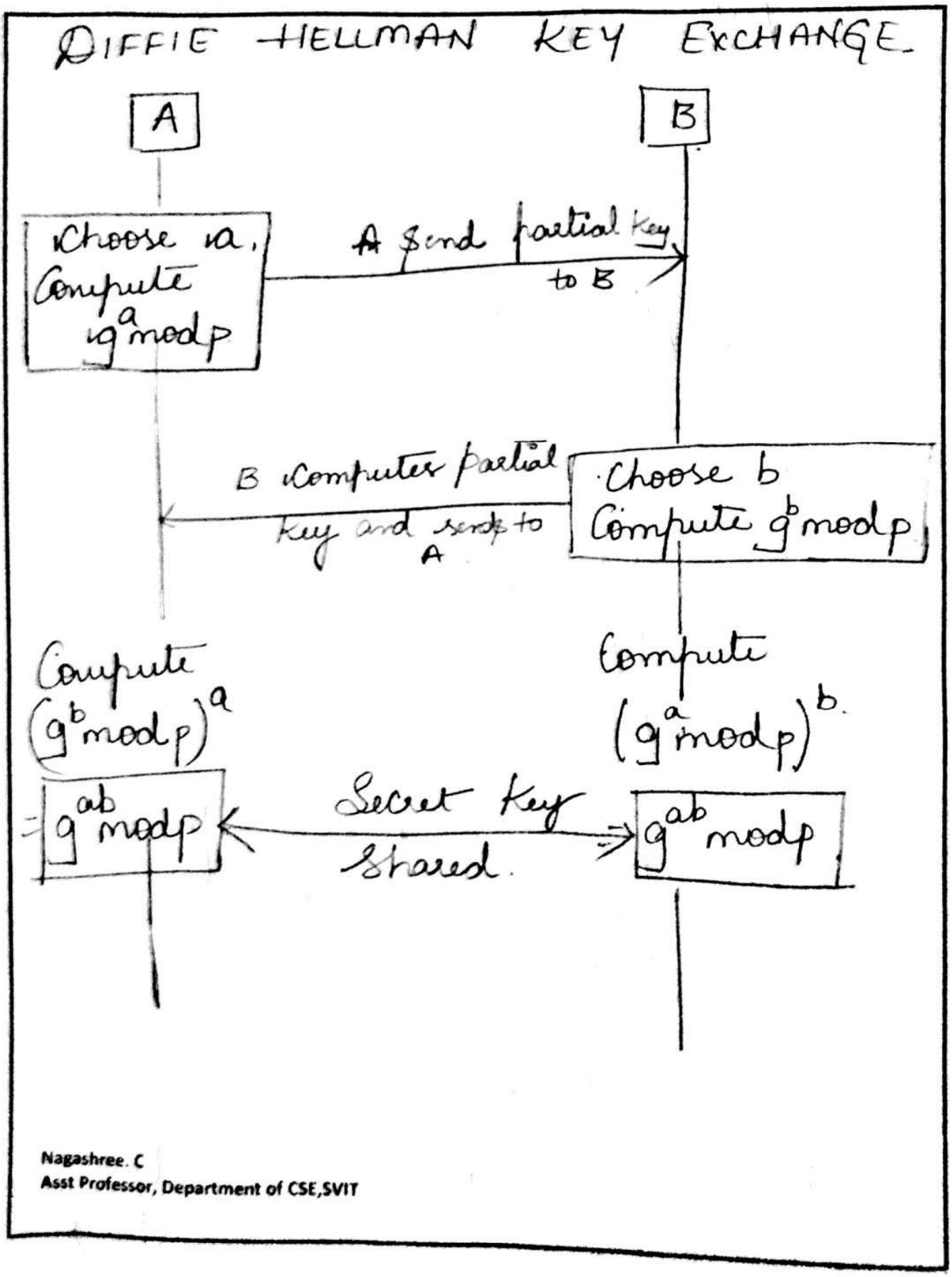
$= 27^7 \bmod n.$

$= (27^5 \bmod 33 \times 27^2 \bmod 33) \bmod 33.$

$\boxed{m_2 = 3.}$

Nagashree. C
Asst Professor, Department of CSE,SVIT

DIFFIE HELLMAN KEY EXCHANGE



Nagashree. C
Asst Professor, Department of CSE,SVIT

Example:

Compute Diffie Hellman partial keys and Secret keys.

where $a = 24$, $b = 17$,

$g = 2$ and $p = 131$.

1) A computes partial key:

$= g^a \bmod p$

$= 2^{24} \bmod 131$

$= 46$

2) B Computes partial key:

$= g^b \bmod p$

$= 2^{17} \bmod 131$

$= 72$

3) A computes Secret key after receiving B's partial key.

$= (g^b \bmod p)^a$ ← B's partial key.

$= (72)^{24} \bmod 131$

$= \boxed{13}$

4) B computes Secret key: $(g^a \bmod p)^b$

$= 46^{17} \bmod 131$

$= \boxed{13}$.

MODULE 3

Chapters	
1.Key Management	2.Authentication-I
3.Authentication-II	4.IPSec-security at the network layer
5.Security at transport layer	

CHAPTER 1 Key Management

3.1 INTRODUCTION

- Key management is related to the ***generation, storage, distribution, and backup of keys.***
- The focus is on the management of ***public key—private key pairs.***
- The public key—private key pairs are used for ***encryption/decryption, signature generation/ verification, and for authentication.***
- To encrypt a session key for use in communication between A and B, A needs to know B's public key.
- The key issue hereis **"How does A know B's public key?"**
- **Possibility 1:**
  - ✓ **A** may frequently communicate with **B** in a secure manner, so she may already have **B's public key.**
  - ✓ First, ***B must have securely communicated his public key to A*** at some point in the past. A actually receives B's public key and not a public key from someone posing as B.
  - ✓ If at ***any time B's private key is compromised***, the confidentiality of messages from A to B using the corresponding public key can no longer be guaranteed.
  - ✓ An individual, with the compromised private key, can decrypt messages encrypted with the old public key.
- **Possibility 2:**
  - ✓ Every entity's public key is securely maintained in a **centralized directory.**
  - ✓ Suppose A wishes to securely communicate with an e-commerce website, B-Mart.
  - ✓ All she has to do to obtain B-Mart's public key is to query the directory for it.
  - ✓ The question here is "***Who would take the responsibility for maintaining such a directory?"***
  - ✓ There are huge scalability problems associated with such a directory,spoofing and denial of service attacks, the non-uniqueness of names.
- **Possibility 3**:
  - ✓ A receives a document signed by a trusted source, C, containing B's public key.

3.2 DIGITAL CERTIFICATES

3.2.1 Certificate Types

- A digital certificate is a signed document used to ***bind a public key to the identity of a person.***
- Example such as An individual's identity could be his/her name, national identification number, e-mail .or postal address, employer, etc. or some combination of these.
- **CA:**The entity that issues certificates is **a trusted entity called a certification Authority (CA)certificate authority.**
- Certificates may be issued to individuals, to organizations, or even to servers.
- The most basic type of certificate may be applied for through regular e-mail with the applicant stating his/her public key, name, e-mail address, etc.
- In this case, the CA requires no credentials from the applicant.
- It simply assumes that the applicant is in possession of the (uncompromised) private key corresponding to the Public key contained in the application received via e-mail.
- The verifier of such a certificate should realize that the above certificates are "**Trust at your own risk certificates."**
- To carry more weight, certificate issuance would require the CA to perform identity verification of the applicant.
- The CA may have to obtain and verify several details of the applicant this task would be delegated by the ***CA to the registration Authority (RA)***

3.2.2 X.509 Digital Certificate Format

- X.509 is an ITU standard specifying the format for **public key certificates.**
- The fields of an X.509 certificate together with their meaning are as follows:
    1. **Certificate Serial Number and Version :**Each certificate issued by a given CA will have a unique number.
    2. **Issuer information:** The distinguished name of an entity includes his/her/its "common name," e-mail address, organization, country, etc.
    3. **Certificate signature and associated signing algorithm information:** It is necessary to verify the authenticity of the certificate. For this purpose, it is signed by the issuer. So, the certificate should include the issuer's digital signature and also the algorithm used for signing the certificate.
    4. **Validity period:** There are two date fields that specify the ***start date and end date*** between which the certificate is valid.
    5. **Subject information :**This includes the distinguished name of the certificate's subject or owner.
        - For example, if a customer intends to communicate with an e-commerce web server at www.B-Mart.com, then the customer's browser will request B-Mart's certificate.

- Client-side software will check whether the "Common Name" in B-Mart's certificate tallies with B-Mart's domain name.
- Other information, such as the subject's country, state, and organization, may be included.

6. **Subject's public key information:** The public key, the public key algorithm (e.g., RSA or DSA), and the public key parameters (modulus in the case of RSA and modulus + generator in case of Diffie-Hellman).

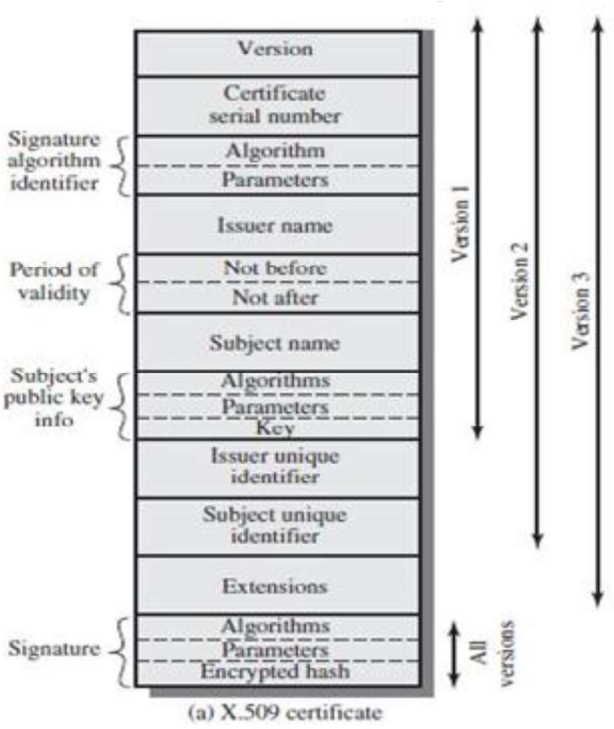


Figure 3.1 A digital certificate

3.2.3 Digital Certificates in Action

- Assume that A needs to securely ***transmit a session key*** to B.
- So, she encrypts it with B's public key.
- A will need to retrieve the public key from B's certificate.
- A may already have B's certificate or she may send a message to B requesting it.
- There are a number of checks that A will have to perform on B's certificate prior to using B's public key.
  - ✓ Is this indeed B's certificate?
  - ✓ This can be determined by checking whether the certificate contains B's name. But the "common name" field alone may be inadequate (since there are probably many John Browns, for example).

- ✓ It may be necessary to check other fields in the certificate such as the subject's web page URL or e-mail address.
- ✓ A should check if the certificate is still valid. Since the validity period is contained in the certificate, this is easily done.
- ✓ Finally, the certificate must be signed by a CA or RA.
- ✓ A should verify the signature contained in the certificate.
- ✓ A requires the CA's public key for signature verification.
- ✓ The CA may be globally known or may be known to the community that A and B belong.
- ✓ In this case A has access to the CA's public key.

3.3 PUBLIC KEY INFRASTRUCTURE

3.3.1 FUNCTIONS OF A PKI

- A public key infrastructure includes the **CA's ,the physical infrastructure(**encryption technologies, hardwareetc.), and the formulation and enforcement of policies/procedure.
- It includes the following services:
  - ✓ **Certificate creation,issuance,storage and archival**
  - ✓ **Key generation and key escrow**
  - ✓ **Certificate/key updation**
  - ✓ **Certificate revocation**
- There are crucial differences in the support required for private keys used for decryption versus those used for signing.
- In the case of encryption/decryption, it is often necessary to have ***a back-up of the decryption key.***
- If not, an employee who looses his decryption key will be unable to decrypt the archives of sensitive data he may have accumulated.
- For this reason, the PKI within an organization, for example, might hold the private keys in **escrow, i.e., they may be securely backed up and made available to the owner or to a trusted authority** (such as a law enforcement agency) under special circumstances.
- On the other hand, there ***is no need to back up a private key used for digital signing.***
- If such a key is lost, the owner could inform the CA or PKI administrator (within an organization).
- He/she could then obtain a new signing key and receive a new certificate carrying the corresponding public key.
- An important function of the PKI is to provide a safe archival facility for all issued certificates.

3.3.2 PKI Architectures

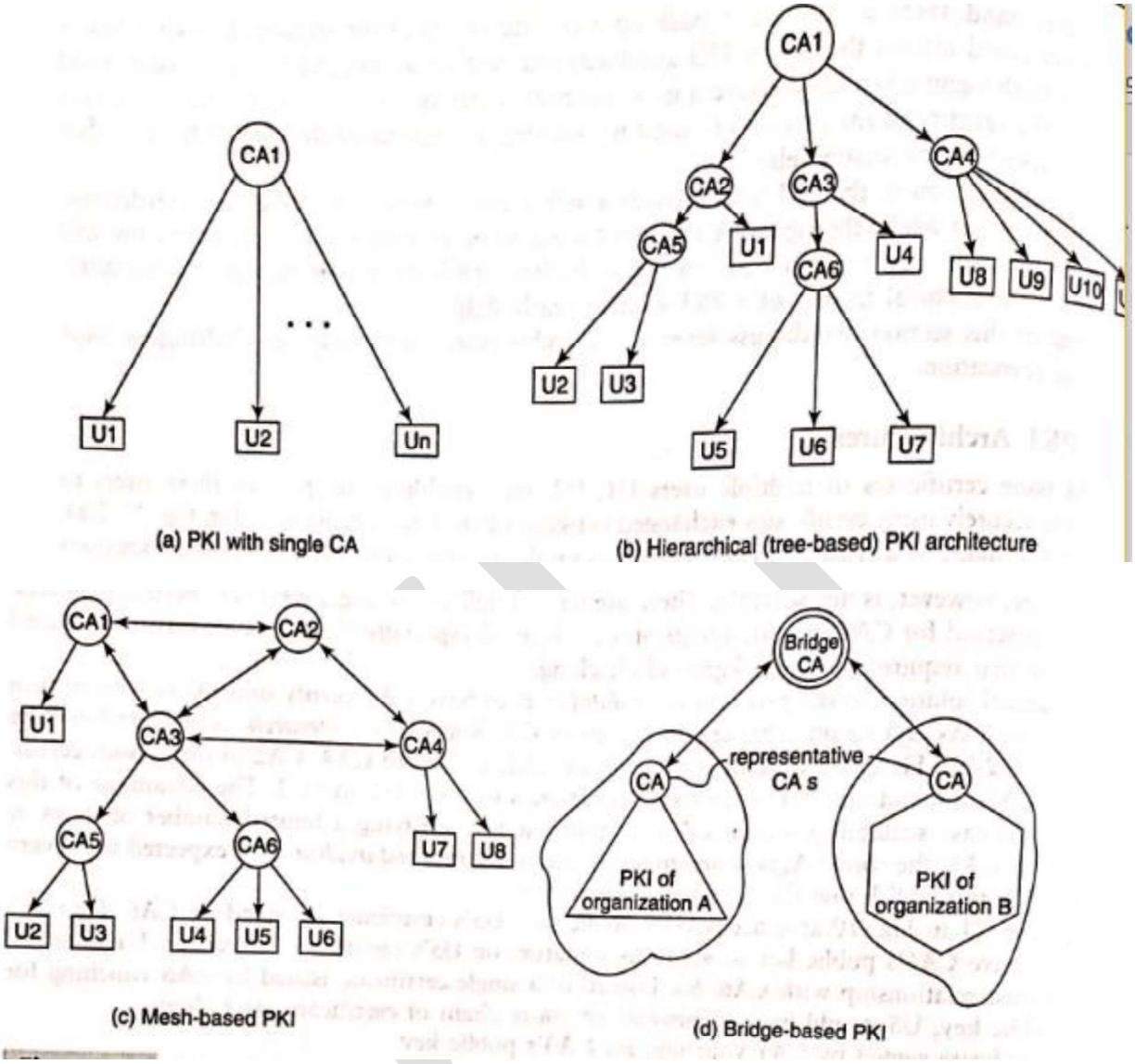


Figure: PKI architecture

1. **PKI with single CA:**

- CA1 could issue certificates to multiple users Ul, U2, etc., enabling any pair of these users to communicate securely using certificates exchanged between them.
- This is represented in above Fig.(a).
- Each arc in the figure is a trust relationship.
- For example, the arc from the CA1 to U2 expresses the fact that CA1 vouches for U2's public key in the certificate issued by the CA1 to U2. Such an architecture, however, is not scalable.
- There are tens of millions of users who may need certificates. It is not practical for CA1 to issue certificates to all.

2. Hierarchial (tree-based PKI architecture)

- Apractical solution to the problem of scalability is to have CA1 certify other CAs who in turn certify other CAs and so on.
- This creates a tree of CAs known as a **hierarchical PKI architecture** [see above Fig.(b)].
- Here, CA1 issues certificates to CA2, CA3, and CA4.
- CA2 in turn issues certificates to CA5 and end user Ul.
- CA5 issues certificates to users U2 and U3.
- The advantage of this approach is easy scalability — each CA is responsible for certifying a limited number of users or other CAs.
- CA1, the root CA, is sometimes referred to as the trust anchor.
- every node in the tree will know the root CA's public key.
- Suppose U1 in Fig.(b) needs U5's public key.
- U5 would have to provide an entire chain of certificates as follows:
- **(1) Certificate signed by CA1 vouching for CA3's public key**
- **(2) Certificate signed by CA3 vouching for CA6's public key**
- **(3) Certificate signed by CA6 vouching for U5's public key**
- It is assumed that each node has a copy of the root's public key.
- So, upon receiving the above certificate chain, U1 can verify the signature on the first certificate using CA1's (the trust anchor'! public key.

3. Mesh based PKI

- A more dense web of trust is shown in Fig. (c) and is referred to as a **mesh-based PKI.** This could include mutually trusting CAs — CA1 trusting CA2 and. CA2 trusting CA 1shown by a bidirectional arc between CA1 and CA2.
- In tree based PKI , there may be multiple trust paths between two users.
- Example there could be multiple trust paths between user 1 and user 7

  Path 1:CA1,CA3, and CA 4

  Path 2: CA1,CA2, and CA 4.

4. Bridge based PKI

- Another PKI architecture, referred to as ***bridge-based PKI***, is motivated by the need for secure communications between organizations in a business partnership.
- Suppose that the partnering organizations already have their own PKIs.
- **A bridge CA is introduced that establishes a trust relationship** with a **representative CA from each organization**.
- This is accomplished by the bridge CA and the organizational representatives issuing certificates to each other.
- The representative CA is one that has a trust path to all (or at least most) of the users in that organization.

- Figure 10.2(d) shows a bridge CA that extends the web of trust between two existing organizational PKIs.

3 .3.3 Certificate revocation

Revocation Scenarios

- The validity period of an X.509 certificate is always contained in the certificate.
- However, there are other reasons why a seemingly valid certificate may actually be invalid.

Scenario 1: The certificates subject, Prashant, was issued a certificate valid between Jan 01, 2010, and Dec 31, 2010.however he quit the organization on April 1, 2010.

- Assume that Prashant's certificate is used for key exchange/authentication and that **he has made a copy of it.**
- The session key itself is then used to encrypt all messages in both directions for the duration of the ensuing session.
- Generally speaking, it is not legal for Prashant to act on behalf of his company beyond the date of his resignation. However, that is precisely what he could do when he attempts to establish official business communication with a customer of his company on say June 10, 2010.
- Based on the expiration date in Prashant's certificate, the customer would deduce that the certificate was valid.
- Moreover, Prashant would be able to authenticate himself or perform unauthorized decryption since he knows the ***private key corresponding to the public key in his certificate.*** Thus, Prashant might continue to do business on behalf of his company even after resigning.
- Based on Scenario 1, we need a mechanism to revoke a certificate issued by an organization to an employee when the he leaves or changes roles.

Scenario 2:

- Consider a single chain in a PKI (Fig. 3.3).
- Suppose that the ***private key of CA3*** were compromised.
- An attacker with access to the ***compromised private key*** could then do the following:
- Generate a ***public key, private key pair (X, Y).***
- Create a certificate containing the public key X with subject name = **U'.**
- Sign the above certificate using the compromised private key of CA3.
- The attacker has thus created a fictitious entity **U'**, masquerading as a legitimate subject, U (see Fig. 3.3).
- Now the attacker can forge the signature of U on any message by signing with the private key, Y.

- The attacker would provide a certificate chain of two certificates — the certificate issued by CA1 vouching for CA3's public key and the above certificate created by him.
- This chain is a valid trust path from the root CA to the subject U.
- Using the public key of CA1 and the certificate chain, the verifier would accept the fraudulent signature generated using Y as an authentic signature of U.
- Scenario 2 is that if a CA's private key is compromised, then any certificate issued by that CA is invalid and it should not be included in any trust path or certificate chain.

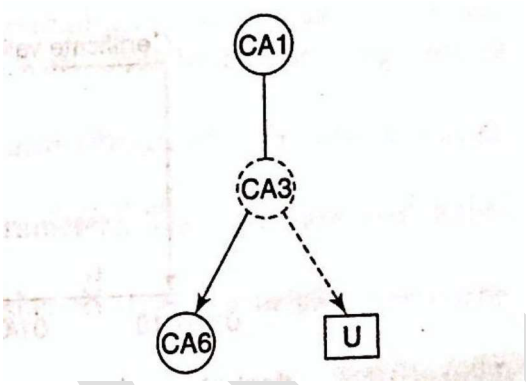


Figure 3.3 Revocation scenario 2

Handling Revocation

Solution 1:

- One possible solution to the problem of certificate revocation is to use an ***on-line facility*** that provides information on the ***current status of digital certificates.***
- For this purpose, a ***protocol called On-line Certificate Status Protocol (OCSP)*** is employed.

Solution 2:

- Another proposed solution is to distribute lists of revoked certificates — ***Certificate Revocation Lists (CRLs).*** The frequency of list updation is an important consideration.
- If CRLs are distributed, toofrequently, they could consume considerable bandwidth.
- On the other hand, if they were distributed infrequently, information on recently revoked certificates may not reach those who need it in a timely fashion.

Solution 3

- Design a system wherein the signer requires the cooperation of a **Trusted Third Party (TTP)** in generating a signature.
- Both, the signer and the TTP have a part of the private key with neither party knowing the other part.
- To sign a document, the signer would contact the TTP.

- Before requesting to sign , the TTP could check whether the signer's certificate has been revoked and participate only if the signer's certificate has not been revoked.
- Indeed, the TTP may itself maintain **certificate revocation information.**
- The TTP may also act as a **timestamp authority** and certify the time at which the document is signed.
- This may be done, for example, by signing a value obtained by **concatenating a timestamp with the hash of the document.**

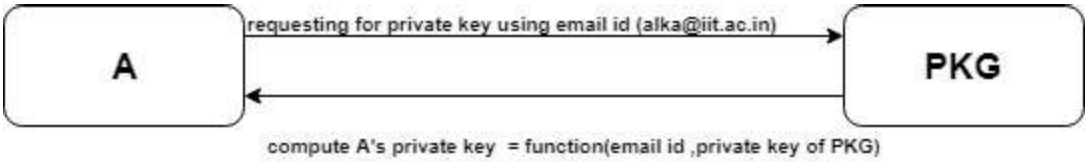
3.4 IDENTITY-BASED ENCRYPTION

3.4.1 Preliminaries

- The digital certificate is a verifiable way of communicating the public key of a entity .
- Certificates are transmitted along with messages for purposes such ***as authentication, signature verification, and encryption.***
- An alternative to digital certificates emerged in 1984 in the form of **Identity-based Encryption (IBE).**
- Shamir's used a scheme wherein a person's public key could be computed as a function of that **person's unique credential such as his/her e-mail address**. Thus, anyone can reliably compute A's public key only knowing A's e-mail address, for example.
- IBE assumes the use of a **TTP called the Private Key Generator (PKG).**

Here is how a generic IBE scheme works:

- The PKG has a **private key and associated public key parameters.**(K_{pr} ,**public key parameters)**
- To obtain a private key, A informs the PKG that she wishes to receive a private key corresponding to her ID, say alka@iitb.ac.in
- The PKG makes sure that that the credential does indeed belong to A.
- The PKG also makes sure that this ID is universally unique, i.e., there is no other individual with the same credential (in this case alka@iitb.ac.in).
- If so, it generates a **private key for A,** which is a **function of her ID** and the **private key of the PKG**.
- The PKG then securely transmits the private key to A.
- **Disadvantage:**With knowledge of the PKG's public parameters and A's unique ID, anyone can compute A's public key



3.5 Bilinear mapping

- A bilinear mapping ,B(x,y) maps any pair of elements from one given set to an element in a second set.
- The term bilinear follows from the following property mapping:

$$B(k_1 \times u_1 + k_2 \times u_2, v) = k_1 \times B(u_1, v) + k_2 \times B(u_2, v)$$

- Here u1,u2 and v are elements of the first set and k1 and k2 are integer constants.
- An example of dot product of vectors

Let $u = (2, 4, 1)$ and let $v = (5, 3, 2)$.
Then, $(2, 4, 1) \bullet (5, 3, 2)^T = 24$.
We next verify that the dot product is a bilinear operation.
Now let $u_1 = (2, 4, 5), u_2 = (7, 1, 2), k_1 = 3, k_2 = 4$
So,

$$\begin{aligned} k_1 u_1 + k_2 u_2 &= 3(2, 4, 5) + 4(7, 1, 2) \\ &= (6, 12, 15) + (28, 4, 8) \\ &= (34, 16, 23) \end{aligned}$$

So,

$$\begin{aligned} (k_1 u_1 + k_2 u_2) \bullet v &= (34, 16, 23) \bullet (5, 3, 2)^T \\ &= 264 \end{aligned}$$

Next, consider

$$\begin{aligned} k_1 (u_1 \bullet v) + k_2 (u_2 \bullet v) &= 3 ((2, 4, 5) \bullet (5, 3, 2)^T) + 4 ((7, 1, 2) \bullet (5, 3, 2)^T) \\ &= 3 * 32 + 4 * 42 \\ &= 264 \end{aligned}$$

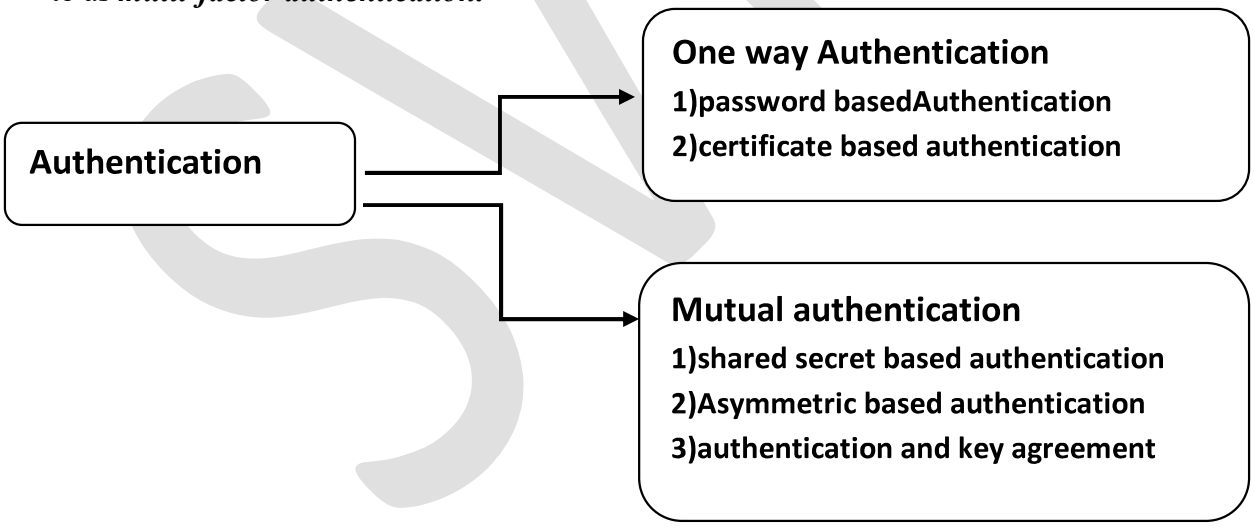
In general,

$$(k_1 u_1 + k_2 u_2) \bullet v = k_1 (u_1 \bullet v) + k_2 (u_2 \bullet v)$$

CHAPTER 2

Authentication-I

- ***Authentication is a process in which a principal proves that he/she/it is the entity it claims to be.***
- The principal is referred to as the ***prover,*** while the party to whom proof is submitted identity verification is called the ***verifier***.
- Authentication may be based on what the principal knows (e.g., a password or a passphrase) or has (an identity card or passport, for example).
- A principal is often a ***human ,a computer, an application, or a robot.***
- In the case of a human principal, authentication may use physical characteristics such as ***voice, a fingerprint, a retinal scan, or even a DNA sample*** — this form of authentication is referred to as ***biometric authentication.***
- With password-based authentication, an individual is often expected to communicate his/her password to a verifying entity. However, in many cases it may not be advisable for the individual to reveal his/her password.
- Instead, he/she may be required to perform some "one-way" cryptographic operation using his/her secret, which cannot be performed without knowledge of it.
- Finally, many authentication systems today use a combination of techniques. This is referred to as ***multi-factor authentication.***



3.6 ONE-WAY AUTHENTICATION

- In client—server communications over a campus, network, for example, it is often the case that the client authenticates itself to the server.
- The server may or may not be authenticated to the client. This is referred to as ***one-way authentication.***

- **Categorized to**
    1. **password based authentication**
    2. **certificate based authentication**

3.6.1 Password-based Authentication

- One of the most common mechanisms to implement authentication is the ***password.***
- To login to a server, a user enters his/her ***login name and password.***
- The password is the secret that is known only to the ***user and server.***
- The ***login name identifies a user,*** while the user's knowledge of thecorresponding ***password constitutes proof*** that he/she is the person with the given login name.
- As shown in below Figthe server uses the login name "Alka" to index into a database of (login name, password pairs),
- It Verifies that the ***submitted password matches*** the one stored against "Alka."
- Drawbacks/threats :
- First, the **password is sent in the clear**, so an attacker can eavesdrop on the messagecontaining the password and later impersonate the real user.
- Second, the ***passwords are stored in unencrypted form in a file on the server.***
- If an internal attacker obtains access to that file, all passwords stored on that server could get compromised.

Figure a : Communicating Password

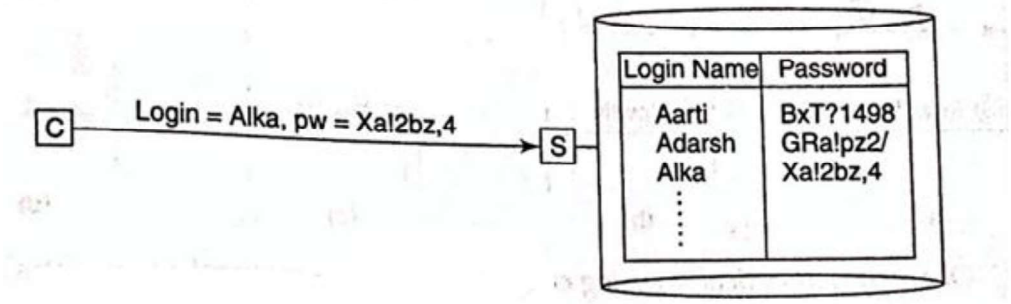


Figure b: Communicating Hash of Password

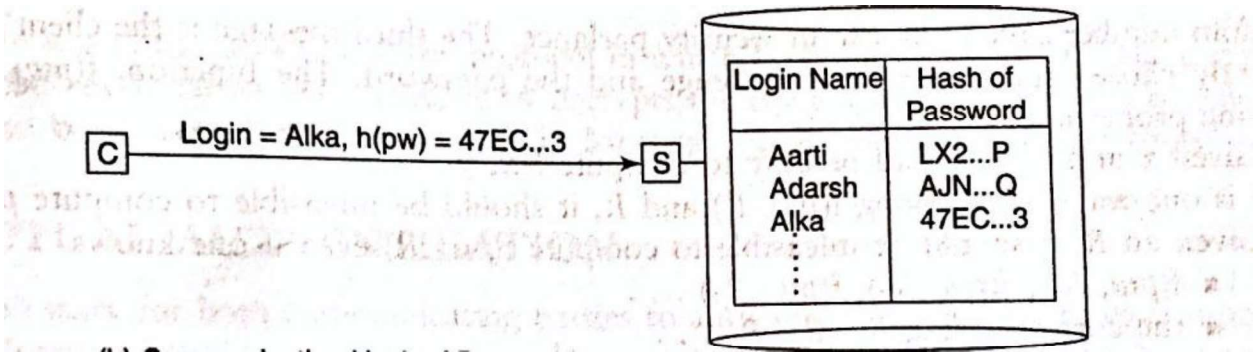


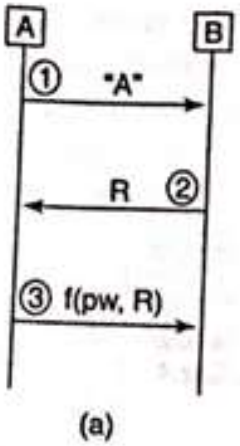
Figure: Password-based one-way authentication

- In Fig(b), the cryptographic hash of the password is stored on the server.
- Also, the login software prompts the user for his/her password and computes its hash which is transmitted.
- The one-way property of the cryptographic hash helps prevent an attacker from deducing user passwords from information in the password file or from communications on the transmission line. However, an attacker could snoop on the communications between Alka and the server and obtain the hash of the password.
- He can, at a later point in time, replay it to the server thus impersonating Alka.
- Such an attack in which one plays back all or a part of one or more previous messages, with the intent of impersonating a legitimate user, is referred to as a **replay attack**.

Challenge response protocol

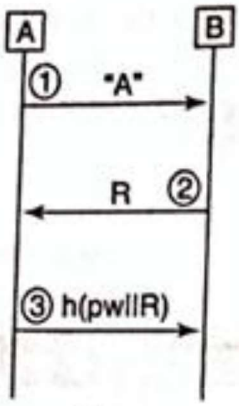
- An effective strategy to thwart a replay attack is for the verifier (in this 'case the server) to offer a fresh challenge to the prover (the client).
- In response, the client **does not communicate its password** but rather proves that it knows the password.
- The server is thus able to verify whether the client is genuine or not.
- The freshness of the challenge requires previous response to answer the current challenge. Such an authentication protocol is commonly referred to as a **Challenge—Response Protocol.**

One-way authentication using challenge—response protocol



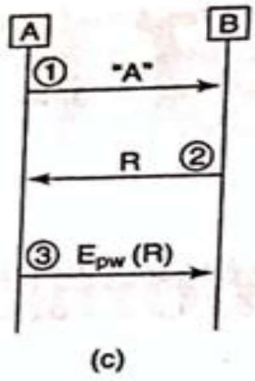
(a)

- In the first message, A conveys its identity.
- The second message contains the challenge from the server.
- The challenge is a random number called a ***nonce(number used only once)*** in security parlance.
- The third message is the client's response - function of the **challenge and the password.**
- The function, ***f(pw, R),*** has the following properties:
- Given x and y, it should be easy to compute f(x, y)
- f is one-way; so, knowing f(pw, R) and R, it should be ***infeasible to compute pw***
- Given an R, it should be ***infeasible to compute f(pw, R***) even if one knows
- f(pw, R1,), f(pw, R1), f(pw, R3) .. .
- the corresponding R1, R2, R3 .. .



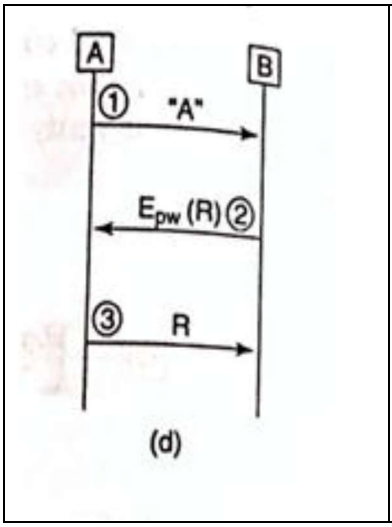
(b)

- An obvious choice for f is the cryptographic hash [Fig. (b)], which is applied over ***the concatenation of the password and the nonce.***



(c)

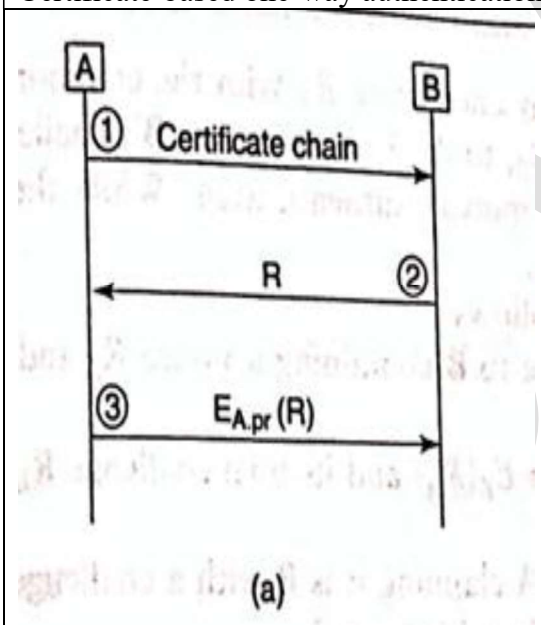
- Another choice is a secret key encryption function with the where ***password is used as a key*** for encryption of random number R [Fig. (c)].



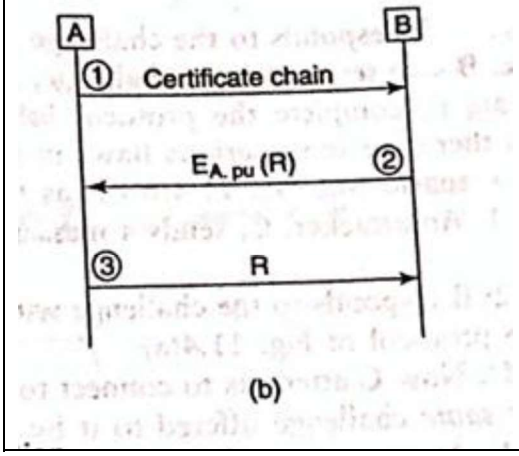
- In Fig. (d), the challenge sent by the server is an encrypted nonce.
- the client would need to decrypt the challenge to obtain the nonce and return it to the server to prove knowledge of his/her password.
- The underlying assumption in these and other protocols is that nonces are random and non-recurring.
- It is the "freshness" of a nonce that precludes a replay attack.
- The term nonce means "used only once."
- In actual implementations, neither the sender nor receiver keeps track of nonces generated or received.

3.6.2 Certificate-based Authentication

Certificate-based one-way authentication



- **MSG 1:**A client need not share a secret with the server but sends a public key certificate.
  - ✓ As shown in Fig.(a), ***A sends her certificate in Message 1.***
  - ✓ B performs certain checks such as on the ***validity period and name of principal***.
  - ✓ He also verifies the ***signature of the CA*** on the certificate.
- **MSG 2:**He then sends his challenge — a nonce **R**.
- **MSG 3:**
- A responds by "encrypting" the challenge with her **private key.**
  - ✓ When B receives $\mathbf{E_{A,pr}(R)}$, he "decrypts" it with **A's public key** and compares it with nonce he transmitted in Message 2.
  - ✓ If they match, he concludes that A has used the private key corresponding to the public key in her certificate.
  - ✓ Assuming that A's private key is safely protected, she must be the entity who created the correct response in Message 3.

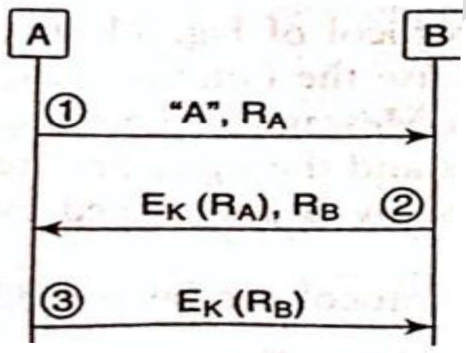
	➢ **MSG 2:**Figure (b) is a slight variation of the protocol in which ***B chooses a nonce, R,*** and ***encrypts it with A's public key*** to create the challenge. ➢ **MSG 3 :*A decrypts*** the challenge and sends it to B. ➢ Authentication of A to B succeeds if what B receives in Message 3, is R, the nonce he just chose.

3.7 MUTUAL AUTHENTICATION

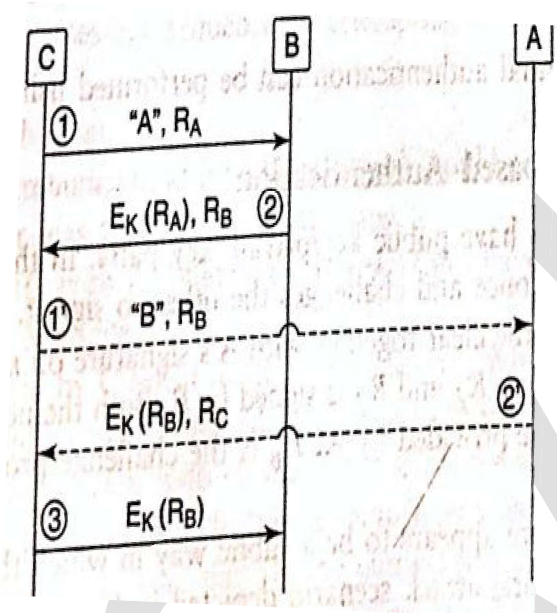
- It is often necessary for ***both communicating parties to authenticate themselves to each other.***
- For example, in Internet banking, it is imperative that a customer interacts with his/her bank and not some entity posing as the bank.
- Likewise, it is important that a bank to verify the identity of the customer.

3.7.1 Shared Secret-based Authentication

- This is a mutual authentication using ***a secret key shared by both parties.***

Figure : Mutual authentication using a shared secret	**Description**
	➢ **Message 1:**A communicates its **identity A**and its challenge in the form of a nonce **R_A.** ➢ **Message 2:**B responds to the challenge by encrypting,**R_A**with **common secret key , K, that A and B share.** ➢ B also sends its own challenge, R_B, to A. ➢ **Message 3:**A's response to B's challenge in the third message appears to complete the protocol for mutual authentication. , there are some serious flaws in it.

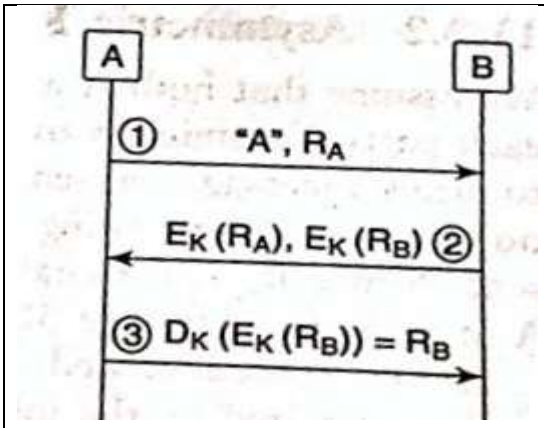
(a) Flawed protocol



(b) Parallel session attack

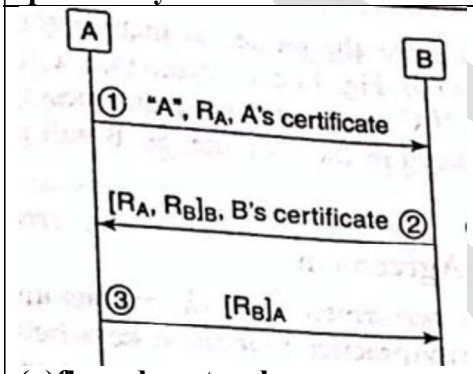
- One attack scenario [figure (b)]is as follows:
- **Message 1:An attacker, C, sends a message to B containing a nonce R_A and claiming to be A**
- **Message 2:** B responds to the challenge with **$E_K(R_A)$** and its own challenge **R_B** as required bythe above protocol of Fig.(a).
- **Message 1'**: Now C attempts to connect to A claiming it is B. with a **challenge R_B.** Note that this is the same challenge offered to it by B in Message 2.
- **Message 2':** A responds to the challenge with **$E_K(R_B)$** and a nonce of its own.
- **Message 3:** C uses A's response **$E_K(R_B)$** to complete the three-message authentication **protocol with B**.

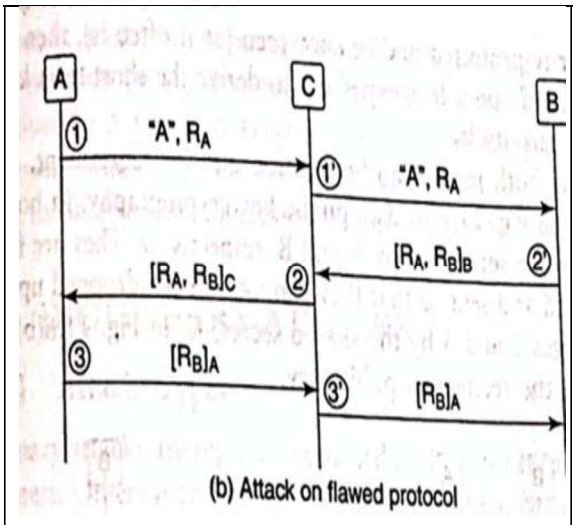
- What has the attacker C accomplished?
- C has successfully impersonated A to B.
- Message 3 was required to complete the authentication of C (posing as A) to B.
- C initiated the authentication protocol with A, presenting to A the same challenge it had received from B.
- A's response to the challenge in Message 2' was used by C to convince B that it was A that was trying to establish communication with him. This attack is termed a **Reflection Attack** since a part of the message received by an attacker is reflected back to the victim.
- In this case, the reflected message fragment is $E_K(R_B)$. This attack is also called a **Parallel Session Attack**

 **(c) Corrected protocol**	➢ Figure c: the protocol might require the responder to encrypt his challenge, while the initiator would be required to decrypt herchallenge. ➢ **Encrypting both R_A and R_B**

3.7 .2 Asymmetric Key-based Authentication

- We assume that both ***A and B have public key/private key pairs.***
- The ***notation $[m]_A$means a messagem, sent together with A's signature on m.***
- In the protocol of Fig. (a), each party transmits its own nonce and challenges the other to sign it.

Asymmetric key based authentication /public key based authentication	**Description**
 **(a)flawed protocol**	➢ figure (a) shows Mutual authentication using public key cryptography /asymmetric based authentication ➢ **MSG1: Identity of A, challenge sent by A , which is R_A , A's certificate** ➢ **MSG2: the string obtained by concatenating R_A , R_B signed by B, B's certificate.** ➢ **MSG3: R_B is thechallenge signed by A(encrypted using A's private key)**
	Figure b shows attack on flawed protocol: ➢ **MSG1: A initiates communication with C, sending the challenge R_A.** ➢ **MSG 1': C initiates communication with B using the same nonce R_A** ➢ **MSG2'**: B responds to "A's challenge" and includes a challenge of his own, **R_B** ➢ **MSG 2**: C responds to A's challenge and uses B's random number , R_B, as his challenge to A. ➢ **MSG3**: A responds to C's challenge (which was actually generated by B). A thus



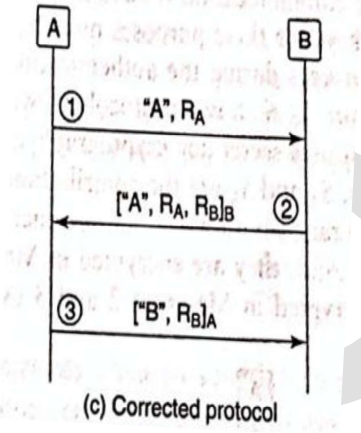
(b) Attack on flawed protocol

completes the mutual authentication protocol with C.

- **MSG3'**: C forwards A's response to B.

It is clear from Fig.(b) :

- That Message 1' is sent by C includes A's identity. And attempts to convince B that A intends to talk to him.
- B responds to what appears to be A's intention to communicate with him.
- Note that, in the current scenario, A may not wish to communicate with B and is not aware that C is attempting to do so on her behalf.
- Yet, after B receives Message 3', he feels A intends to communicate with him since Message 3' contains her signature on a nonce chosen by him.



(c) figure (c): Corrected protocol

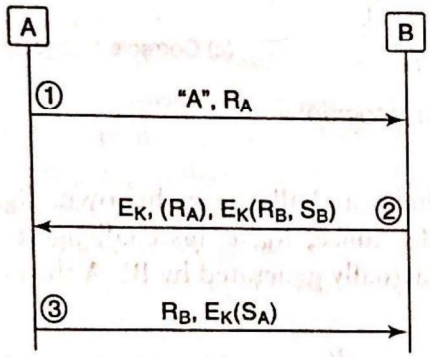
- One solution to the above problem is for the entities to include the **identity of the recipient in all messages signed.**
- This is shown in Fig.(c).
- **MSG 2:**the string obtained by concatenating nonce**R_A** and **R_B** is **signed by B is sent** . (Means encrypted using B's private key)
- **MSG 3:** R_B is the challenge provided by B and signed by A in response .(means encrypted using A's private key)

3.7.3 Authentication and Key Agreement

- In previous sections, authentication was performed using operations involving a long-term, shared secret or a private key.
- Since private key operations are very expensive, the communication can be integrity-protected and/or encrypted using short term keys or session keys .
- **S_A and S_B** are the contributions to the secret key by A and B, respectively.
- They are freshly chosen random numbers that are encrypted and sent so that they cannot be eavesdropped upon
- The key finally chosen could be a simple function of S_A and S_B, $S=S_A$(xor) S_B.

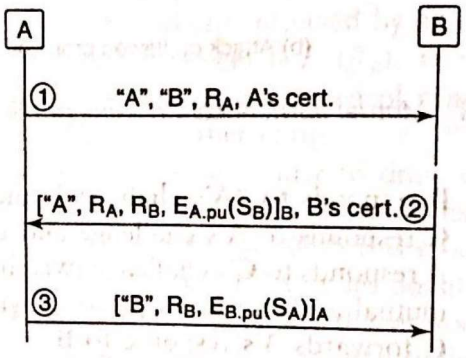
Mutual authentication using session key

Using secret key cryptography



(a) Using secret key cryptography

Using public key cryptography



(a) Using public key cryptography

Use of Timestamps

- The use of nonces was introduced to prevent replay attacks.
- Basically, each party generates a nonce which is used as a fresh challenge to the other party.
- The recipient is often expected to sign or encrypt the challenge using a secret known to only the recipient (and the sender).
- The key idea here is the freshness of the nonce — if nonces were re-used, the response to the challenge could be replayed from a previous session.
- An alternative to nonces are timestamps.
- Ideally**, by securely "stamping" a message with the current time,** you convince the receiving party of its freshness.
- Below Figureshows the use of timestamps in conjunction with public key cryptography for authentication.

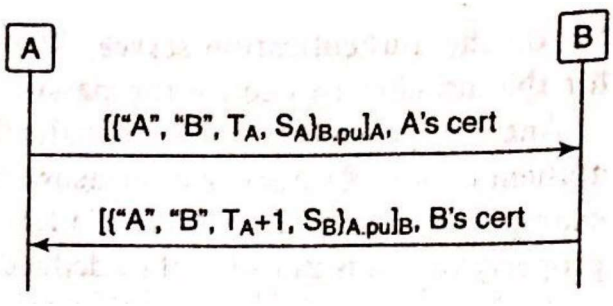


Figure :Mutual authentication with timestamp

- In Message 1, A inserts a timestamp, TA, in her message and signs it.
- B, on receiving the message, checks whether the timestamp is sufficiently recent and then verifies the with timestamps signature.
- He increments the received timestamp, inserts it into his response message to A, and signs the message.
- The notation $\{m\}x._{pu}$, denotes a message, encrypted using the public key of X
- If the clocks maintained by A and B are synchronized, the timestamp in Message 1 signed by A convinces B that the message was freshly created by A.
- The timestamp implicitly serves as A's challenge to B.
- By signing the incremented timestamp, B hopes to satisfy A that he is indeed responding to her message.

3.8 DICTIONARY ATTACKS

3.8.1 Attack Types

- Dictionary attacks are typically launched in the context of passwords.
- Some passwords have too few characters.
- Others may be common celebrity names, place names, etc.
- Some individuals use permutations of characters in the names of their near relatives or friends so that they are easily memorizable.
- Based on such clues, an attacker can build a dictionary of strings which are potential passwords of his/her victim.

Password	**Reason for Weakness**
123 or abcd	Common default passwords
Sm!t	Anything less than 8 characters is too short
nahkhkurhahs	Celebrity name — Shahrukh Khan spelt backwards

23-05-86	Birthdays/anniversaries are convenient but would almost always be part of the attacker's password dictionary
ashyea	Permutation of letters in mother's or spouse's name, (Ayesha name in this example)is a poor choice especially if the attacker has personal information about his victim
Kolkata	Place names are often part of password dictionaries

- There are two types of dictionary attacks —
  1. **on-line**
  2. **off-line.**

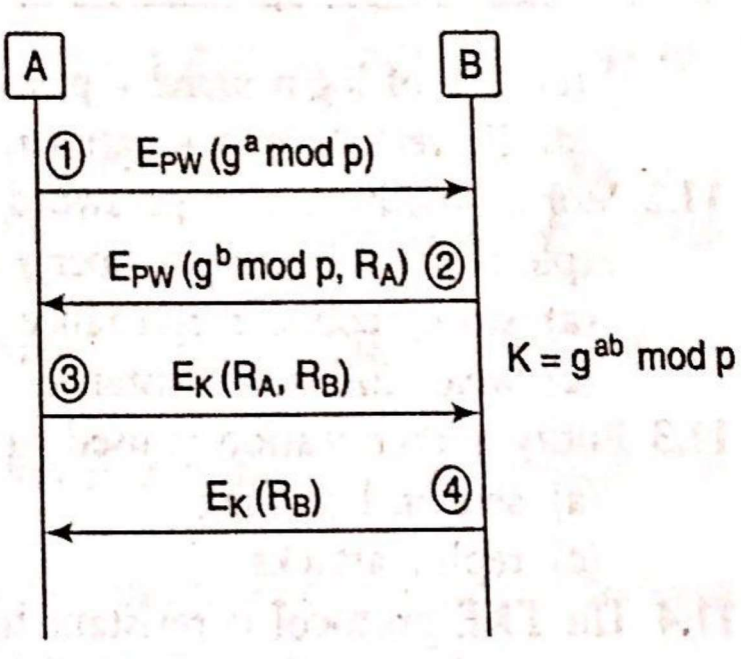
1. **on-line attack:**
   - ✓ In on-line attacks, an intruder attempts to login to the victim's account by using the victim's login name and a guessed password.
   - ✓ There is usually a system-imposed limit on the number of failed login attempts. So, unless the attacker is particularly insightful or lucky.
   - ✓ an on-line attack has a limited chance of success.

1. **off-line attack:**
   - ✓ Unlike an on-line attack, an off-line dictionary attack leaves few fingerprints.
   - ✓ One possibility is for the attacker to get a hold of the password file.
   - ✓ Passwords are typically transformed in some way (by, for example, performing a cryptographic hash on them) before being stored on the authentication server.
   - ✓ The cryptographic hash is a one-way function, so it is not easy for the attacker to deduce the password given its cryptographic hash.
   - ✓ Another possibility is for the attacker to **eavesdrop on the communication link during authentication.**
   - ✓ The attacker could use his/her dictionary of passwords to implement the following attack.

```
// Let D be an array containing the dictionary
// Let F denote f(pw, R) where pw is the client's password
// Let f be the number of permissible guesses (size of D)
found = false
 i = 0
while ( ~found&&i< n)
{
        x = f(D[i], R)
        if (x = F)
        {
                print ("CORRECT PASSWORD ")
                found = true
        }
}
```

3.8 .2 Defeating Dictionary Attacks

- One approach to frustrating a dictionary attack is to increase the cost of performing such an attack. The cost is the time to successfully complete the attack.
- The most time-consuming operation in each iteration of the dictionary attack program is f(D[i], R). Hence, to decrease the attacker's chance of success, the function f(D[i], R) could be made more computationally expensive.
- Suppose, for example, instead of the function f being a simple cryptographic hash, it was the cryptographic hash, h, applied successively a hundred times,that is,
- **h (... h (h (D[i], R)) .....)**
- If the above function were used in the loop of the program, we would expect the program to run about 100 times slower.
- A protocol that virtually eliminates off-line dictionary attacks is the Encrypted Key Exchange (EKE) protocol.
- This is a password-based protocol that combines Diffie—Hellman key exchange with mutual authentication based on a shared secret.
- the Diffie—Hellman protocol is vulnerable to a man-in-the-middle attack which is due to the unauthenticated exchange of "partial secrets", **g_a mod p and g_b mod p.**
- To mitigate this attack, EKE uses a novel idea — each side transmits its partial secret after encrypting it.
- The encryption key, **PW,is the hash of the password.**
- Below Figure shows the four messages that are exchanged in EKE.



- After MSG 2, both sides should be able to compute the **new session key k = g^{ab} mod p** denoted by K in the figure.

- Mutual authentication is accomplished using the familiar challenge—response protocol in which each side selects a random nonce and challenges the other side to encrypt it with the newly computed session key.
- It is assumed that the victim's password is "weak," that is, it can be guessed using moderate effort. That being the case, basic password-based mutual authentication protocolscould yield to an off-line dictionary attack.
- Assume that an attacker has access **to $E_{pw}(g^a \bmod p)$ and $E_{pw}(g^b \bmod p)$.**
- The attacker would attempt to guess the victim's password and hence PW.
- If the attacker guessed correctly, he/she would be able to obtain the true values of
- $g^a \bmod p$ and $g^b \bmod p$. But even so, he/she would not be able to obtain the session key, $g^{ab} \bmod p$.
- This is so, since the computational Diffie—Hellman problem is infeasible in large groups that are carefully chosen,
- Thus, EKE is not susceptible to an off-line dictionary attack.
- Another property of EKE is that it provides perfect forward secrecy
- A protocol is said to have perfect forward secrecy if it is not possible for an attacker to decrypt a session between A and B even if he/she records the entire encrypted session and then at a later point in time (say a week later) obtains or steals all relevant long term secrets of A and B.

Chapter 3

Authentication-II

- **Key Distribution Centre (KDC) – a trusted third party that shares long-term keys-with clients and servers alike.**
- Two protocols that make use of a KDC—the ***Needham-Schroeder protocol and Kerberos.***
- We then look at the biometric authentication as a complement to and, in some cases, as a substitute for cryptographic authentication.

3.9 CENTRALISED AUTHENTICATION

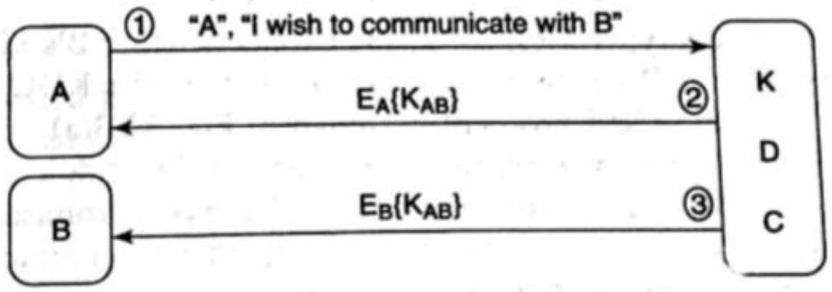
- There are a number of advantages of secret key cryptography over public key cryptography inauthentication protocols.
- First, digital certificates and a public key infrastructure (PKI) are needed in support of public key cryptography.
- There is a substantial cost to set up and maintain a PKI.
- Also, public key/private key operations are relatively slow compared to secret key operations.
- In secret key cryptography,If the entity communicates with a large number of other entities over time, it must share a secret with each of those parties.

- Managing and securely storing a large number of keys is a nontrivial task.

Note:Throughout this chapter, $E_A\{m\}$denotes a message encrypted using A's long, term secret shared with the KDC(derived using A's password). $E_{AB}\{m\}$denotes a message encrypted using the session key

- One approach to alleviating the risk is to employ a ***trusted third***party which, in this case, functions as a ***key distribution centre*****(KDC).**
- Each user registers with a KDC and chooses a password.
- A ***long-term secret,***which is a function of the password, is to be exclusively shared by that user and the KDC.
- The main function of the KDC is to securely communicate a fresh, common session key to the two parties who wish to communicate with each other.

- In Figbelow,



- ***Message 1:*** **A** informs the KDC that it intends to communicate with B
- The KDC generates a random secret, $\mathbf{K_{AB}}$**,**and dispatches this to **A** and **B** through two encrypted messages.
- ***Message 2*** is encrypted using the long-term secret, $\mathbf{K_A}$**,** that A shares with KDC.
- ***Message 3*** is encrypted with $\mathbf{K_B}$, the secret shared between B and the KDC.
- Both A and B decrypt their messages and obtain the ***short-term session key.***
- A and B thenall subsequent messages during the session using K_{AB}**.**
- The above Figurewas meant to convey the general idea in using a KDC but the protocol is susceptible to numerous types of replay and man-in-the-middle attacks.

3.9 THE NEEDHAM-SCHROEDER PROTOCOL

- Below Figure 3.10(a) enhances the protocol of Fig 3.9 to provide **mutual authentication** by including, **challenge—response phase**.
- Here, both sides proceed to challenge the other to prove knowledge of the session key, K_{AB}.
- The challenge is a **nonce.**
- The response involves ***decrementing the nonce*** and ***encrypting the nonce*** with the **session key, K_{AB}.**
- There are four versions
    1. **Preliminary version 1**
    2. **Preliminary version 2**
    3. **Preliminary version 3**
    4. **Final version**

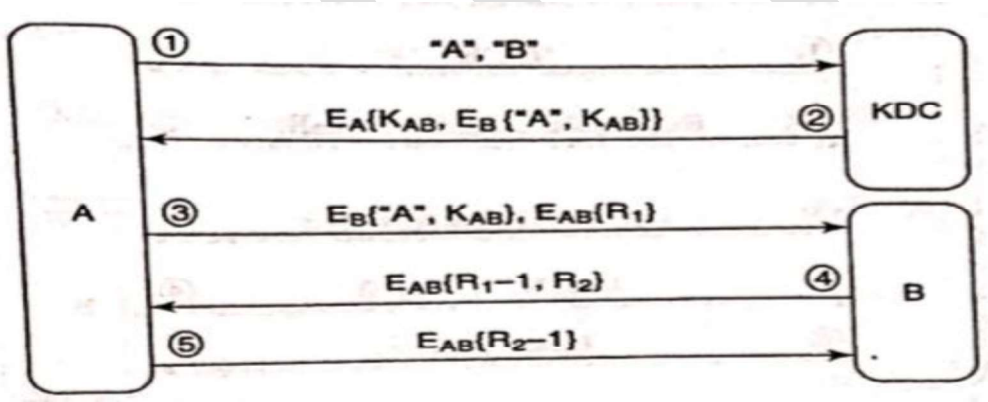


Figure 3.10 a

Preliminary Version 1

- ***MSG 1:***Identity of A and B
- ***MSG 2:*** *In* ***response KDC***encloses the ticket to B.ie $\mathbf{E_B\{A,K_{AB}\}}$
- ***MSG 3:*** A then forwards the **ticket**along with the challenge to B (R_1)
- ***MSG 4: R1 is decremented and B challenges A with R2.***
- ***MSG 5:R2 is decremented by A and forwarded to B.***

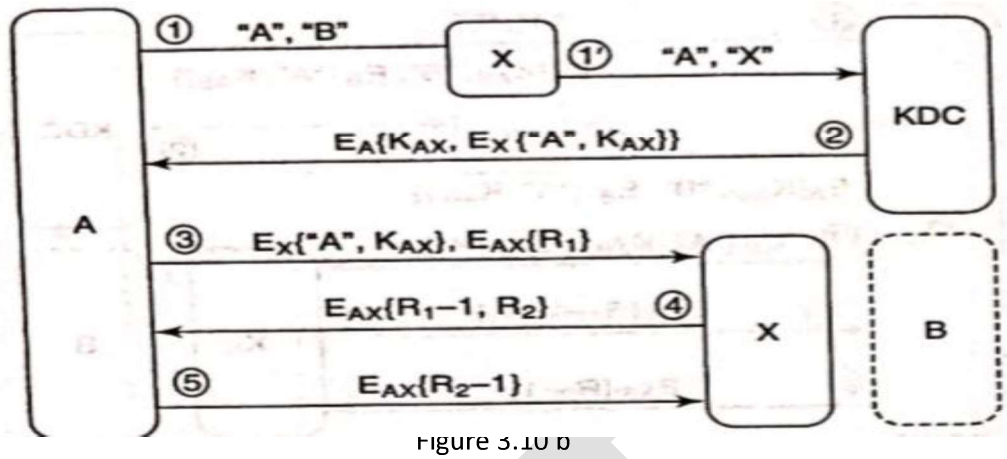


Figure 3.10 b

Man in the middle attack on Preliminary Version 1

- The protocol in Fig. 3.10(a) is susceptible to an impersonation attack shown in Fig. 3.10(b).
- Theattacker, X, is an insider who shares a long-term key with the KDC.
- The attacker, X, intercepts Message 1, substitutes "B" for "X" and sends the modified message to the KDC.
- In response, the KDC creates a ticket encrypted with X's long-term key and sends it to A in Message 2.
- Now X Intercepts Message 3. He decrypts the ticket using the long-term secret he shares with the KDC. He thus obtains the session key, K_{AX}.
- Message 3 also contains A's challenge R1.
- X uses the session key, K_{AX} to decrypt the part of the message containing A's challenge. He successfully responds to A's challenge in Message 4.
- Thus, X successfully impersonates B to A.

Preliminary Version 2

- A simple fix to the protocol is to include B's identity in the encrypted message from the KDC to A (Message 2). The modified message is
- **$E_A\{K_{AB}$, "B", E_B {"A", $K_{AB}\}\}$**

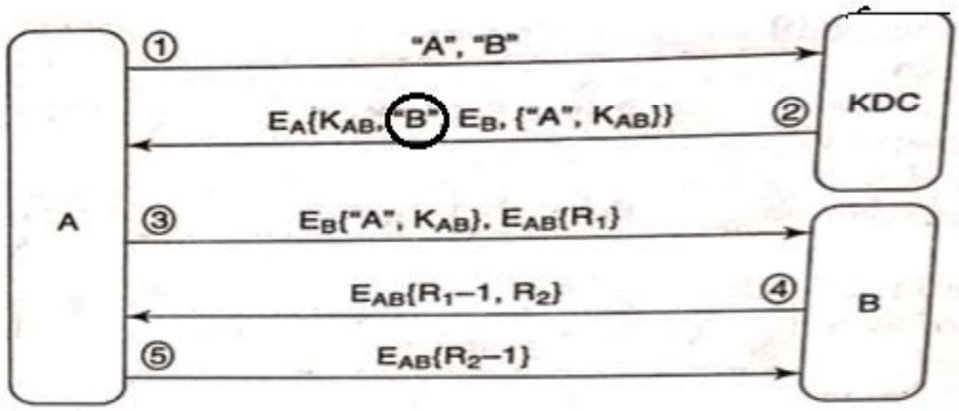


Figure 3.11 a Preliminary version 2

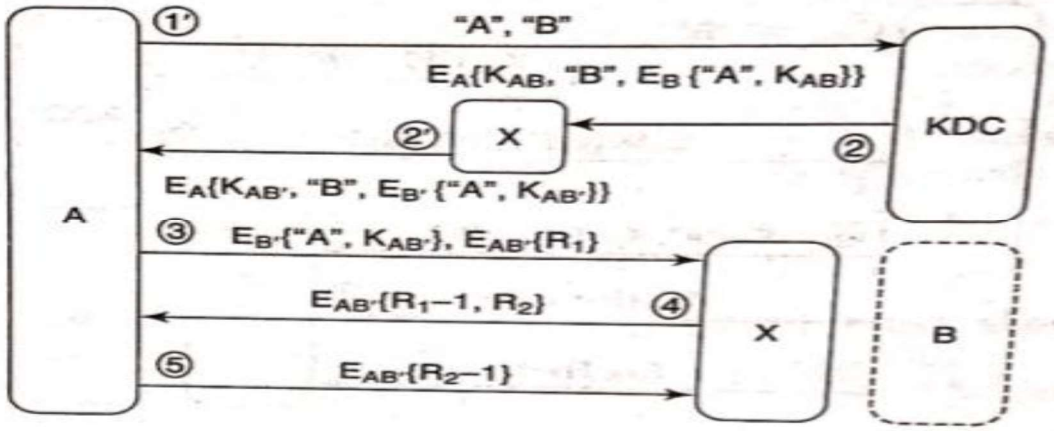


Figure 3.11 b Man in the middle attack and replay attack.(B' is the old key , B is the new key)

Preliminary version 2

- ***MSG 1:***Identity of A and B
- ***MSG 2:*** *In* ***response KDC***encloses the ticket to B.ie **$E_B\{A,K_{AB}\}$, b's identity**
- Now, after A receives and decrypts Message 2, she checks whether B's identity is contained inside the message. The presence of B's identity confirms to A that the KDC knows that A wishes to communicate with B.
- ***MSG 3:*** A then forwards the **ticket**along with the challenge to B (R_1)
- ***MSG 4: R1 is decremented and B challenges A with R2.***
- ***MSG 5:R2 is decremented by A and forwarded to B.***

Man in the middle attack and replay attack on preliminary version 2

- The attacker, X, does the following:
- X eavesdrops and records many of A's sessions with the KDC and with B over a period of time and steals B's password or long-term key.
- B recognizes that his password has been stolen and immediately reports the incident to the KDC.
- He obtains a **new long-term key, K_B,** which he uses subsequently.
- Even then , the following scenario shows X successfully impersonates B to A.

1. A wishes to communicate with B and sends Message 1 in Fig.3.11 (b).
2. X intercepts the KDC's response (Message 2) and instead plays a previous recording of Message 2.
3. This message contains a ticket encrypted with **B's old key, K_B'.**
4. X then intercepts Message 3 from A, which contains the old ticket and a fresh challenge to B. Because X has access to B's old key, he can decrypt this ticket and recover the session key, K_{AB}.
5. Because X knows **K_{AB}',** he can respond to A's challenge in Message 4. X's response is exactlywhat A expected to receive from B. Hence, A is convinced that she is talking to B.

Preliminary Version 3

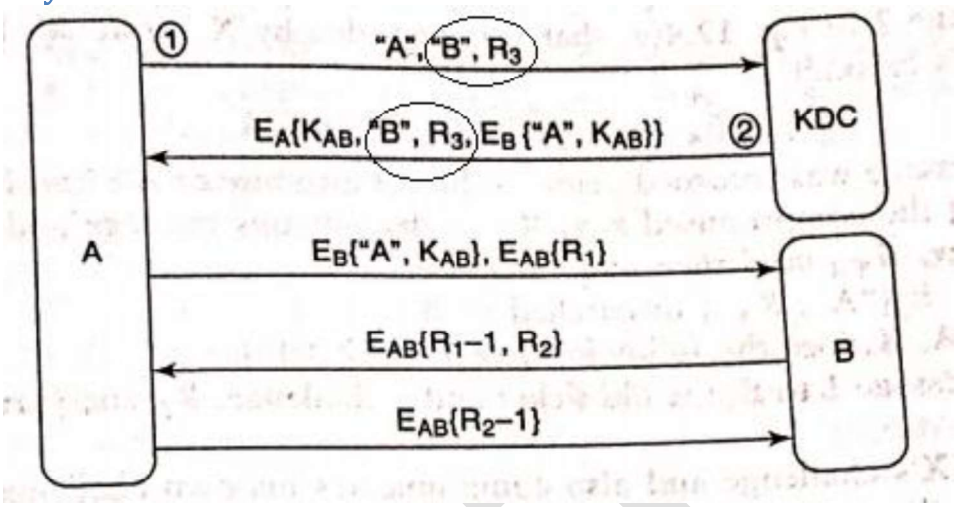


Figure 3.12 a Needham-Schroeder protocol: Preliminary version 3

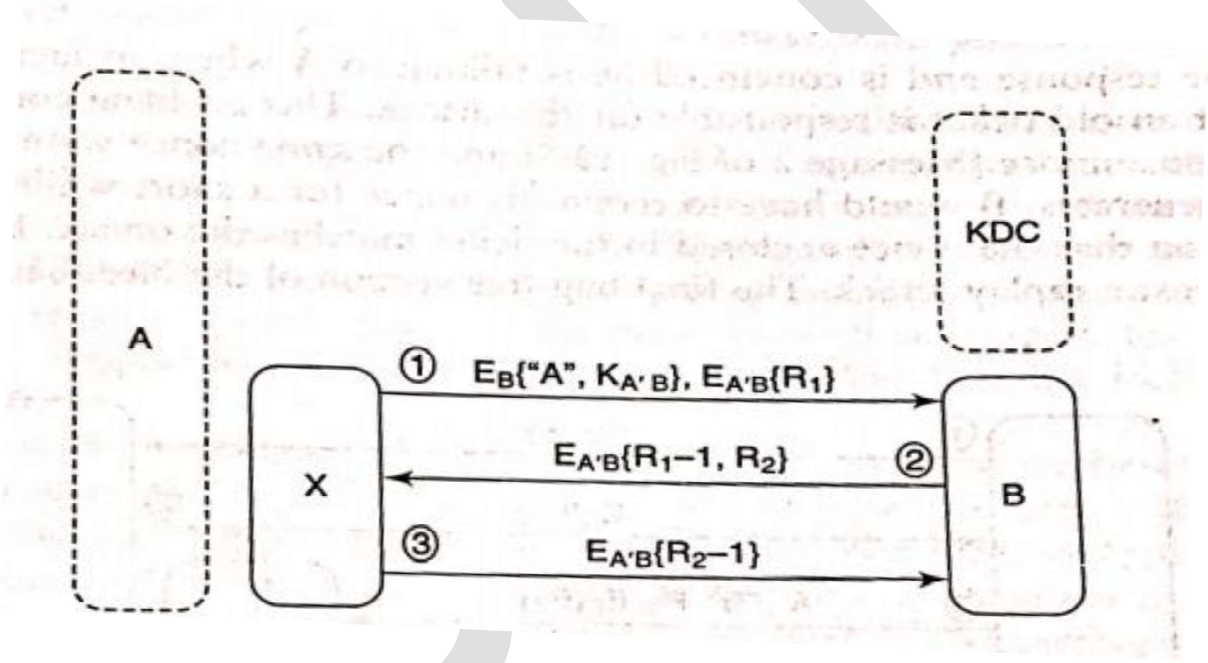


Figure 3.12 b Replay Attack on Preliminary version 3

- **A' is lost password generated key**
- **A is new password generated key**

Preliminary version 3

- We can fix this vulnerability in version 2 by ensuring the *freshness* of Message 2.
- This is accomplished by **A sending a (fresh) nonce in Message 1** [Fig. 3.12(a)] and receiving **confirmation of its receipt by the KDC in message 2.**

Replay attack on preliminary version 3

- The version 3 is still not secure despite the modifications made.
- X could still attack the protocol by recording previous messages and selectively replaying them when the right opportunity presents itself.
- Such as he attempts to steal A's password or long-term key.
- Assume again that A suspects the compromise of her password and promptly reports this to the KDC without delay.
- X then manages to steal A's long-term key that she shares with the KDC and perform a**n impersonation attack.**
- **A'** is the old password generated key
- **A** is new password generated key.
- Using the compromised (old)key, X can decrypt this message and recover
- Theold session key, $\mathbf{K_{A'B}}$
- The old ticket $\mathbf{E_B\{A,K_{A'B}\}}$
- To impersonate A, X does the following [see Fig.3.12(b)]:
  1. X sends,the old ticket and a challenge, R1, encrypted with the old session key.
  2. Message 1 to B,
  3. B responds to X's challenge and also communicates his own challenge, R2.
  4. Because X has the session key, he responds to the challenge by encrypting R2 with the session key.
- B receives the response and is convinced he is talking to A(impersonated by X).

Needham Schroeder Protocol: Final Version

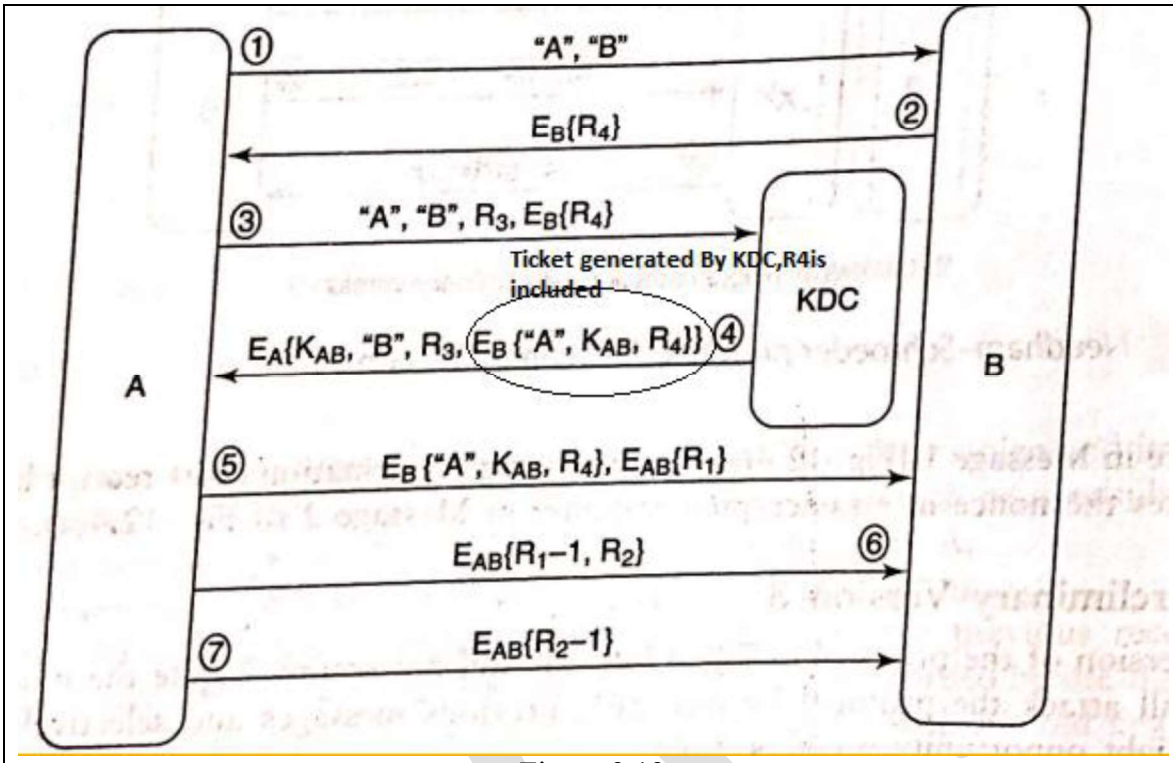


Figure 3.13

- The problem in previous versions could be fixed if B were allowed to choose a nonce(R4)andthe same nonce were enclosed by the KDC in the **ticket it generates.**
- ***MSG 1:***Identity of A and B sent from A to B
- ***MSG2: random number R4 generated by B***
- ***MSG3:A forwards his challenge as R3 along with R4.***
- ***MSG 4:KDC generates ticket and includes R4***
- ***When B receives this ticket in msg 5, B can verify the random number/nonce generated in msg 2 is same or not.***

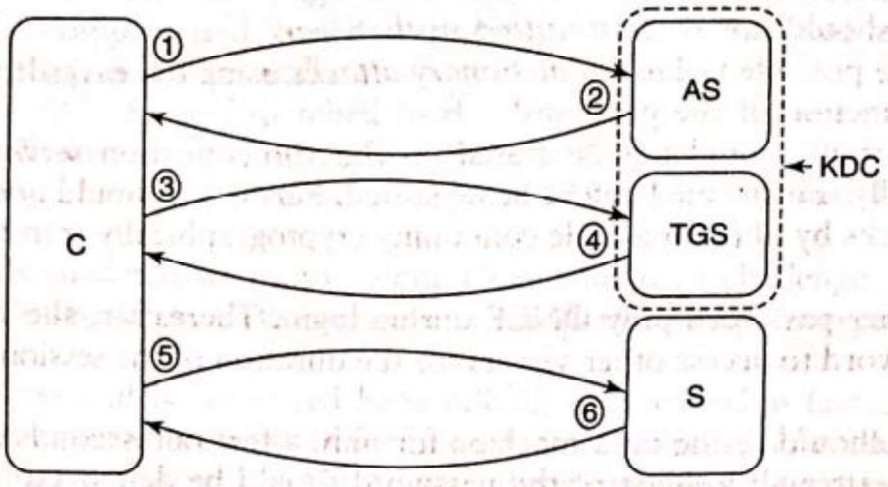
3.10 Kerberos

- A user could use the same password for all servers but distributing and maintaining a password file across multiple servers poses a securit risk.
- A password-based system should ensure the following: y

1. The password should not be transmitted in the clear.
2. It should not be possible to launch dictionary attacks
3. The password itself should not be stored on the authentication server, rather it should be cryptographically transformed before being stored.
4. It should not be possible to launchdictionary attacks by obtaining a file containing cryptographically transformed versions of the password.
5. A user enters her password only ONCE during login. Thereafter, she should not have to re-enter her password to access other servers for the duration of the session. This feature is called **single sign-on.**
6. The password should reside on a machine for only a few milliseconds after being entered bythe user.

The Kerberos protocol elegantly addresses many of these issues.

- Developed at MIT, Kerberos has been through many revisions.
- The latest is Kerberos Version 5.
- The KDC used in the Needham—Schroeder protocol is logically split into two entities here — the Authentication Sewer (AS) and the Ticket Granting Server (TGS).
- The sequence of messages exchanged between the client (C), the Kerberos servers (AS and TGS) and the requested server(S) is shown in Fig.3.14 .
- **There are three steps — each involving two messages**



① C request Ticket-Granting Ticket
② C receives Ticket-Granting Ticket
③ C request Service-Granting Ticket
④ C receives Service-Granting Ticket and session key
⑤ C authenticates itself to S
⑥ S authenticates itself to C

Kerberos message sequence

Step 1: Receipt of Ticket-Granting Ticket

Message 1 $C \rightarrow AS$: "C", "TGS", Times, R_1

Message 2 $AS \rightarrow C$: "C", $Ticket_{TGS}$, E_C {"TGS", $K_{C,TGS}$, *Times*, R_1}

where

$Ticket_{TGS} = E_{TGS}$ {"C", "TGS", $K_{C,TGS}$, *Times*}

Step 2: Receipt of Service-Granting Ticket

Message 3 $C \rightarrow TGS$: "S," *Times*, $Authenticator_C$, $Ticket_{TGS}$, R_2

where

$Authenticator_C = E_{C,TGS}$ {"C", TS_1}

Message 4 $TGS \rightarrow C$: "C", $Ticket_S$, $E_{C,TGS}$ {"S", $K_{C,S}$, *Times*, R_2}

where

$Ticket_S = E_S$ {"C", $K_{C,S}$, *Times*}

Step 3: Client-Server Authentication

Message 5 C → S: $Ticket_S$, $Authenticator_C$
where
$Authenticator_C = E_{C,S}$ {"C", TS_2}

Message 6 S → C: $E_{C,S}$ {TS_2 + 1}

Step 1: Receipt of Ticket-Granting Ticket

Message 1

C →AS

- In Message 1, the client informs the AS that it wishes to communicate with the TGS.
- *"Times"* field specifies the start time and expected duration of the login session.
- "C," is the ID of the user/client who has logged in.
- R1 is a nonce generated by C

Message 2

AS → C

- The response from the AS (Message 2) contains a session key, **$Kc_{,TGS,}$** to be used for communication between C and the TGS.
- This key is encrypted with the long-term key, KC known to C and the AS.
- This key is a function of the user's password.
- AS encrypts the nonce, that it received in Message 1.
- The nonce is used to prevent replay attacks.
- The AS also includes a TGT **(Ticket $_{TGS}$)** in connection with C's request.

Step 2: Receipt of Service-Granting Ticket

Message 3

C→TGS

- In Message 3, C forwards the TGT (Ticket $_{TGS}$), Authenticator $_C$ to the **TGS**
- **Using** this Ticket $_{TGS}$,**TGS server** extracts the session key, **$K_{C,TGS,}$** known only to **C** and the TGS.
- As shown above, the **Authenticator $_C$** encrypts the current time (timestamp) and ID using***$K_{C,TGS}$***

Message 4

TGS→C

- The *TGS* generates a fresh session key, Kc,s, to be shared between **C** and S.
- This key is encrypted using the session key **$K_{C,TGS,}$** so only **C** can decrypt it.
- The fresh nonce, **R2, from C is also encrypted by** the TGS using **K $_{C,TGS}$**
- This convinces **C** that the received message is from the **TGS**
- Finally, the fresh session key Kc,s is enclosed in a *service-granting ticket* to be forwarded by C to **S.**

- The service-granting ticket is encrypted with the **long-term secret shared between the TGS and S.**

Step 3: Client-Server Authentication

Message 5

C →S

- C forwards to **S** the ticket containing the session key, $K_{c,s}$.
- **C** also creates and sends to **S** an authenticator by encrypting a timestamp with the session key *Kc,s*

Message 6

S→ C

- **S** retrieves Kc,s from the service-granting ticket.
- **S** verifies the authenticator from **C.**
- **S** then increments the timestamp and encrypts it with the fresh session key.
- The encrypted timestamp serves to authenticate S to C.

3.11 BIOMETRICS

3.11.1 Preliminaries

- A biometric is a ***biological feature***or ***characteristic of a person*** that *uniquely identifies* him/her over his/her lifetime.
- Common forms of biometric identification include face recognition, voice recognition, manual signatures, and fingerprints.
- More recently, patterns in the iris of the human eye and DNA have been used.
- Behavioural traits such as keystroke dynamics and a person's walk have also been suggested for biometric identification.
- Biometric forms were first proposed as an alternative or a complement to passwords.
- Passwords are based on what a user knows.
- Commonly used ID cards, including personal smart cards, are based on what a person has.
- A biometric, on the other hand, links the identity of a person to his/her physiological or behavioural characteristics.
- The two main processes involved in a biometric system are enrolment and recognition.
    1. ***Enrolment:***
        - ✓ In this phase, a subject's biometric sample is acquired.

- ✓ The essential features of the sample are extracted to create a ***reference template.***
- ✓ Sometimes multiple samples are taken and multiple templatesare stored to increase the accuracy of a match in the subsequent recognition phase.

2. **Recognition:**
   - ✓ A fresh biometric sample of a person is taken and compared with the reference templates to determine the extent of a match.

- Biometrics are used in two different scenarios:
  1. **Authentication**
     - ✓ Biometric system stores (login name and biometric sample)
     - ✓ authentication involves a one-to-one match
  2. **Identification**
     - ✓ As in authentication, a biometric sample of the subject is taken but the subject's identity is not presumed to be known beforehand.
     - ✓ It is assumed that a database of biometric samples of several users already exists.
     - ✓ The subject's biometric sample is compared with the samples in the database to determine if a match exists with any one of them.
     - ✓ identification involves a one-to-many match
     - ✓ A typical application of authentication is in access control, while identification finds widespread uses in forensics/criminology.
- The characteristics of a good biometric include the following:
  - ✓ **Universality:** All humans should be able to contribute a sample of the biometric. For example, the speech-impaired may not be able to contribute towards a voice recognition system.
  - ✓ **Uniqueness.** Biological samples taken from two different humans should be sufficiently different that they can be distinguished by machine intelligence.
  - ✓ One litmus test of uniqueness is whether the biometric samples of two identical twins serves to unambiguously identify them.
  - ✓ **Permanence**. The biometric should not change over time. The samples acquired during enrolment may be several years old (even tens of years old). Still, it should be possible to detect a match between the newly acquired sample and that stored in a database of samples of thousands of individuals.
  - ✓ Permanence is not a given. For example, a person's voice may temporarily change due to a cold, the manual signature of a senior

citizen may change and fingerprints of people in certain professions may wear out over time.

Case studies

1. **Fingerprints**
2. **Iris scan**

1. **Fingerprints:**
   - ✓ A fingerprint is an impression left by the ridges and valleys of a human finger.
   - ✓ Each individual fingerprints exhibit distinctive patterns.
   - ✓ During the enrolment and recognition phase ,an image of the fingertip is taken by placing it on the plane surface of a scanner.
   - ✓ During the recognition phase the input template must match with the patterns stored in database.
   - ✓ The simplest approach involves identification of distinctive patterns formed by ridges.these are called as singularities.
   - ✓ They are:arch,loop and whorls.
   - ✓ Arch : the ridge starts from one side of the finger and forms an arc and ends on other side.
   - ✓ Loop: the ridge starts and ends at the same side of the finger.
   - ✓ Whorls:appear as closed cycles or spirals in a fingerprints.

2. **Iris scan**
   - ✓ The **iris is a thin opaque diaphragm of smooth muscle** situated in front of the lens in the human eye.
   - ✓ Its annular shape surrounds the pupil.
   - ✓ The intricate patterns on the iris appear to be unique.
   - ✓ Two identical twins have iris pattern s that are different as those of two unrelated individuals.
   - ✓ The patterns of an iris are also stable with age.

Chapter 4
IPSec-Security at the Network Layer

4.1 SECURITY AT DIFFERENT LAYERS: PROS AND CONS

- Security may be implemented at different layers of the OSI model.
- Security is commonly implemented
  1. In the network layer or
  2. between the transport and application layers and/or
  3. within the application
- **IPsec** — the best-known protocol for providing security at the network layer.
- There are two main **IPSec protocols** and their modes of operation.

4.2 IPSec IN ACTION

- IPSec has been designed by committee in the late 1990's, it was intended to protect against sniffing, spoofing, hijacking, and Denial of Service (DoS) attacks.
- It provides a host of services including:
  1. ***Data origin authentication and data integrity***
  2. ***Protection from replay attacks***
  3. ***Data confidentiality and***
  4. ***Partial traffic flow confidentiality.***
- The end-points of the protocol can be **two hosts, two gateways, or a host and a gateway.**

4.2.1 IPSec Security Associations

- Before two parties can communicate ***securely***, they need to establish a ***Security Association* (SA)** with each other.
- A node (either host or gateway) may establish IPSec SAs with several nodes.
- An **SA**is uniquelyidentified by a combination of a **32-bit *Security Parameter Index*(SPI) and the IP address**of the connection endpoint.
- The information in an SA includes :
  1. *Lifetime* of the association
  2. IPSec *Mode* — transport or tunnel
  3. *Cryptographic parameters*(the algorithm used for encryption, if any, and for computing the integrity check, together with the keys)
  4. 32-bit *sequence number* — the first packet protected by a newly established SA bears the
  5. *sequence*number 1, the sequence number gets incremented for each new packet sent.
  6. *anti-replay window*
- Each IPSec packet contains a value of SPI in its header. This is used by the receiving node to identify the SA to be used for processing the packet.

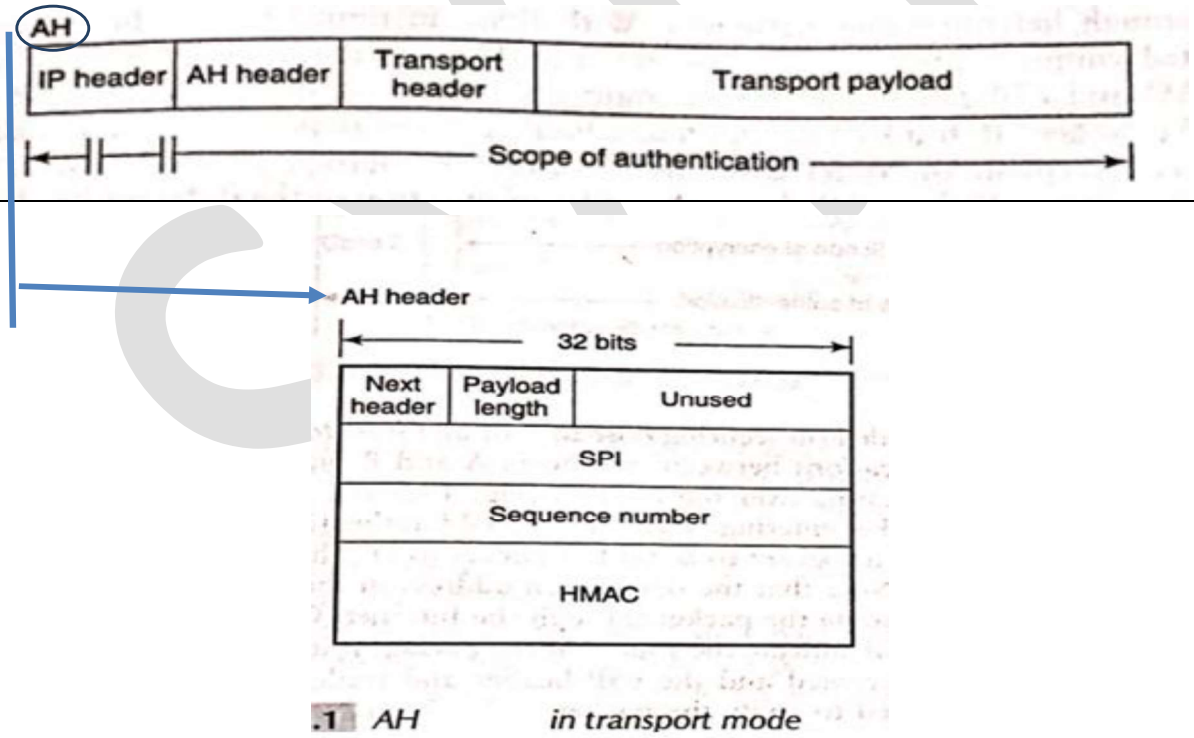
- Each node has a database of SAs for all connections originating from or terminating at it. This database is referred to as the ***SA Database (SADB).***
- Finally, it should be noted that two communicating parties, A and B, establish two SAs — one for communications from A to B and another from B to A.

4.2.2 IPSec Protocols: AH and ESP

- IPSec includes two protocols —
  - ✓ **AH (Authentication Header)**
  - ✓ **ESP (Encapsulating Security Payload).**
- The main ***difference***between AH and ESP is that **AH has no provision** for confidentiality.
- ESP **provides confidentiality** as an option.
- These protocols can be used in either transport or tunnel mode.
- Below Figure shows the headers introduced by AH and ESP and their coverage of authentication and encryption.

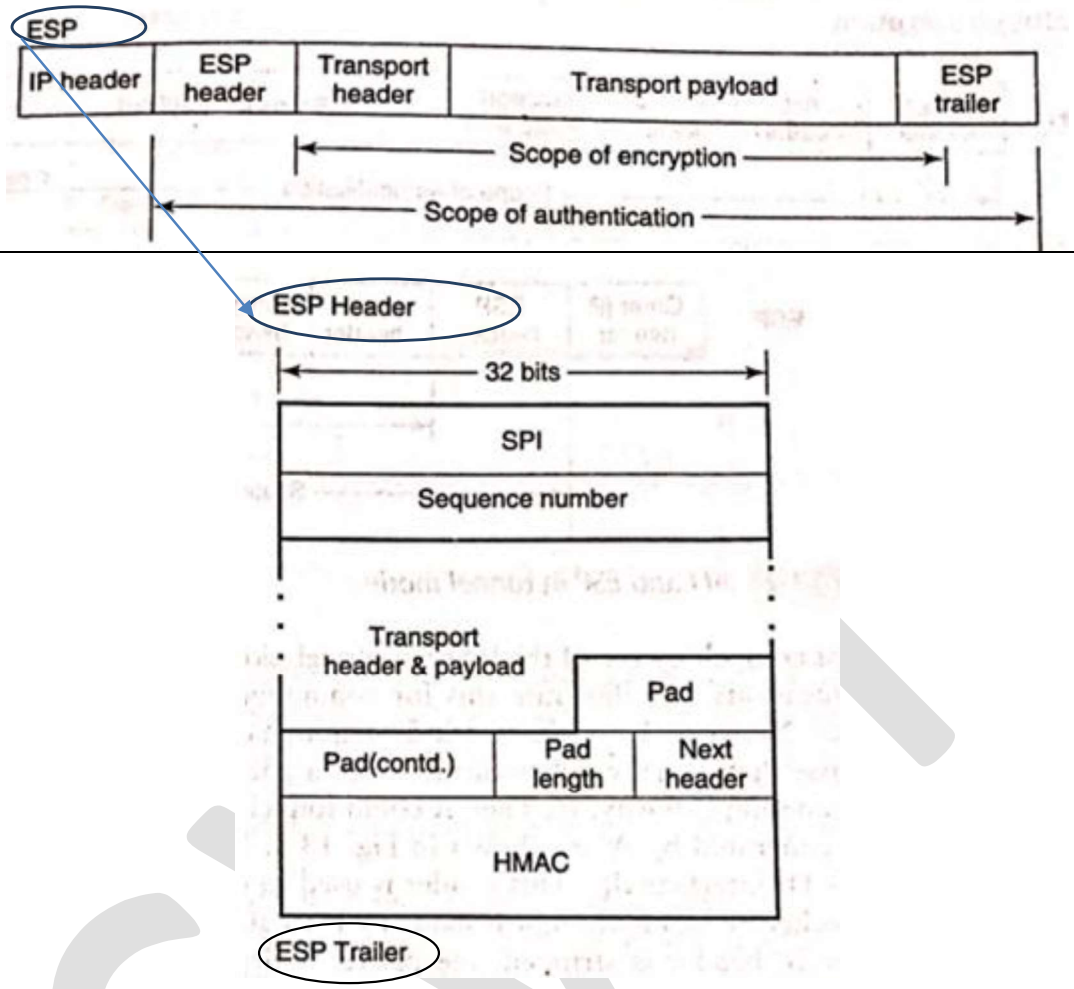
1. Transport mode

AH in transport mode



.1 AH in transport mode

ESP in transport mode



ESP in transport mode

- This mode of operation is called ***transport mode*** because it protects the transport header and the transport payload
- The IPSec header(AH and ESP) is sandwiched between the IP and TCP headers
- ***HMAC***:
  - ✓ Message authentication and integrity areprovided by the use of a ***keyed MAC*** in both cases.
  - ✓ MAC is included in the AH header.
  - ✓ With AH, the integrity check is computed on parts of the IP header such as the source and destination IP addresses.
  - ✓ Mutable parts of the header such as the TTL,ToS and header checksum fields are zeroed before computing the HMAC. (These fields are zeroed for the purpose of computing the MAC, not during actual transmission.)
  - ✓ MAC is included in the trailer of an ESP packet.
  - ✓ The MAC field in the IPSec packet is only 96 bit. So, only the 96 most significant bits of the computed MAC are actually transmitted.

- ESP does *not* provide protection to any part of the IP header in transport mode.
- ***Sequence number:***
- All IPSec headers (both AH and ESP in both modes) have 32-bit fields each for the SPI and a ***packet sequence number*** , to protect against replay attacks.
- ***Padding:***
- Padding is added so that the length of the encrypted payload is a multiple of block size (as required by most encryption algorithms).
- Padding also helps in hiding the actual length of the data when ESP is used with the encryption option turned on.

4.2.2 Tunnel versus Transport Mode

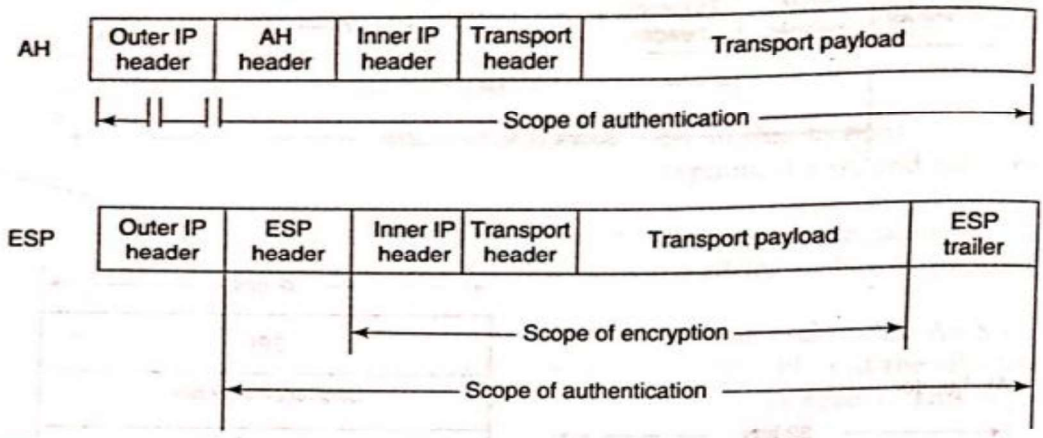


Figure 13.2 *AH and ESP in tunnel mode*

- To protect the entireIP header, IPSec has an option called ***tunnel mode.***
- Both, AH and ESP can employ tunnel mode.
- With encryption turned on, ESP in tunnel mode encrypts the "inner" IP header thus providing limited ***traffic flow confidentiality***. Because the inner, IP header is encrypted, an "outer IP header" is used for routing.
- Figure 13.2 shows the order of insertion of the different headers and the scope of authentication/encryption.
- The most compelling use of the IPSec in tunnel mode is in securing ***host-to-host and host-to-gateway communications.***

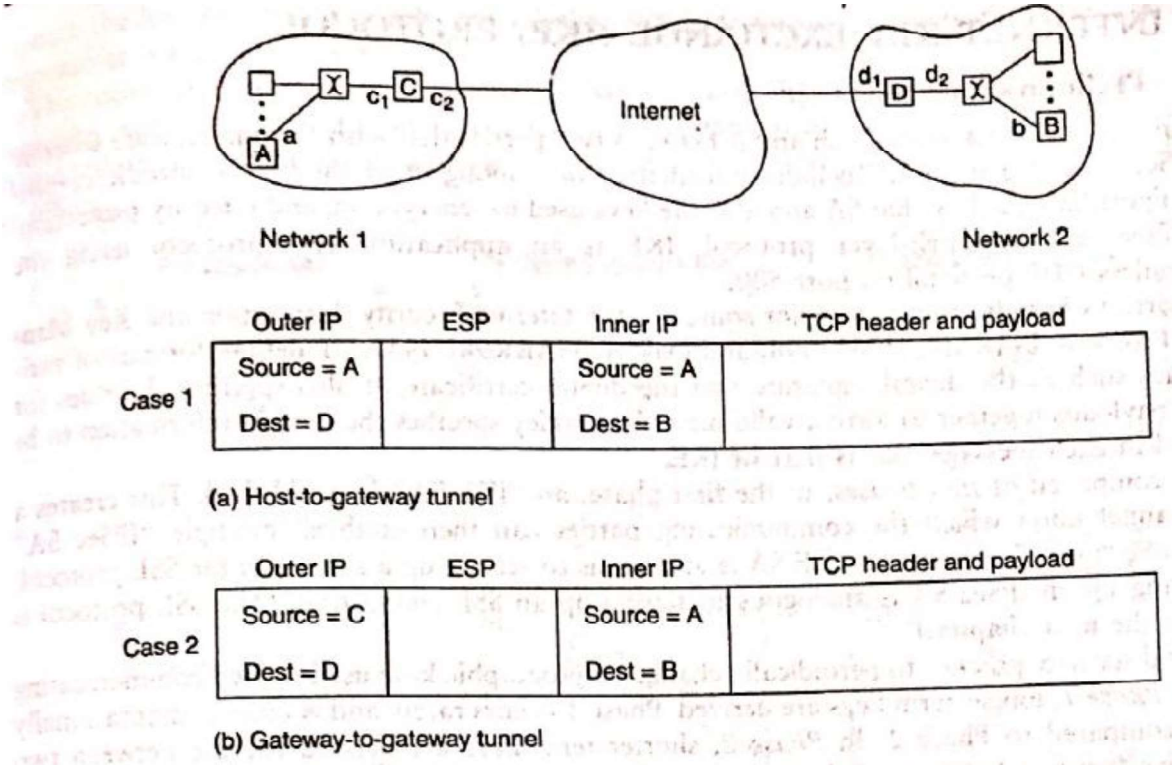


Figure 13.3 *Tunnel mode header contents*

- Consider the communications between two hosts A and B on different networks,
- Network 1 and Network 2, communicating over the Internet [Fig. 13.3(a)].
- Suppose that security policy dictates that a packet entering Network 2 should authenticate itself to the incoming gateway, D. Then A could tunnel its packet to B inside a packet to D. The IP/IPSec headers generated by A are shown in Fig. 13.3.
- The destination address in the outer IP header is D (interface d1). This header is used to route the packet through the Internet.
- On receipt of the packet by D, an attempt is made by D to authenticate the source of the packet. If it succeeds, the outer IP header is stripped, the packet is decrypted and the ESP header and trailer are also stripped.
- The inner IP header of the packet is used to route the packet to B within Network 2.
- In the above example, it is assumed that Host A is IPSec-enabled.

Gateway to gateway tunnel

- The gateway, C, is responsible for providing authentication and encryption service to all outgoing traffic destined to hosts in Network 2.
- In this case, C would prepend an ESP header and another IP header to A's packet for communication through a secure tunnel between C and D [see Fig. 13.3(b)].
- On receipt of the packet, D would strip off the outer IP header, authenticate the source of the packet, decrypt its contents, strip off the ESP header and trailer and route the packet to B.

4.3 INTERNET KEY EXCHANGE PROTOCOL

- The main goal of IKE is to ***establish an SA*** between two parties that wish to communicate securely using IPSec.
- IKE is comprised of ***two phases***.
- In the first phase, an ***"IKE SA"*** is established. This creates a***secure***channel upon which the communicating parties can then establish multiple "IPSec SA' instances over time.
- Setting up an IKE SA is similar to setting up a *session* in the SSL protocol, while setting up an IPSec SA is analogous to setting up an SSL *connection.*
- It is good security practice to periodically change cryptographic keys used by two communicating parties.
- In *Phase 1,* longer term keys are derived.
- Phase 1 occurs rarely and is more computationallyintensive compared to Phase 2.
- In *Phase 2,* shorter term keys are derived for use between two parties. This key is a function of the long term keys computed in Phase 1 together with noncesexchanged in Phase 2.

4.3.1 IPSec Cookies

- To prevent DoS attacks, IKE makes extensive use of ***cookies.***
- ***One cookie*** is created by the initiator, A, and another by the responder, B.
- ***Phase 1*** of IKE uses Diffie—Hellman key exchange, which involves the computationally intense modular exponentiation operation.
- IKE mandates that B should compute a 64-bit integer called a *cookie.*
- Cookie value computed by A : This is a hash function of many variables including the *IP address of* A, an *temporary secret* known only to B and possibly the *time.*
- A is required to send this cookie to B in all subsequent messages.
- On receipt of a message from A, B will check to see whether the cookie corresponds to A's IP address. If the check fails, B will abort session establishment .

4.3.3 IKE Phase I

- The following are accomplished in IKE Phase 1:

✓ The authentication method, encryption, and hash algorithms together with the Diffie—Hellman group to be used are negotiated.

✓ Both parties authenticate themselves to each other.

✓ Key_S, KEYa and KEYe, are computed.

✓ These keys are used for message integrity ,encryption, respectively in both, Phases 1 and 2.

✓ Cookies are created at the start of Phase 1 and serve the purpose of an IKE connection.

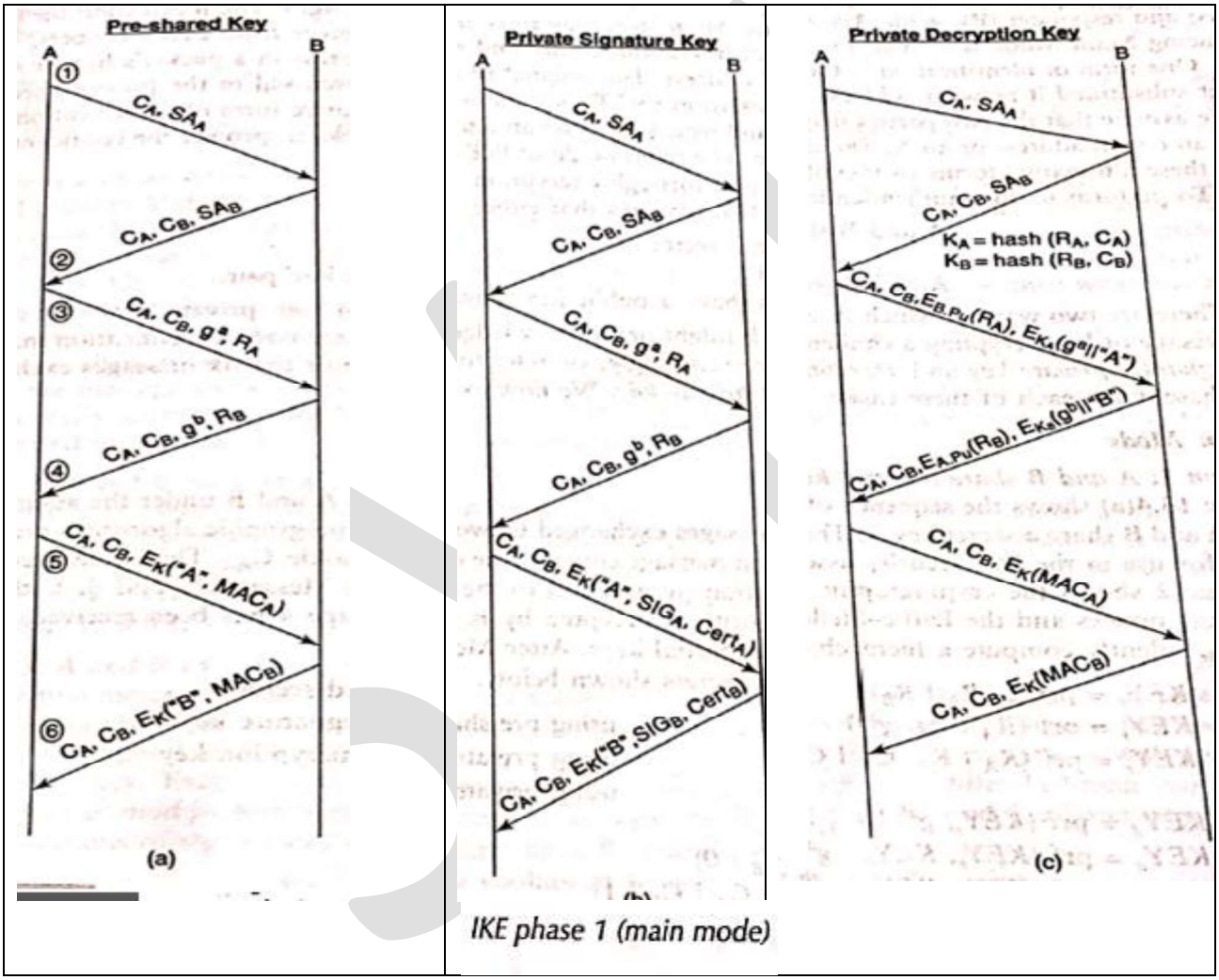
✓ Phase 1 uses one of two modes.

✓ Main Mode involves a ***total of six messages*** between ***initiate (A) and responder (B),***

✓ Aggressive Mode uses only ***three messages.***

- ✓ The motivation for introducing Main Mode is to hide the identities of the sender and receiver from eavesdroppers.
- ✓ The six messages exchanged in Phase 1 for each of these cases.
- ✓ ***Main Mode***
- ✓ IKE derives a hierarchy of keys —**KEYr, KEYd, , KEYa, and KEYe.**
- ✓ The subscripts **r, d, a, and e** denote ***'root', 'derived', 'authentication', and 'encryption',*** respectively.

IKE PHASE 1 main mode



IKE phase 1 (main mode)

$KEY_r = \text{prf}(s, R_A \mid R_B)$ using pre-shared secret
$KEY_r = \text{prf}(R_A \mid R_B, g^{ab})$ using private signature key
$KEY_r = \text{prf}(R_A \mid R_B, C_A \mid C_B)$ using private encryption key

$KEY_d = \text{prf}(KEY_r, g^{ab} \mid C_A \mid C_B \mid 0)$
$KEY_a = \text{prf}(KEY_r, KEY_d \mid g^{ab} \mid C_A \mid C_B \mid 1)$
$KEY_e = \text{prf}(KEY_r, KEY_a \mid g^{ab} \mid C_A \mid C_B \mid 2)$

$MAC_A = \text{prf}(KEY_a, g^a \mid g^b \mid C_A \mid C_B \mid SA_A \mid \text{"A"})$
$MAC_B = \text{prf}(KEY_a, g^b \mid g^a \mid C_B \mid C_A \mid SA_A \mid \text{"B"})$

Option 1 :

- ✓ A and B share a secret key Figure 13.4(a) shows the sequence of messages exchanged between A and B under the assumption that A and B share a secret key, s.
- ✓ The first message contains the cryptographic algorithms proposed by A for use in the IKE security association (in addition to the cookie CA). This is denoted $\mathbf{SA_A}$.
- ✓ Message 2 shows the cryptographic algorithms accepted by B.
- ✓ In Messages 3 and 4, both sides exchange nonces and the Diffie—Hellman partial keys.
- ✓ After Message 4 has been received, A and B independently compute a hierarchy of secrets such as keyr,keya,keye.

Option 2 :A and B each have private signing keys

- ✓ The sequence of messages exchanged between A and B [Fig. 13.4(b) is very similar to that in the shared key case.
- ✓ The main difference is that authentication and integrity protection of messages is effected by digital signatures on MAC_A and MAC_B using their private keys.
- ✓ Also, A and B dispatch, their signing key certificates in Messages 5 and 6 so the other party can perform signature verification [Fig. 13.4(b)].

Option 3 :A and B each have private decryption keys

- ✓ The first two messages used with this option [Fig. 13.4(c)1 are as in the two previous cases.
- ✓ The main difference between this option and the previous ones is that both sides exchange their identities earlier in Messages 3 and 4.
- ✓ Each side generates a nonce (RA or RB) and encrypts it with the other side's public key.
- ✓ Each side encrypts its identity together with its Diffie—Hellman partial key (g^a mod p or ^{gb}mod p) with temporary keys,.K_A and K_B in Messages 3 and 4, respectively.
- ✓ K_A and K_B are functions of nonces nd cookies as below
- ✓ KA = hash (RA, CA)
- ✓ KB = hash,(RB, CB)
- ✓ In Messages 5 and 6, each side transmits a MAC.
- ✓ The MAC key is KEY_awhich,,in turn, is a function of the two nonces, RA and RB.
- ✓ If A or B were unable to decrypt (with their private key)the nonce generated by the other side, then they wouldnot be able to compute the correct MAC .

IKE PHASE 1 Aggressive mode

Aggressive Mode

,The aggressive mode involves ***only three*** messages.

- As shown in Fig. 13.5, the identities of A and B are sent in the clear in messages 1 and 2.
- Another aspect of IKE Phase 1 in aggressive mode is that the Diffie—Hellman group is used and the group parameters are decided by A.
- A computes its partial key and sends it to B *in* Message 1.
- *B* has no choice but to quietly accept the group chosen by A.

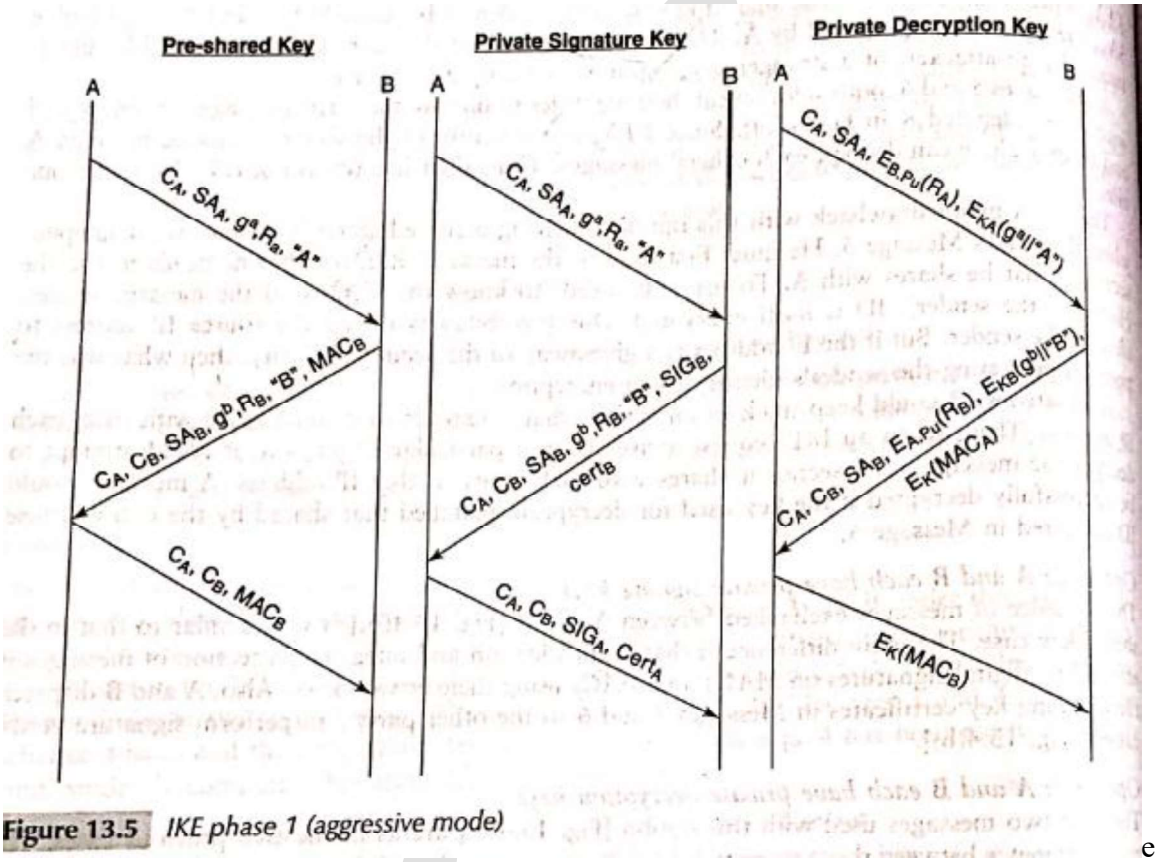


Figure 13.5 *IKE phase 1 (aggressive mode)*

e

IKE phase 2

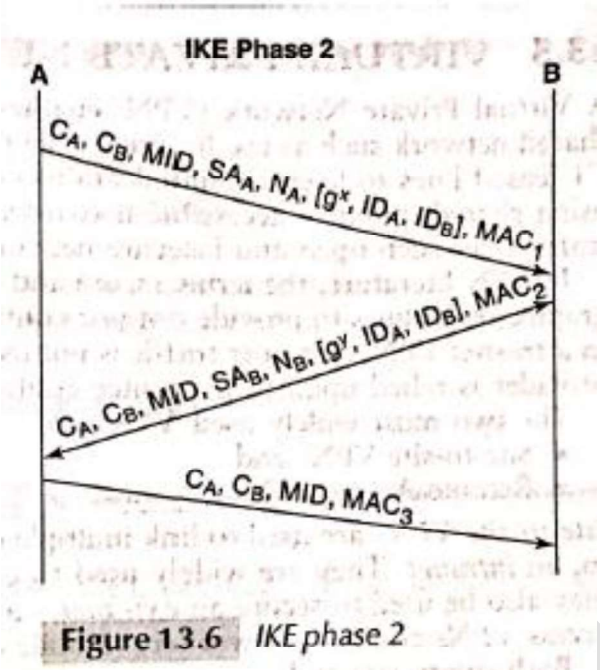


Figure 13.6 *IKE phase 2*

IKE Phase 2

- IKE phase 2 to be used for authentication (and encryption) as part of the IPSec SA.
- Two parties participate in an IKE Phase 2 exchange in order to establish a ***new IPSec SA.***
- Either party can initiate this phase in which the cipher suite and the keys that comprise the ***new IPSec SA*** are agreed upon.
- Above Figure shows the three messages exchanged in ***"Quick Mode."***
- All messages, encrypted using the secret, KEY_e, computed in theprevious phase .
- Message integrity and data source authentication is provided by using an HMAC.
- The key for the HMAC is KEY_a, also computed in Phase 1.
- A 32-Bit "Message ID", MID, is used to distinguish this phase 2 session from, possibly, others that may be set up concurrently within the same IKE SA.
- The MID together with the two cookies created in Phase 1, CA and CB, are dispatched as part of each of the three messages.
- Both sides send their proposals for a suite of cryptographic algorithms to be used in the IPSec SA.
- These are denoted $\mathbf{SA_A}$ **and** $\mathbf{SA_B}$ in Fig. 13.6.
- To guarantee freshness, both sides also generate and transmit nonces, NA and NB.
- In addition to the agreement on a cryptographic suite, the purpose of this phase is to agree on the secrets
- These secrets are computedsimultaneously by both sides and are a function of KEY_d computed in Phase 1 and the nonces N_A ***and*** N_B exchanged in this phase.
- To integrity protect the messages each side computes a MAC on their messages. MACS includes integrity protection for both nonces, N_A ***and*** N_B.

- The Diffie—Hellman partial secrets are $g^x \bmod p$ and $g^y \bmod p$
- The Diffie—Hellman secret key, $g^{xy} \bmod p$ is also used as an additional input in computing the necessary keys for the IPSec SA
- The IPsec SA set up in Phase 2 includes the mutually agreed upon cryptographic suite and secret keys for authentication and/or encryption.
- Both sides will then create an entry, for, the new SA in their database of SAs.

4.4 SECURITY POLICY AND IPSEC

- ***A Security Policy Database (SPD)*** is used to determine whether a packet sent or received should pass through security, bypass it, or simply be dropped. Such a decision is made based on fields in the IP and transport headers.
- These fields, called ***selectors,*** include the ***destination IP address, the type of transport layer protocol (whether TCP or UDP) and type of application (indicated by transport Selectors are used to index into the SPD.***
- The output of this lookup indicates whether security should be applied.
- If so, and if the packet is part of the IP traffic that already has an existing SA, protocol port number). then the SPD returns a pointer to that SA.
- If an SA does not exist or has expired, the IKE protocol is used to establish an SA between the sender and receiver.

4.5 Virtual private networks

- ✓ A virtual private network (VPN) enables organization to communicate securely over a public, shared network such as internet.
- ✓ A secure VPN uses cryptographic techniques to provide not just confidentiality but also authentication and message integrity.
- ✓ In a trusted VPN, customer traffic is not usually encrypted ,service provider has to guarantee confidentiality of traffic.
- ✓ The two most widely used VPNs are:
- ✓ ***Site to site VPN***: are used to link multiple offices of an organization ,known as intranet.
- ✓ Site to site VPN may also be used to secure an extranet-a network connecting multiple business partners.
- ✓ ***Remote access VPNs*** connect teleworkers to their offices.

Chapter 5
Security at the Transport Layer

3.12 INTRODUCTION

- Developed by Netscape in 1994, the **Secure Sockets Layer (SSL)** protocol has emerged as the principal means of ***securing communications between an Internet client (such as a browser) and a server.***
- It was standardized by IETF in 1999 and called **Transport Layer Security (TLS).**

SSL(Secure Sockets Layer)

- SSL is sandwiched between ***TCP (it only runs over TCP) and an application layer protocol.***
- It is application protocol independent.
- Protocols such as HTTP, FTP, SMTP, IMAP, and POP can all be run over SSL.
- Application protocols **secured by SSL are usually suffixed by an "S"** and run on different port numbers.
- For example, ***HTTP runs on port 80 but HTTPS runs on port 443***.
- FTP runs on ***port 21 but FTPS runs on port 990.***
- SSL is comprised of two main protocols (see Fig. 14.1)

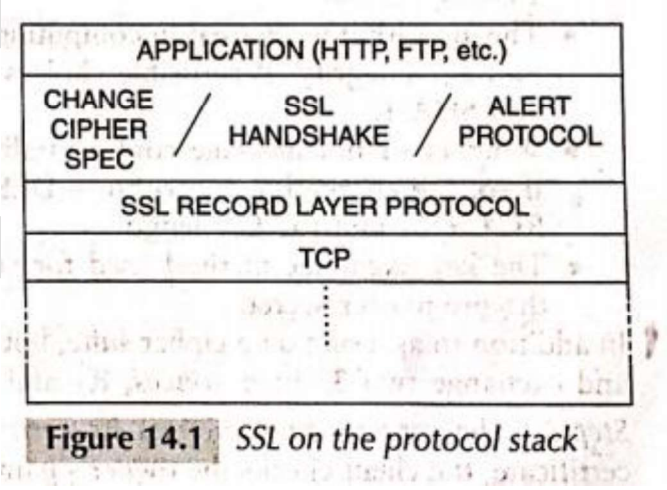


Figure 14.1 *SSL on the protocol stack*

1. **The Handshake Protocol**
2. **The Record Layer Protocol**

- The SSL handshake protocol is used to negotiate the set of algorithms to be used for securing the communication link.
- Server authentication in SSL is mandatory and performed as part of the **handshake**.
- The hand-shake protocol is also responsible for deriving keys , for encryption and MAC computation

- The actual job of providing ***message authentication + integrity checking and encryption*** is performed by the **SSL record layer protocol.**
- It sits just below the handshake protocol and protects each message exchanged by the two communicating parties.
- The record layer protocol also detects ***replayed, re-ordered, and duplicate packets.***

3.13 SSL HANDSHAKE PROTOCOL

3.13.1 Steps in the Handshake

- The client initiates a handshake with the server to either
  (a) Start a new session or
  (b) resume an existing session or
  (c) Establish a new connection within an existing session.
- The main steps in the SSL handshake for establishing a new session are as follows:
  (1) Agreement on a ***common cipher suite***to be used in the new session
  (2) Receipt and validation of the ***server certificate***by the client
  (3) Communication of a ***"pre-master secret"***and computation of derived secrets
  (4) ***Integrity verification***of handshake messages and ***server authentication***
- These steps are realized by the sequence of messages shown in below figure
- The steps are:

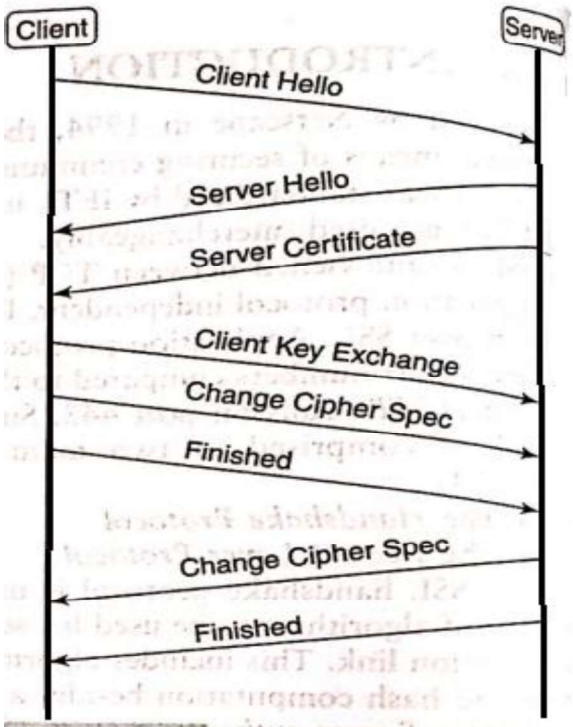


Figure 14.2 *SSL handshake*

- ***Step 1:*****Two messages are communicated in this step —*Client Hello* and *Server Hello.***

The following decisions are taken here:

- Should a new session be established or should an existing one be re-used?
- For a new session the session ID field in the Client Hello message is 0; else the field is set to the ID of the session to be re-used,
- Thesession ID field in the Server Hello message is the ID of the new session to be established or the ID of an existing session.
- The algorithm to be used in computing the MAC for message integrityinclude MD5 and SHA-1.
- The key exchange method used for communicating the pre-master secret.
- In addition to agreeing on a cipher suite, both sides choose and exchange two 32-byte *nonces,* **RA** and **RB,** in this step.

- ***Step* 2. The server communicates its *certificate* to the client (see Fig. 14.2).**
- On receipt of the certificate, the client checks the owner's name/URL and validity period.
- It also verifies the signature of the CA on the certificate.
- Successful verification of these fields does not guarantee the authenticity of the sender
- Authentication of the server only occurs at the end of Step 4,

Step 3.

- The client chooses a *pre-master secret* — a 48-byte random number.
- The pre-master secret is encrypted with the server's public key and sent to the server in the Client key exchange messages.
- Thereafter, both client and server compute the master secret. This is an HMC style function, f , of the pre master secret,the two nounces exchanged in step 1 and some pre defined constants.
- The computation uses a standard cryptographic hash function such as the SHA-1 or the MDS.

$$\textbf{Master\_Secret} = f(\textbf{Pre-Master\_Secret } R_A, R_B, \textbf{constants})$$

- Finally six secrets are derived using HMAC-style functions of the master secret, the two nonces, anddifferent pre-defined constants

$$\textbf{Derived\_Secret\_i} = f(\textbf{Master\_Secret}, R_A, R_B, \textbf{constants}), \quad 1<i<6$$

- The six derived secrets are:
  - ✓ Initialization vector for encrypting messages from client to server
  - ✓ Initialization vector for encrypting messages from server to client
  - ✓ Secret key for encrypting messages from client to server
  - ✓ Secret key for encrypting messages from server to client
  - ✓ Secret for computing keyed hash on messages from client to server(Client MAC Secret)
  - ✓ Secret for computing keyed hash on messages from server to client (Server MAC Secret)

Step 4: **This step involves the exchange of two messages in each direction.**

- The first of these is the **"Change_Cipher_Spec"** message (Fig. 14.2).
- The party that sends this message signals that from now on the cipher suite and the keys computed will be used.
- The second message in this step is the "Finished" message.
- This message includes a keyed hash on the concatenation of *all*the handshake messages sent in the preceding steps + a pre-defined constant.
- The keyed hash serves as an ***integrity check***on the previous handshake messages.
- After the server receives the "Change_Cipher_Spec" and "Finished" messages from the client, it verifies the computation of the keyed hash.

- It then computes its own keyed hash that covers the previous handshake messages + a pre-defined constant, which is distinct from the one used by the client.
- The client receives the keyed hash and verifies it. Only at this point is the server authenticated to the client.
- On the other hand, client authentication as part of the SSL handshake is optional.

3.13.2 Key Design Ideas

Key Exchange Methods

- In Step 2, the server dispatches its certificate so the client can use the public key contained in the certificate to encrypt the pre-master secret.
- In some cases, however, the server's certificate may be a "signature-only certificate."
- This means that the public key in the certificate and the corresponding private key may only be used exclusively for signature generation/verification, not for encryption.
- In that case, SSL permits the server to create a **temporary public key/private key pair**. The public key (including modulus) are signed by the server using the private key corresponding to the public key in the **signature-only certificate.**
- The signed public key and certificate are communicated by the server to the client.
- The client verifies the signature on the public key and then uses it to encrypt the pre-master secret.
- SSL offers a rich set of options for key exchange.
- Such as RSA-based key exchange methods, Diffie—Hellman key exchange may be used.

Server Authentication

- The MAC computed by both parties and sent in step 4 is used as an ***integrity check***on the previous handshake messages.
- All the handshake messages are sent in the clear (except for encryption of the pre-master secret).
- It is possible for an attacker to alter one or more of the handshake messages.
- For example, he may replace**128-bit DES by a 56-bit DES.**
- This may induce both parties to use a weaker cipher, which can be compromised by the attacker.
- The MAC detects any modification in the handshake messages.
- The hash computed by the server and verified by the client uses the *server* MAC s*ecret*,
- It is a function of the master secret which in turn is a function of the pre-master secret.
- Recall that the pre-master secret is chosen by the client and encrypted with the server's public key so that the server alone can read it. So, nobody but the server and client could compute the six secrets.
- Only after the client receives and verifies the keyed hash from the server, is it convinced that it is talking to the authentic server.

Sessions and Connections

- It is good security practice to change keys during a long-lasting session.
- SSL has provision for changing keys by creating new connections within an existing session.
- In creating a new connection, the pre-master secret which is part of the existing session state is *not* chosen anew.
- Instead, a new master secret is computed as a function of the *existing pre_ master secret* and two *fresh notices* contributed by the client and server.
- The *session state* includes the *pre-master secret,* the negotiated *cipher suite* and, of course, the *session* ID.
- The *state of a connection* includes the two *nonces,* the *master secret,* the six *derived secrets,* and two message *sequence*(one for each direction of message transfer).

3.14 SSL RECORD LAYER PROTOCOL

- The SSL record layer protocol is used to securely transmit data using the negotiated cipher suite andthe keys derived during the SSL handshake.
- Its main tasks are computation of a per-message MAC and encryption.
- If the data to be transmitted is very large, it performs fragmentation.
- Each fragment is 16 kb or less.
- When a connection is established, both sides initialize a sequence counter to zero.
- The counter is incremented for each packet sent.
- The sequence number itself is not sent. However, it is used in the computation of the MAC (at the sender) and in its verification (at the receiver).
- The MAC is computed on the concatenation of the 64-bit sequence number and the compressed fragment (if compressionis used).
- The next step after computing the MAC is encryption.
- If the combined size of the data fragment and MAC is not a multiple of block size,a pad is appended.
- The data fragment, MAC, and pad (if any) are then encrypted, prepended with a header, and passed on to the TCP layer for further processing.
- The SSL record layer protocol header :there is a 1-byte Content Type field, which identifies the higher layer protocol used to process thefragment.
- Two bytes are used to specify the Versionnumber.
- Finally, the Length field indicates the fragment size in bytes.

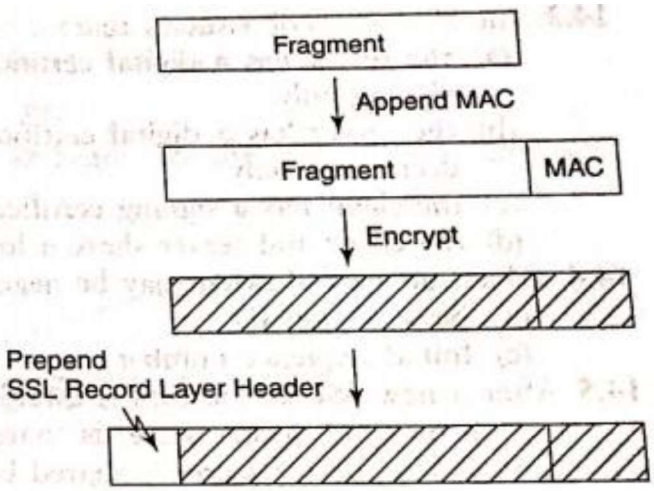


Figure 14.3 *Function of the SSL record layer protocol*

3.15 OpenSSL

- **OpenSSL** is open source software that implements the SSL/TLS protocol.
- It is comprised of a number of libraries that implement various cryptographic algorithms.
- It provides extensive support for communicating and validating digital certificates.
- OpenSSL is based on the **SSLeay** library developed by Eric A. Young and Tim J. Hudson.
- OpenSSL enhances the productivity of application developers by providing a rich set of APIs that handle diverse aspects of SSL-enabled communication from connection set-up and tear-down to certificate storage, management, and verification.
- The developers can rely on the OpenSSL APIs to implement the required security.

MODULE 4

Chapters	
1.IEEE 802.11 wireless LAN security	2.viruses,worms and other malware
3.firewalls	4.Intrusion Prevention and detection
5.Web Services Security	

1. IEEE 802.11 Wireless LAN Security

- Wireless networks present formidable challenges in the area of security.
- The open nature of such networks makes it relatively easy to sniff packets or even modify and inject malicious packets into the network.
- The ease with which such attacks are launched necessitates careful design and deployment of security protocols for wireless networks.

1.1 BACKGROUND

Wired network

- In many organizations, the wired network is an Ethernet LAN with an existing security infrastructure that includes an authentication server (AS).
- ***AAA (Authentication/Authorization/Accounting)*** functionality is often provided by a RADIUS (Remote Authentication Dial in User Service) server.

WLANs(wireless LANs)

- There are ***two principal types of WLANs —***
    1. ***Ad hoc networks: where*** stations (possibly mobile) communicate directly with each other.
    2. ***Infrastructure WLANs:*** which use an access point (AP) as shown in below figure.

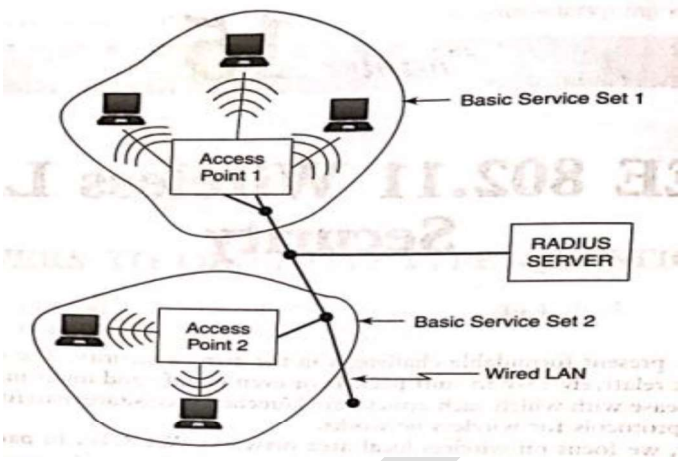


Figure: Infrastructure wireless LAN

Infrastructure WLANs :

- A station first, sends a frame to an AP and the AP then delivers it to its final destination.
- The destination may be another wireless station or it may be a station on the wired network that the AP is connected to.
- The **AP** thus serves as a **bridge** between the WLAN and the existing wired network.
- The challenge then is to develop protocols that seamlessly integrate the WLAN with the security infrastructure of the wired network.
- A network of wireless stations associated with an AP is referred to as a ***basic service set.*** Such a network may be adequate for a home or small enterprise.
- The union of the basic service sets comprises an ***extended service set (ESS).***
- Each station and AP in the ESS is uniquely identified by a MAC address — ***a 48-bit quantity.***
- Each AP is also identified by an ***SSID (service set ID),*** which is a character string of length at most ***32 characters.***
- A wireless station, on power-up, needs to first discover an AP within its range.

➢ This can be done by monitoring the wireless medium for a special kind of frame called a ***Beacon,*** which is periodically **broadcast by the AP.** ➢ The Beacon usually contains the ***SSID*** of the broadcasting AP.	➢ Alternatively, a station may send a ***Probe Request frame,*** which probes for APs within its range. ➢ An AP, on hearing such a request, responds with a Probe Response frame. ➢ Like the Beacon, the ***Probe Response frame*** contains the SSID of the AP and also information about its capabilities, supported data rates, etc.

- A station that wishes to associate with an AP sends it an ***Associate Request frame.***

- The AP replies with an ***Associate Response frame*** if it accepts the request for associating with it.

1. The earliest protocol that incorporated security in WiFi was **WEP.**
   - ✓ Designed to provide authentication/access control, data integrity, and confidentiality, it failed on all three counts.
2. **WiFi Protected Access (WPA)**
   - ✓ WPA was intended to fix the shortcomings of WEP without requiring new wireless network cards.
   - ✓ But WPA is not perfect — it too is susceptible to attacks on its cryptographic algorithms.
3. **WPA2**
   - ✓ All the deficiencies in WEP have been addressed in the IEEE 802.11i (implemented in WPA2) .

1.2 AUTHENTICATION

1.2.1 Pre-WEP Authentication

- Drawbacks:

1. An attacker could easily sniff the value of SSID from frames such as the beacon or probe response and then use it for authentication.
2. Another approach was to restrict admission to the WLAN by MAC address.
   - ✓ The AP would maintain a list of MAC addresses (access control list) of stations permitted to join the WLAN.
   - ✓ valid MAC addresses could be obtained by sniffing the wireless medium.
   - ✓ The attacker could then modify his network card to spoof a valid MAC address. So, neither of these approaches was truly secure.

1.2.2 Authentication in WEP

- In WEP, the station authenticates itself to the AP using a challenge—response protocol .
- Basically, the AP generates a challenge (nonce) and sends it to the station.
- The station encrypts the challenge and sends it to the AP.
- The stream cipher, RC4, is used for encryption.
- **Response From Station**: the station computes a keystream, which is a function of a 40-bit shared secret, S, and a 24-bit Initialization Vector (IV).
- The challenge is then XORed with the keystream to create the response.

RESPONSE = CHALLENGE (XOR) KEYSTREAM(S, IV)

- The response together with the IV is sent by the station to the AP.
- The shared secret, S, is common to all stations authorized to use the WLAN.

Drawbacks:

- All an attacker needs to do is to monitor a challenge—response pair.
- From this, he can compute the keystream.
- To authenticate himself to the AP, he needs to XOR the challenge from the AP with the computed keystream.
- It may also be possible for an attacker to obtain S itself.
- By eavesdropping on several challenge—response pairs between the AP and various stations, an attacker could launch a **dictionary attack** and eventually obtain S.

1.2.3 Authentication and key agreement in 802.11

- 802.11i uses IEEE 802.1x — a protocol that supports authentication at the link layer.
- Three entities are involved:
  1. **Supplicant (the wireless station)**
  2. **Authenticator (the AP in our case)**
  3. **Authentication server**

- Different authentication mechanisms and message types are defined by the ***Extensible authenticationProtocol (EAP)*** standardized by Internet Engineering Task Force (IETF).
- EAP is not really an authentication protocol but rather a ***framework*** upon which various authentication protocolscan be supported.
- EAP exchanges are mostly comprised of **requests and responses.**
- For example one party requests the ID of another party.
- The latter responds with its user_name or e-mail address.
- EAP also defines messages that may contain challenges and responses used in authentication protocols.
- The AP broadcasts its security capabilities in the Beacon or Probe Response frames.
- The station uses the Associate Request frame to communicate its security capabilities.
- 802.11i authentication takes place after the station associates with an AP.

IEEE 802.11i

- The generic authentication messages in IEEE 802.11i are shown in Fig. 15.2.
- The protocol used between the ***station and the AP is EAP*** but that used between the AP and the authentication server depends upon the specifics.
- For example, the authentication server is often a RADIUS server which uses its own message types and formats. (RADIUS stands for Remote Authentication Dial in User Service. It is a client—server protocol used for authentication, authorization, and accounting.)

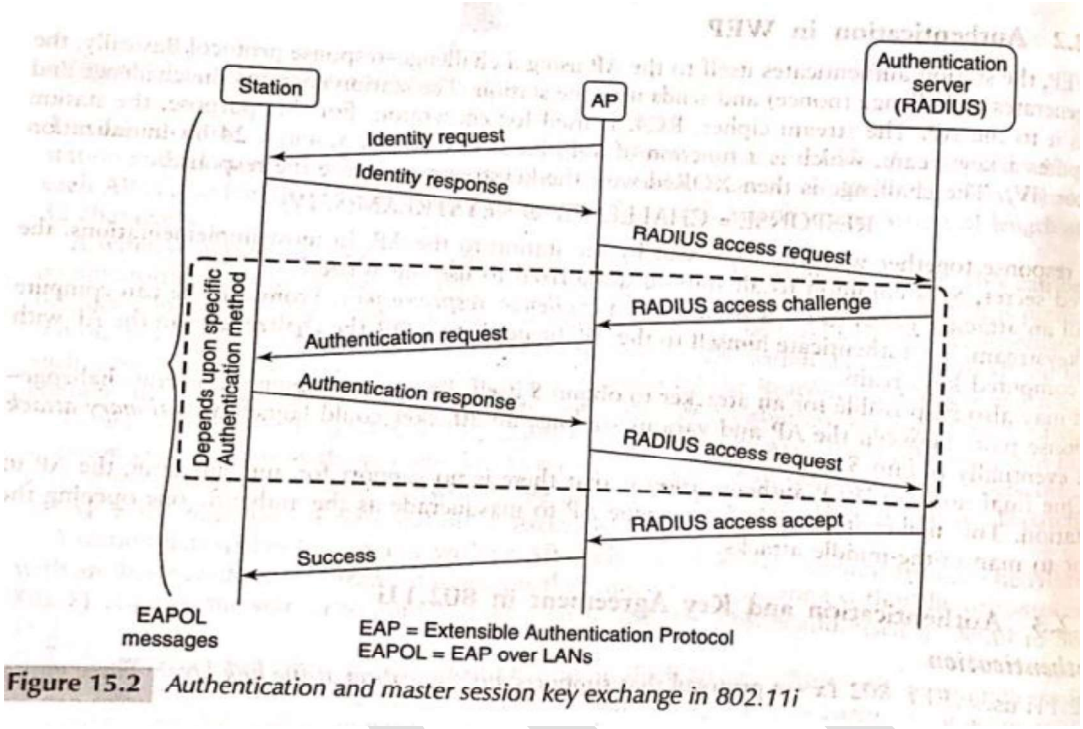


Figure 15.2 *Authentication and master session key exchange in 802.11i*

Figure 15.2 Authentication and master session key exchange in 802.11i

EAP = Extensible Authentication Protocol messages

EAPOL = EAP over LANs

- The main authentication methods supported by EAP include the following:
  1. **EAP-MDS**
  2. **EAP-TLS**
  3. **EAP-TTLS**
  4. **EAP-PEAP**

1. **EAP-MDS**
   - ✓ This is most basic of the EAP authentication methods.
   - ✓ Here, the authentication server challenges the station to transmit the MD5hash of the user's password.
   - ✓ The station prompts the user to type his/her password.
   - ✓ It then computes the hash of the password and sends this across.
   - ✓ This method is insecure since an attacker could eavesdrop on such a message exchange and then replay the hashed password thus impersonating the owner of the password.
   - ✓ Also, this method does not support authentication of the AP to the station.

2. **EAP-TLS**
   - ✓ EAP-TLS is based on the SSL/TLS protocol
   - ✓ most secure and provides mutual authentication and agreement on a ***master session key.***
   - ✓ It requires the AP as well as the user (station) to have digital certificates.
   - ✓ It is relatively straightforward to equip each AP with a digital certificate and a corresponding private key but extending the Via to each user of the WLAN may not be feasible.

3. **EAP-TTLS**
   - ✓ (tunnelled TLS) requires certificates only at the AP end.
   - ✓ TheAP authenticates itself to the station and both sides construct a secure tunnel between themselves.
   - ✓ Over this secure tunnel, the station authenticates itself to the AP.
   - ✓ The station could transmit **attribute-value** pairs such as
     user_name = akshay
     password = 4rP#mNaS&7

4 **Protected EAP (PEAP)**
   - ✓ This was proposed by Microsoft, Cisco, and RSA Security, is very similar to EAP-TTLS.
   - ✓ In PEAP, the secure tunnel is used to start a second EAP exchange where in the station authenticates itself to the authentication server.
   - ✓ The enhanced security offered by EAP-TLS, EAP-TTLS, and PEAP does, however, come at a steep price in performance measured by the message and computational overheads incurred during authentication.

Key Hierarchy

- There are two types of keys used in WLANs.
- The first are ***pairwise keys*** used to protect traffic between a station and an AP.
- The second type of key is the ***group key*** intended to protect broadcast or multicast traffic between an AP and multiple stations.

The hierarchy of 802.11i keys:

- The root of the key hierarchy is the ***Pairwise Master Key*** **(PMK).**
  - ✓ This is obtained in one of two ways
  - ✓ The station and the authentication server may agree on a Master Session Key (MSK) as part a of the authentication procedure.
  - ✓ The authentication server communicates this key to AP
  - ✓ The AP and station then derive the PMK from the MSK.

- An alternative to computing a fresh PMK for each session is the ***Pre-Shared Key, (PSK),*** which is used as the PMK.
- ***Pairwise Transient Key (PTK).***
  - ✓ The 256-bit PMK is used to *derive* a ***384-bit pairwise Transient Key (PTK).***
  - ✓ The PTK is a pseudo random function of the PMK.
  - ✓ PRF(nonce of AP,nonce of station,MAC address of AP.MAC address of station, PMK).
- Three **128 bit** chunks are extracted from the 384 bit PTK for the following purposes:
  1. **A *Temporal Key* (TK) is used for both encryption and integrity protection of data between the AP and the station.**
  2. **A *Key Confirmation Key* (KCK): Integrity protection is supported by a MAC computed as a function of the message and and Associate Request Frames and the KCK.**
  3. **A *Key Encryption Key* (KEK) is used to encrypt the message containing the group key.**

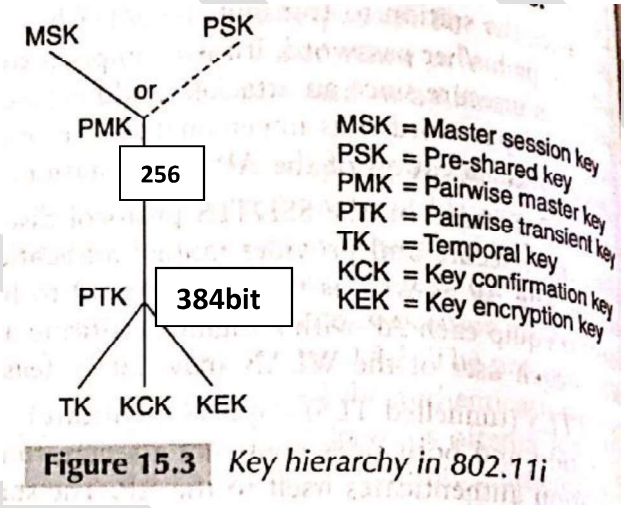


Figure 15.3 *Key hierarchy in 802.11i*

Four-way Handshake

- The main goals of the four-way handshake are to

  (a) Derive the PTK from the PMK,
  (b) Verify the cipher suites communicated in the Beacon and Associate Request Frames.
  (c) Communicate the group keys from the AP to the station.
- Figure 15.4 shows the messages comprising the four-way handshake.

1. Message 1:The AP first sends a nonce, N_A, to the station.
2. Message 2:
   - ✓ The station chooses a nonce, ***N*** station computes the PTK as follows
   - ✓ ***PTK* = prf *(PMK, NA, N_S, MAC_A, MAC_S) ...***
   - ✓ The PTK is a pseudo-random function (prf) of the PMK, the MAC addresses of the station and AP and nonces contributed by the station and the AP.

- ✓ The two nonces help prevent replay attacks.
- ✓ Three 128-bit keys — **TK, KCK, and KEK** are extracted from the 384-bit PTK (Fig. 15.3).
- ✓ The station sends nonce ,cipher suite and uses KCK to compute MIC(message integrity check).

3. Message 3 :
   - ✓ On receiving the message 2,AP computes the PTK from the above expression used by the station .
   - ✓ AP verifies the integrity and source of message 2 using the key KCK.
   - ✓ Message 3 contains group transient key (GTK).this is the key used by the AP ans all staions to integrity protect and multicast and broadcast.
4. Message 4:
   - ✓ This is an acknowledgement from the station that it has received the previous messages without error.
   - ✓ It is a signal to the AP that henceforth all messages will be integrity-protected and encrypted with the TK.

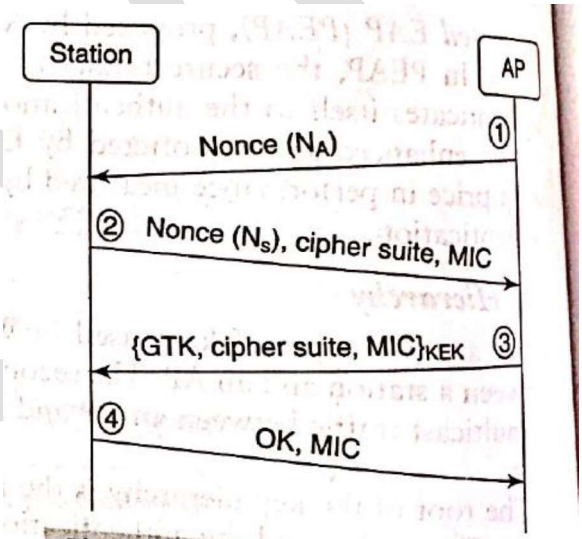


Figure 15.4 *Four–way handshake in 802.11i*

1.3 CONFIDENTIALITY AND INTEGRITY

1.3.1 Data Protection in WEP

- WEP was designed to provide message confidentiality, integrity, and access control but it failed on all three counts.
- In this section, we show how plaintext can be recovered and messages can be modified due to flawed design decisions in WEP.
- There are many lessons to be learned from WEP — the most important being how not to design protocols for security.

WEP Encryption and Integrity Checking

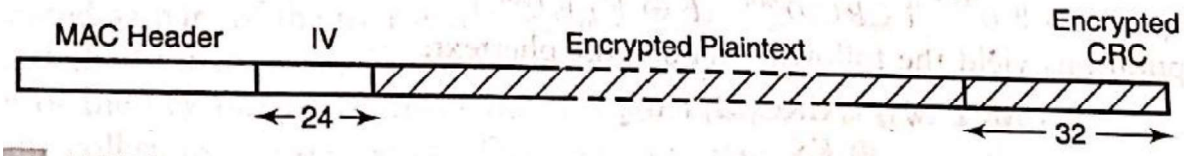
- WEP uses the stream cipher, RC4, for encrypting messages.
- It generates a pseudo-random keystream, KS, which is a function of a static secret shared between the two communicating parties.
- In order to have KS vary from message to message, a random per-message initialization vector, IV, is also used to generate KS.
- Early implementations of WEP used a 40-bit secret, S, concatenated with a 24-bit IV to create, in effect, a "64-bit key."
- KS is xored with the plaintext, P, to obtain the ciphertext, C or

$$C = \mathcal{P} \oplus KS(S, IV)$$

The plaintext p is:

$$\mathcal{P} = C \oplus KS\ (S, IV)$$

-



.5 *WEP frame*

Known plaintext attack

- The first problem with WEP is the possibility of keystream re-use.
- Since the IV is 24 bits in length, there are only 2^{24} distinct keystreams that could be constructed given a secret S.
- Suppose an attacker finds two frames which were encrypted using the same IV.
- Let their ciphertexts be C and C'.
- Let the corresponding plaintexts be P and P'.

$$P \oplus P' = C \oplus C'$$

$$P' = P \oplus C \oplus C'$$

- Thus knowing c,c', and p, we can obtain p' which is called as known plaintext attack.

Message modification

- Consider an attacker who wishes to modify a message sent by a legitimate user.
- Let the sender's plaintext (not including the CRC checksum) be M1 F M2 where M1, F, and M2 are each binary strings.
- The attacker wishes to substitute the substring, F, with another substring, F',
- so that the decrypted message seen by the receiver is M1 F' M2. The attacker does not need to know the values, M1 and M2. However, we assume that he knows F and F'.
- Ideally, the message integrity check should detect any modification to an existing message. Can the attacker modify the message (including checksum) in such a way so that the modification is undetected at the receiver end?
- For the above plaintext, the **ciphertext** computed by the sender is :

$$CT = ((M_1 \; F \; M_2) \parallel CRC(M_1 \; F \; M_2)) \oplus KS$$

The attacker intercepts the ciphertext and performs the following operations:

1. He first constructs the string, $0^{|M_1|} \parallel (F \oplus F') \parallel 0^{|M_2|}$. Here, $0^{|M_1|}$ is a string of $|M_1|$ zeros where $|x|$ is the length of the substring x.
2. He then computes the CRC on this string, $CRC(0^{|M_1|} \parallel (F \oplus F') \parallel 0^{|M_2|})$.
3. He finally XORs the original ciphertext with $0^{|M_1|} \parallel (F \oplus F') \parallel 0^{|M_2|} \parallel CRC(0^{|M_1|} \parallel (F \oplus F') \parallel 0^{|M_2|})$.

The last step follows from the fact that the CRC is a linear operation, i.e.,

$$CRC(m_1 \oplus m_2) = CRC(m_1) \oplus CRC(m_2)$$

The receiver, on decrypting the ciphertext, obtains

$$(M_1 \; F' \; M_2) \parallel CRC(M_1 \; F' \; M_2)$$

- The modified message has a ***valid CRC*** and so passes the integrity check at the receiver.
- Hence, the receiver accepts the message, ***unaware that it has been modified by an attacker.***

1.3.2 Data protection in TKIP and CCMP

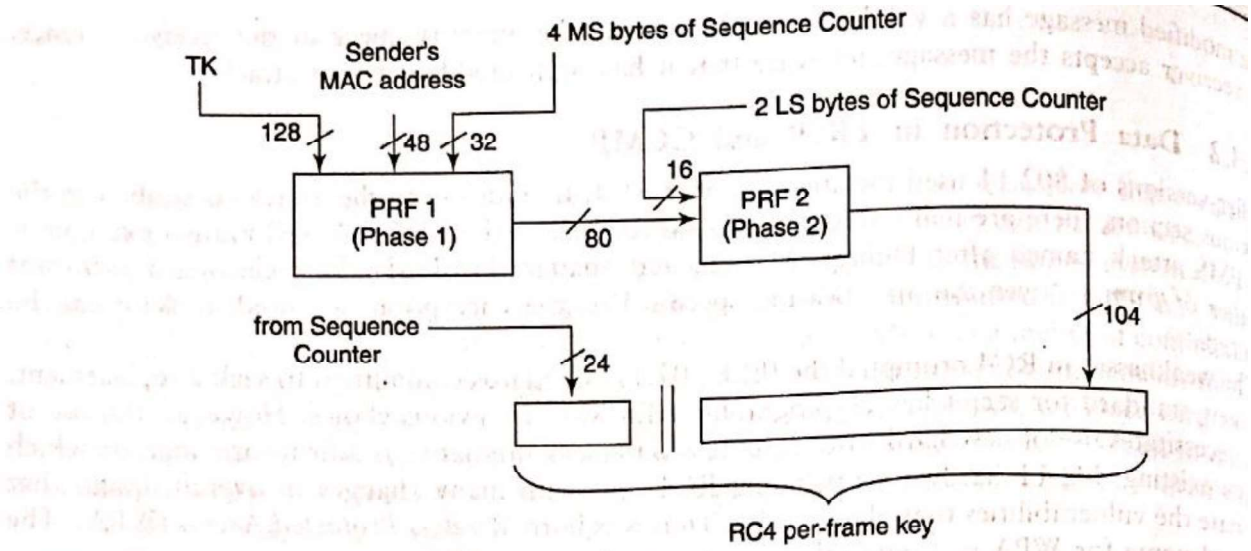


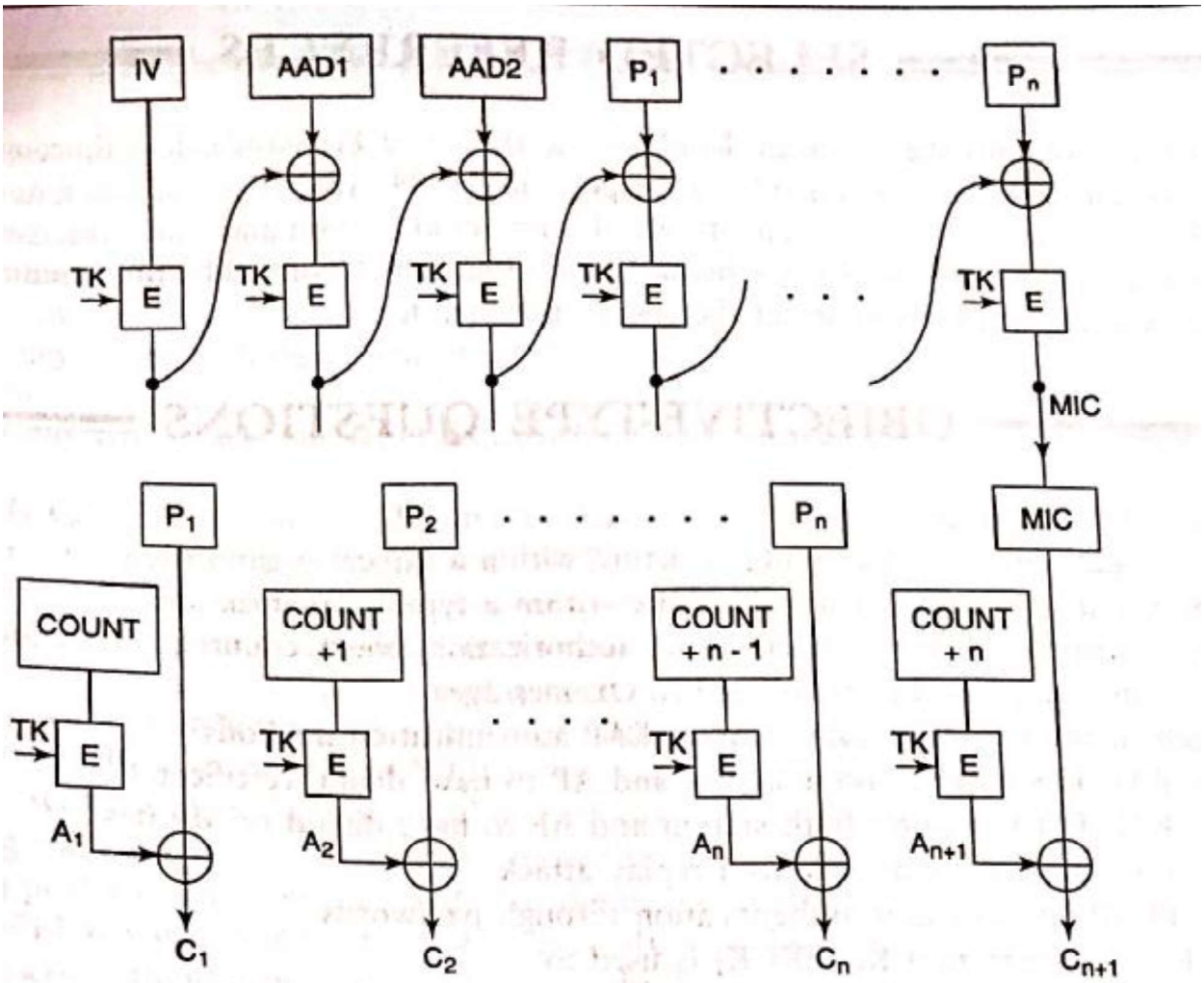
Figure 15.6 *Two-phase key mixing in TKIP*

- The technical name for WPA is *Temporal Key Integrity Protocol* (TKIP).
- By contrast, the encryption key in TKIP is 128 bits.
- TKIP generates a random and different encryption key for each frame sent. It employs a process called ***two-phase key mixing***.
- The inputs to this process are the 128-bit temporal key, TK, computed as part of the four-way handshake ,the sender's MAC address and the four most significant bytes of a 48-bit *frame sequence counter.*
- The randomizing capability of the key mixing function and the large size of the key space virtually guarantee that "keystream collisions" never occur.
- Thus, known plaintext attacks that could be successfully launched on WEP have no chance of success with TKIP.
- The sequence counter is incremented for each frame sent.
- It is also carried in the header of each frame.
- This helps protect the receiver from ***replay attacks.***
- Figure 15.6 shows the two phases used in generating the **RC4 key.**
- Two pseudo-random function (PRF1 and PRF2) are employed in the two phases.
- The 32 most Significant bits of the sequence counter are input to PRF1.
- The least significant 16 bits of the sequence counter are inputs to PRF2 So, the output of PRF2 changes for each frame sent.
- MIC is computed as a function of the data in the frame and also some fields in the MAC header such as the source and destination addresses.

- It also uses as input a key derived from the PTK which was computed during the four-way handshake.
- Due to design constraints on WEP cards, MIC's implementation uses simple logical functions, shifts, etc. Hence, it is not as secure as a keyed cryptographic hash.
- On the other hand, it is much better compared to the CRC checksum used in WEP.

CCMP

- The implementation of 802.11i that uses AES is referred to as WPA-2.its technical name is counter mode with CBC MAC protocol(CCMP).
- In CCMP terminology, this count is referred to as a packet number (PN).
- The count is maintained at both sender and receiver ends.
- The PN is also included in a special CCMP header field in a CCMP frame.
- The PN is incremented by the sender after each frame is sent.
- Upon receipt of a fresh frame in that session, the receiver compares the value of PN in the CCM header versus the value stored by it.
- If the value is less than the stored value, the frame is likely to be a replayed frame and is hence discarded.
- The first task in preparing a frame for transmission is to compute a MIC.
- The ***MIC*** is computed using AES in ***Cipher Block Chaining (CBC) mode*** with block size 128 bits.
- The key for performing encryption in each stage of Fig below is TK(temporal key).
- The IV for the MIC computation is a "nonce," which includes the 48-bit PN.
- The second and third blocks used in the MIC computation are specific fields in the frame header such as the MAC addresses, sequence control, and frame type.
- Next, the blocks in the frame data are sequentially processed resulting in an 8-byte MIC.
- The next step is encryption.
- The frame data and the MIC are concatenated and then encrypted using AES in counter mode (Fig. 15.7).
- Let n be the total number of blocks in the ***frame body + MIC.***
- The procedure for encrypting the i-th block is:
- Compute **$A_i = E_{TK}(PN + i*j)$.** Here, PN is the packet number and j is a constant known to both sender and receiver.
- Compute i-th block of ciphertext = A (xor)Pi.
- Here, Pi is the i-th block of plaintext.
- The frame now includes two new fields — the CCMP header and the MIC.
- Upon receipt of the frame, the receiver reverses the operations performed by the sender.
- It performs decryption followed by MIC verification.



IV = Initialization Vector (includes 48-bit Packet Number)

AAD1, AAD2 = Additional Authentication Data (includes certain immutable fields of the MAC header)

COUNT is a function of the Packet Number

Fig : MAC generation and encryption in CCMP

Firewalls

- ***Definition:*** A firewall acts as a ***security*** guard controlling access between an internal protected network and an external untrusted network based on a given security policy.
- Besides preventing intruders getting in, a firewall also helps prevent confidential inside data from getting out.
- A firewall may be implemented in hardware as a ***stand-alone "firewall appliance" or in software on a PC.***
- A single firewall may be adequate for small businesses and homes. However, in several large enterprises, multiple firewalls are deployed to achieve ***defence in depth.***

2.1 BASICS

2.1.1 Firewall Functionality

- The main functions of a firewall are listed as follows:
- ***Access Control:***
  - ✓ A firewall filters incoming (from the Internet into the organization) as well as outgoing (from within the organization to the outside) packets.
  - ✓ A firewall is said to be configured with a rule set based on which it decides which packets are to be allowed and which are to be dropped.
- ***Address/Port Translation.***
  - ✓ NAT was initially devised to alleviate the serious shortage of IP addresses by providing a set of private addresses that could be used by system administrators on their internal networks but that are globally invalid (on the Internet).
  - ✓ it is possible to conceal the addressing schema of these machines from the outside world through the use of NAT.
  - ✓ Through NAT, internal machines, though not visible on the Internet, can establish a connection with external machines on the Internet. NATing is often done by firewalls.
- ***Logging.***
  - ✓ A sound security architecture will ensure that each incoming or outgoing packet encounters at least one firewall.
  - ✓ The firewall can log all anomalous packets or flows for later study.
  - ✓ These logs are very useful for studying attempts at intrusion together with various worm and DDoS attacks.
- ***Authentication, Caching,*** etc. Some types of firewalls perform authentication of external machines attempting to establish a connection with an internal machine.
- A special type of firewall called ***web proxy*** authenticates internal users attempting to access an external service. Such a firewall is also used to cache frequently requested webpages. This results in decreased response time to the client while saving communication bandwidth.

2.1.2 Policies and Access Control Lists

- High-level policies for access to various types of services are formulated within an organization or campus. Examples of these include the following:
  - ✓ All received e-mail should be filtered for spam and viruses.
  - ✓ All HTTP requests by external clients for access to authorized pages of the organization's website should be permitted.
  - ✓ DNS queries made by external clients should be allowed provided they pertain to addresses of the organization's publicly accessible services such as the web server or the external e-mail server. However, queries related to the IP addresses of internal machines should not be entertained.
  - ✓ The organization's employees should be allowed to remotely log into authorized internal machines. However, all such communication should be authenticated and encrypted.
  - ✓ Only two types of outgoing traffic are permitted. First, all e-mail from within the organization to the outside world are permitted. Second, requests emanating from within the organization for external webpages are permitted. However, requests for pages from certain "inappropriate" websites should be denied.
- ***High-level policies are translated into a set of rules that comprise an Access Control List.***
- A rule specifies the action to be taken as a function of

(i) the packet's source IP address and port number

(ii) the packet's destination IP address and port number

(iii) the transport protocol in use (TCP or UDP)

(iv) ***the packet's direction — incoming or outgoing***

- The Access Control List for the high-level policies is described in Table 21.1.
- Policies can, in general, be either ***permissive or restrictive.***
- ***A permissive policy is defined as follows:***
  - ✓ ***Permit all packets except those that are explicitly forbidden.***
- ***A restrictive policy, on the other hand, is defined as follows:***
  - × ***Drop all packets except those that are explicitly permitted.***
- The ACL in Table 21.1 implements a restrictive policy — the default action is Deny as expressed in rules 5 and 8.
- The rules are scanned top to bottom.
- As soon as a rule is found' that matches the packet's attributes (IP addresses, port numbers, etc.), the action in that rule (usually permit or deny) is taken and no further rules are processed for that packet.
- The scanning order is important.
- For example, if rules 4 and 5 in Table 21.1 are interchanged, then IPSec traffic will be dropped.
- Also, from a performance perspective, it makes sense to put the most frequently acted upon rule earlier on.
- By so doing, we can expedite the decision on what to do with a packet.

- Finally, it is important to include the default deny rule at the end of the rule set — this prevents ambiguity over what action to take for a packet that has not been matched against the attributes in any of the previous rules.

Table 21.1 *Example access control list*

No.	In-bound (I) or out-bound (O)	Transport protocol	Src. IP addr.	Src. port	Dest. IP addr.	Dest port	Action	Comment
1	I	TCP	Any	Any	MS	25	Permit	Allow incoming e-mail
2	I	TCP	Any	Any	WS	80	Permit	Allow requests for organization's webpages
3	I	UDP	Any	Any	NS	53	Permit	Allow DNS queries
4	I	IPSec	Any	Any	*	*	Permit	Allow incoming VPN traffic
5	I	Any	Any	Any	Any	Any	Deny	Forbid all other incoming traffic
6	O	TCP	Any	Any	Any	25	Permit	Allow outgoing e-mail
7	O	TCP	Any	Any	*	80	Permit	Allow requests for external webpages
8	O	Any	Any	Any	Any	Any	Deny	Forbid all other outgoing traffic

Note: MS, WS, and NS are the IP addresses of the organization's e-Mail Server, Web Server, and DNS server (Name Server), respectively. * depends on configuration

2.1.3 Firewall Types

- Firewalls can be classified into the categories
  1. **Packet Filters**
  2. **Stateful Inspection**
  3. **Application Level Firewalls**

1. <u>Packet Filters</u>

- This involves checking for matches in the IP, TCP, or UDP headers.
- For example, it may be necessary to check whether a packet carries a certain specific source or destination IP address or port number.
- It is often performed by the border router or access router that connects the organization's network to the Internet.
- In effect, the border router becomes the first line of defence against malicious incoming packets.
- why the packet filtering firewall is inadequate????

Drawbacks:

- Consider an external mail server (IP address = ABC) that wishes to deliver mail to an organization.
- For this purpose, it should first establish a TCP connection with the organization's mail server, MS.
- Consider the arrival of a packet with the following attributes:
- Source IP address = ABC
- Destination IP address = MS
- TCP destination port = 25 (SMTP port)
- ACK flag set
- Such a packet would be part of a normal flow provided a connection between ABC to MS has been established. But suppose such a connection has not yet been established.
- Should the packet still be allowed in? The simple packet filter will allow the packet to enter even if no prior connection between ABC and MS was established.
- It should be noted that such packets are often used to perform port scans.
- A simple packet filter merely inspects the headers of an incoming packet in isolation. It does view a packet as part of a connection or flow. Hence, it will not be able to filter out such pack `'t arriving from ABC.

2. Stateful Inspection

- A firewall uses packet's TCP flags and sequence/acknowledgement numbers to determine whether it is part of an existing, authorized flow.
- If it is participating in the establishment of an authorized connection or if it is already part of an existing connection, the packet is permitted, otherwise it is dropped.
- In the above example of the packet from ABC, the stateful packet inspection firewall will realize that it has not encountered the first two packets in the three-way handshake and will hence drop this packet.

3. Application Level Firewalls

- A packet-filtering firewall, even with the added functionality of stateful packet inspection, is still severely limited.
- What is needed is a firewall that can examine the application payload and scans packets for worms, viruses, spam mail, and inappropriate content. Such a device is called a ***deep inspection firewall.***
- A special kind of application-level firewall is built using proxy agents. Such a ***"proxy firewall" acts as an intermediary*** between the client and server.
- The client establishes a TCP connection to the proxy and the proxy establishes another TCP connection with the server as shown in Fig. 21.1.
- To a client, the proxy appears as the server and to the server, the proxy appears as the client. Since there is no direct connection between the client and the server, worms and other malware will not be able to pass between the two, assuming that the proxy can detect and filter out the malware. Hence, the presence of the proxy enhances security.

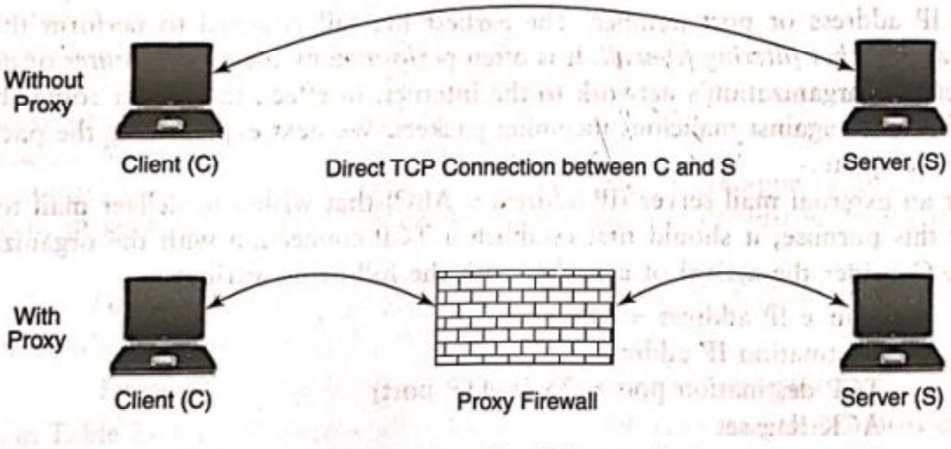


Figure 21.1 *Proxy firewall*

Two TCP connections between C and Proxy and between Proxy and S
Figure 21.1 Proxy firewall

- There are proxy agents for many application layer protocols including HTTP, SMTP, and FTP.
- In addition to filtering based on application layer data, proxies can perform client authentication and logging.
- An HTTP proxy can also cache webpages.
- Caching has a major impact on performance.
- If the webpage is cached in a web proxy server located in the client's organization, the response time could be greatly reduced compared to that where the page has to be fetched from the external web server.
- Also, caching reduces the demand on external communication bandwidth while easing the load on the web server.
- Firewalls are a necessary element in the security architecture of an organization that permit access to/from the external world. In the next section, we study firewall deployment.

2.2 PRACTICAL ISSUES

- The security architecture of a medium size or large organization includes firewalls, proxy servers, VPN terminators, and intrusion detection/prevention (IDS/IPS) devices.

2.2.1 Placement of Firewalls

- We first note that firewalls help segregate or isolate the network into ***multiple security zones.***
- Each firewall in the organization enforces rules that control the transfer of packets between different security zones.
- At the very least, there are three zones —

1. the Internet,
2. the region containing the publicly accessible servers,
3. and the internal network.

- Figure 21.2 depicts a four-zone layout using three firewalls.

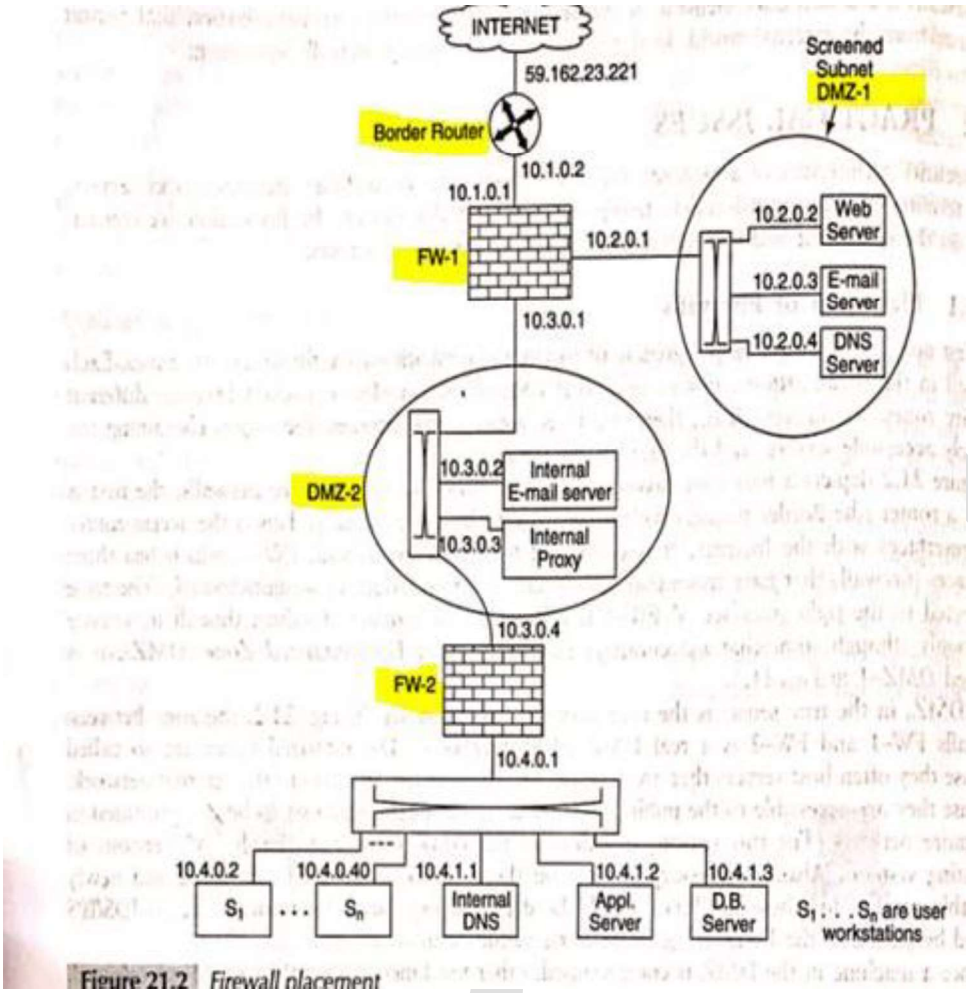


Figure 21.2 *Firewall placement*

- Of the three firewalls, the first is really a router (the Border Router) with some packet-filtering capability.
- This is the access router interfaces with the Internet.
- It is connected to a stateful firewall, FW-1, which has three interfaces (firewalls that have more than two interfaces are referred to as multi-homed).
- The zone connected to the right interface of FW-1 is referred to as ***a screened subnet*** though it is more commonly referred to as a De-Militarized Zone (DMZ). It is labelled DMZ-1 in Fig. 21.2. A DMZ, in the true sense, is the area between two firewalls.
- In Fig. 21.2, the zone between firewalls FW-1 and FW-2 is a real DMZ labelled ***DMZ-2.***

- Demilitarized zones are so called because they often host servers that are accessible to the Internet and also to the internal network.
- Because they are accessible to the public, they are the most likely machines to be compromised in the entire network.
- Once a machine in the DMZ is compromised, other machines in the DMZ could get infected.
- DMZ-1 contains the publicly accessible servers.
- These include the web server, the external e-mail server, and the DNS server. All incoming mail from the Internet is received by this e-mail server, which checks for virus signatures and spam mail.
- The DNS server resolves names of publicly accessible servers. However, care should be taken to ensure that it does not contain address records of any of the internal machines. DMZ-2 contains the internal e-mail server. This is the server that hosts the mailboxes of the company employees. It handles the sending and receiving of all mail between internal parties. It Periodically establishes a connection to the external mail server (in DMZ-1) to retrieve all incoming mail.
- Outgoing mail (from the internal network to the Internet) can be handled in several ways. The internal mail server can set up an SMTP connection to a remote mail server to transfer mail.
- Alternatively, it can connect to the external mail server (in DMZ-1) and use it to relay all outgoing mail.
- DMZ-2 also contains an Internet proxy server.
- All internal users who wish to access external webpages connect to the proxy.
- The proxy authenticates the internal user and decides whether a page can be accessed (different restrictions might apply to different classes of users).
- The proxy scans incoming webpages for virus signatures and objectionable content. Finally, the proxy also performs caching of webpages.
- The internal network contains application servers, database servers, and user workstations.
- It also has an internal DNS server. This DNS server is different from the external DNS server in that it provides mappings between the domain names of the internal machines and their IP addresses.
- The internal machines all have private addresses. It is neither necessary nor desirable for third parties on the Internet to be aware of the private addresses of the internal machines. Hence, this DNS server is placed in the internal network.
- A feature of the security architecture in Fig. 21.2 is that services such as DNS and e-mail are ***split;***that is, there is an internal DNS server as well as an external one.
- Likewise, there is an internal e-mail server and an external one.
- Generally, no external connection should be allowed to the internal servers.
- Connections in the reverse direction from the internal servers to hosts on the Internet should either be forbidden or severely restricted.

2.2.2 Firewall Configuration

- In order to create a firewall ruleset, we need to identify all the possible authorized connections that might be set up between pairs of machines in two different zones adjacent to the firewall.
- We first present a simplified version of the ruleset for firewall FW-2 (Table 21.2).
- Table 21.2 Simplified ruleset for firewall, FW-2

Table 21.2 *Simplified ruleset for firewall, FW–2*

No.	From IP Addr.	From Port	To IP Addr.	To Port	Protocol	Action
1	*	*	Internal	*	*	Drop
2	User	*	Int_Mail_S	25	SMTP	Accept
3	User	*	Proxy	80	HTTP	Accept
4	*	*	DMZ-2	*	*	Drop

*Wildcard

- The first rule states that ***no machine*** from any other security zone is permitted to establish a TCP connection to any internal machine.
- Rules 2-4 assert that, other than connections from internal stations to the internal mail server (on port 25) and web proxy (on port 80), no other connections are permitted to DMZ-1, DMZ-2, or the Internet.
- Table 21.3 shows the ruleset for firewall FW-1.

Table 21.3 *Simplified ruleset for firewall, FW–1*

No.	From IP Addr.	From Port	To IP Addr.	To Port	Protocol	Action
1	*	*	DMZ-2	*	*	Drop
2	Int_Mail_S	*	Ext_Mail_S	25	SMTP	Accept
3	Internet	*	Ext_Mail_S	25	SMTP	Accept
4	Internet	*	Web_S	80	HTTP	Accept
5	Internet	*	DNS-S	53	UDP	Accept
6	*	*	DMZ-1	*	*	Drop
7	Proxy	*	Internet	80	HTTP	Accept
8	Ext_mail_S	*	Internet	25	SMTP	Accept
9	*	*	Internet	*	*	Drop

Rule 1 in Table 21.3 states that no TCP connection is to be established to any machine in DMZ-2 from any machine in DMZ-1 or the Internet.

Rule 2 states that the external mail server can accept connections from the internal mail server to receive incoming mail or to send outgoing mail.

Rule 3 allows connection to the external mail server from mail server on the internet to deposit incoming mail.
Rule 4 and 5 permit connections from the internet to the organizations web server and external DNS server, respectively.
Rule 6 states that no other connection may be set up to any machines in DMZ-1 for any other purpose.
Rule 7 and 8:the internet proxy in DMZ-2 and external mail server are permitted to make connections to machines on the internet to access webpages and to send outgoing mail.
Rule 9: confirms that no other connection from the organizations machine to the internet for any other purpose is allowed.

MODULE 4

Viruses, Worms and Other Malware,. Intrusion Prevention and Detection

INTRUSION PREVENTION AND DETECTION

22.1 Introduction

- ***Definition :***An intrusion is the ***act of gaining*** unauthorized access to a system so as to cause loss or harm.
- Examples of intrusions include the following:
  - ✓ ***Unauthorized login*** to a system by illegally acquiring a password (through, for example, a password guessing attack). –
  - ✓ ***Worm infections*** that use the system as a launch pad to spread and infect other machines.
  - ✓ ***Injection of spyware*** that passively monitors the activities of the user and relays this information back to the attacker (over the Internet, for- example).
  - ✓ Flooding the host with ***spurious connection requests*** that attempt to exhaust the target's resources — processing power, memory, or communication bandwidth.
- Two ways of handling intrusions are ***intrusion prevention and intrusion detection.***

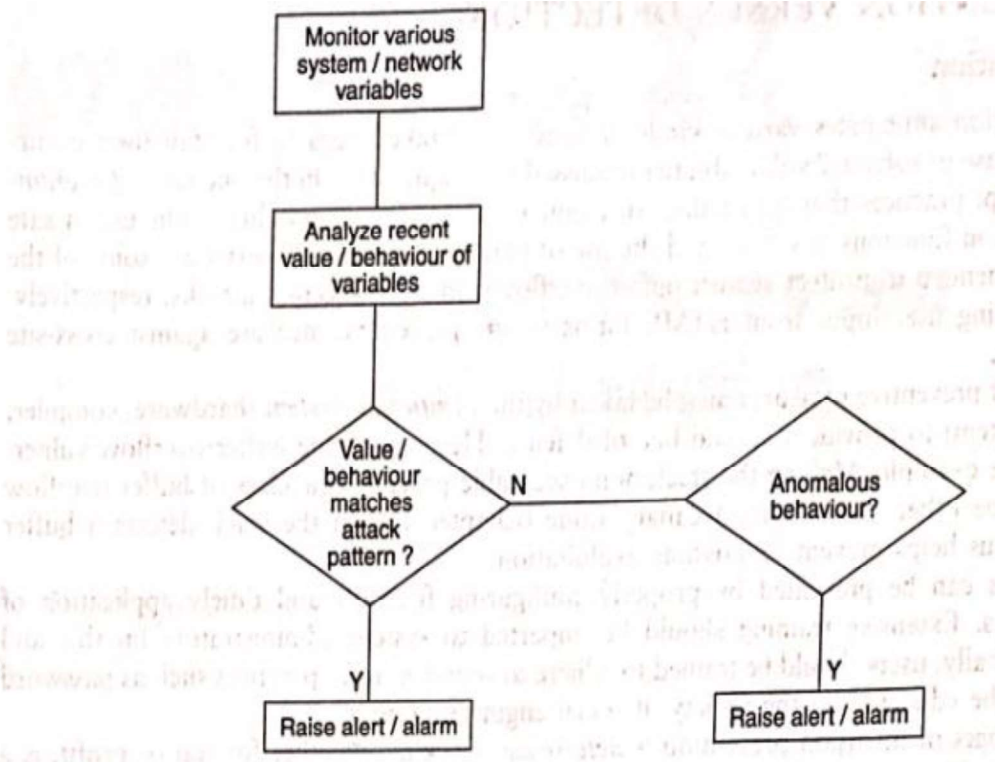


Figure 22.1 *Tasks performed by an IDS*

22.2 Prevention Versus Detection

Prevention	Detection
➢ Intrusion prevention anticipates various kinds of attacks and takes steps to forestall their occurrence. ➢ On the one hand, programmers should adopt practices that help reduce or eliminate software vulnerabilities. ➢ The use of safe string manipulation functions in C/C++ and the use of parameterized SQL queries are some of the practices recommended to protect against buffer overflow and SQL injection attacks, respectively. ➢ Likewise, sanitizing user input from HTML forms is one preventive measure against cross-site scripting attacks. Another set of preventive measures may be taken by the computing system (hardware, compiler, or operating system) to provide a second line of defence. ➢ Extensive training should be imparted to system administrators on this and related tasks. ➢ Finally, users should be trained to adhere to sound security practices such as password protection and be educated on the variety of social engineering attacks. ➢ One final aspect of intrusion prevention is deterrence. Hacking, whether for fun or profit, is a criminal offence.	➢ An intrusion detection system (IDS) (Fig. 22.1) performs the following three tasks: ➢ First, it monitors "events of interest" occurring in the target system or in the network. ➢ An event of interest may be a system call (a call made to the operating system) to, for example, open a file containing sensitive data. ➢ Another event of interest may be the attempted establishment of a TCP connection from a specific IP address to a certain port. 2. ➢ An IDS generates a large amount of data which it then analyzes and converts into valuable information to be used by system administrators. ➢ These are examples of thresholds and parameters set by a human. ➢ On the other hand, it would be highly desirable if the IDS were capable of learning what is normal behaviour, detecting anomalous events when they occur, and flagging such events. ➢ There are a number of key questions related to IDS functioning and deployment: ➢ What are the variables that the IDS should monitor? ➢ When should an alert be raised? When should an alarm be sounded? ➢ Where should the IDS be placed?

22.2.3 Case Study: Unauthorized User Logins

➢ To prevent unauthorized logins owing to compromised passwords, the following should be adhered to:

1.A password should be at least **eight-characters** long, hard to guess, and include at least one non-alphanumeric character.

2. A password should be changed at least ***once in two months.***

3. Passwords should be ***stored securely*** (not written on sticker pads) and should not be communicated to friends, relatives, and co-workers.

4. After three consecutive ***unsuccessful attempts*** to a specific account, the system should be designed to disable all further log-in attempts for the next 20 minutes.

- Rules 1 and 2 must be enforced by the system.
- Rule 3 involves the user alone,
- rule 4 involves the system alone.
- These rules are all measures intended to prevent intrusion.
- As a further preventive measure, a high-security organization may mandate two-factor authentication — ***passwords in conjunction with biometrics.***
- In addition to prevention, an IDS may also be deployed to monitor suspicious log-ins.

22.3 TYPES OF INTRUSION DETECTION SYSTEMS

- A real-world IDS monitors and mines hundreds of variables for interesting patterns.
- Table 22.1 shows a sample of variables together with a condition that may trigger an alert.
- Some of the variables are mere bit patterns in the packet header or the packet payload.
- Other variables are counts of a certain occurrence within a time interval.
- We next classify intrusion detection systems based on their functionality.
    1) **Anomaly versus signature based IDS**
    2) **Host based versus network based IDS**

Table 22.1 *Events of interest to an IDS*

Variable monitored	*Event of interest*	*Possible attack*
No. of accesses to specific file	Tenfold increase over norm	DoS attack or flash event
Login frequency to particular account	Unusually high	Attempted break-in
No. of distinct source IP addresses of arriving packets	Very high	Worm attack
Ratio of ARP request packets to ARP response packets	>> 1	Network scan to identify local active hosts
Ratio of TCP SYN packets to TCP FIN packets	>>1	Possible DoS attack
Percentage of half-open TCP connections	Sudden surge	Possible DoS/DDoS attack
TCP header flags	Invalid combination	Port scan, OS fingerprinting
TCP connection establishment	Unused destination port	Attempt to find which services are open
Payload of incoming packet	Specific bit sequence present	Specific worm attack
O.S. calls	Particular sequence of calls	Specific virus attack

Table 22.1 Events of interest to an OS

22.3.1 Anomaly versus Signature-Based IDS

AnomalyBased IDS	Signature-Based IDS
➢ Anomaly based intrusion detection involves making a determination whether the ***behaviour of the system is a statistically significant departure from normal.*** ➢ The IDS will have to learn, over time, what constitutes normal activity, usage, and behaviour. ➢ The first six conditions in Table 22.1 are examples of what an ***anomaly based IDS would monitor.*** ➢ Consider monitoring the number of TCP SYN packets (with the **SYN flag set**) and FIN Packets (with the **FIN flag set)** in each successive 10-second interval. ➢ A disproportionate number of **SYN** packets vis-a-vis FIN packets indicate several half-open TCP connections and possibly the onset of a ***SYN flooding attack.***	➢ ***Signature-based intrusion detection*** (also called ***misuse detection***) works by identifying specific Patterns of events or behaviour that indicate or accompany an attack. ➢ Each such pattern is called a ***signature.*** ➢ A signature-based IDS maintains a database of known ***signatures.*** ➢ It attempts to obtain a match between the ***currently observed behaviour of the system and an entry in this database***. ➢ A real world signature-based IDS will have thousands of attack signatures against which to compare. ➢ ***An example of an attack signature is a specific bit sequence in a worm payload.*** ➢ In a signature-based IDS it is the presence of a specific signature that raises an alert. ➢ On the other hand, it is possible that a spread of the worm has caused much network traffic congestion and greatly increased CPU utilization on infected machines.

22.3.2 Host-based versus Network-based IDS

Network-based IDS	Host-based IDS
➢ An IDS that captures information about packets flowing through the network is referred to as ***network-based IDS.*** ➢ For reasons of performance, it is common to have stand-alone appliances that perform network-based intrusion detection. These typically run only the IDS and are hence not vulnerable to various worm and virus attacks.	➢ A host-based IDS is typically implemented in ***software and resides on top of the host's operating system.*** ➢ Its main job is to monitor the internal behaviour of the host such as the ***sequence of system calls made, the files accessed***, etc. ➢ For this purpose, it makes use of ***system logs, application logs, and operating***

➢ They may be ***deployed at multiple points*** in a large organization.	***system audit trails to*** identify events related to an intrusion. ➢ ***Operating system logs***, for example, keep track of when ***users log in,*** the number of ***unsuccessful login attempts, the commands executed, network connections made,*** etc. ➢ Application logs keep track of which files have been opened or which registry keys have been accessed during the run of an application. ➢ File system integrity checkers, for example, compute a cryptographic hash on the contents of each file. They detect file changes by comparing the computed hash of a file to its stored hash.

- Two desirable features of an IDS are ***speed and accuracy.***
- Speed is especially important in ***fast-spreading Internet worms,*** for example.
- ***Early worm detection*** and an early response mechanism such as automated system shutdown can help reduce the number of infected 'machines. The IDS should be able to detect every instance of an intrusion.
- An undetected intrusion is referred to as a ***false negative***.

22.4 DDoS ATTACK PREVENTION and DETECTION

22.4.1 DDoS Prevention

1) **Preventive Measures At The Host**
2) **Preventive Measures Inside The Network**

Preventive Measures at The Host

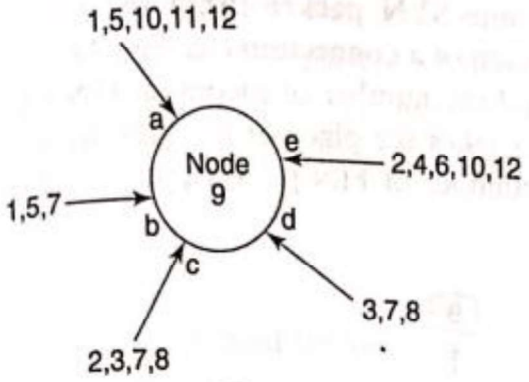
- One possible way of handling **SYN** attacks is to drop requests for TCP connections.
- But this could result in collateral damage if the victim is unable to distinguish between SYN packets that are part of the attack and those from its legitimate clients.
    1. One way to reduce collateral damage is to categorize IP addresses as "almost certainly genuine", ***"probably spoofed",*** etc.
    2. The "***almost certainly genuine"*** addresses are those with whom normal connections were established and terminated in the past.
    3. Under rapidly increasing load, packets with ***unfamiliar source addresses*** are discarded with high probability.

- Another strategy under high-load conditions is to allocate a full buffer of about 300 bytes for a given TCP connection request only upon completion of the three-way handshake.
    1. While the connection is still half-open, minimal information about it is stored in a hash table called the SYN cache.
    2. This information includes the TCP sequence numbers and source/destination addresses and ports.
    3. An alternative to the SYN cache is the SYN cookie, which stores no state information at all for each half-open connection.
    4. Instead, the responding machine places a cookie within the ***Sequence Number field of the second handshake message.***
    5. The cookie is computed as a hash function of the source address, destination address, source port, destination port, and a secret.
    6. The initiator of the connection dispatches the cookie it just received in its ACK message (third message of the three-way handshake).
    7. Upon receiving the ACK, the responder re-calculates the cookie and verifies that it matches the value enclosed in the received ACK.
    8. Only then does it reserve buffer space for the connection.
    9. If the source IP address in the first message of the handshake were spoofed, the cookie in second message would not be received by the initiator but by the machine corresponding to the spoofed IP, address.
    10. The initiator would not be able to complete the three-way handshake since it does not know the cookie value. Hence, its connection request would not be granted buffer space.

Preventive Measures inside the Network

- An intuitively appealing approach to frustrating DDoS attacks is to implement measures closer to the source of the attack.
- One such measure is ***egress filtering.***
- ***Attack:***Most DDoS attack packets use ***spoofed source IP addresses***.
- Address spoofing is employed to confuse cyber sleuths making it hard to pinpoint the true source of the attack.
- The perpetrator hopes to continue the attack for as long as desired and perhaps even resume it at a later point without being traced.
- ***Solution:*** The ***egress router is the last router*** encountered by any packet generated inside the network before it exits that network and enters the Internet.
- Let A be the set of all externally visible IP addresses within the network (behind the egress router). The egress router examines the source address of each packet leaving it.
  If the address does not match any address in A, it drops the packet.
  By thus detecting and filtering spoofed packets' it helps prevent DDoS attacks.
- The idea of ***egress filtering has been extended to routers*** in the core of the Internet.

- A filter, on the other hand, uses the packet's source address to make a decision on whether or not to discard the packet.
- To implement ***Distributed Route Filtering (DRF)***, a filter maintains, for each of its interfaces, the set of all ***source addresses*** from which packets arrive en route to some destination.
- The router uses ***BGP routing information*** to obtain the ***latest mapping*** between each of its interfaces and the subset of source addresses using that interface.
- The filtering decision is straightforward — if a packet with source IP address = S arrives via an interface that it should not have, that packet is assumed to be spoofed and is hence discarded.
- Figure 22.2(a) shows an example of a router implementing DRF.



(a) Router implementing DRF

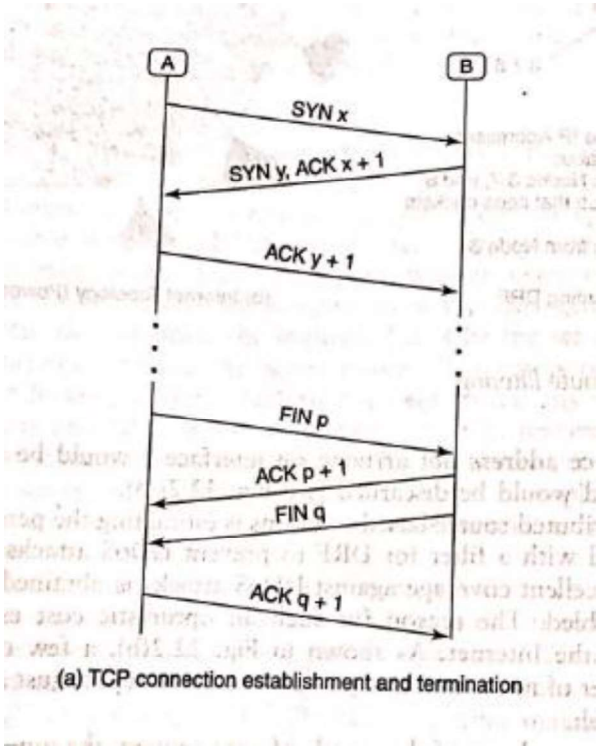
Figure 22.2 *Distributed route filtering*

- Each of its interfaces is marked with the source addresses that use that interface en route to some destination.
- Note that packets from the same source may enter the router through different interfaces.
- For example, packets from source address 7 may arrive through interfaces b, c, or d.
- In the simplest implementation of the filter, the ***router checks whether a packet has arrived on one of its "acceptable" interfaces based only on the packet's source IP address.***

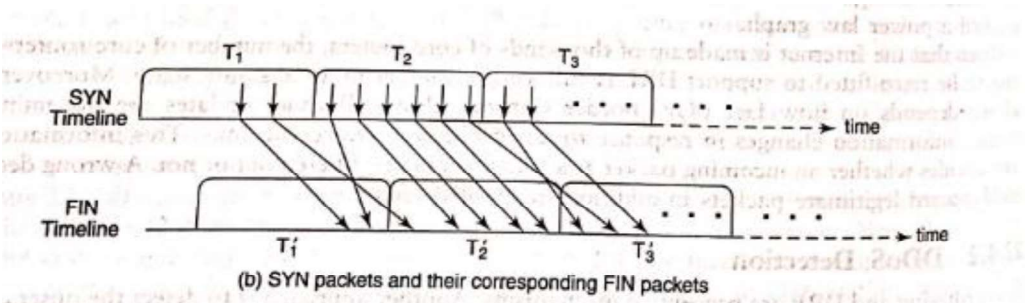
- For example, a packet bearing source address = 7 arriving on interface c would be forwarded. However, another packet with the same source address but arriving on interface e would be suspected of having a spoofed source address and would be discarded [see Fig. 22.2

22.4.2 DDoS Detection

- Egress filtering and DRF are preventive mechanisms.
- Another approach is to detect the onset of DoS and then take remedial action.
- In a SYN flood attack, the victim sees a disproportionate number of SYN packets compared to FIN packets.
- By a SYN packet, we mean any incoming packet with the SYN flag set.
- A FIN packet is sent by the side that wishes to terminate the TCP connection.
- If the other party agrees to termination, it responds with its own FIN packet. Thus, SYN and FIN packets usually occur in pairs.



(a) TCP connection establishment and termination



(b) SYN packets and their corresponding FIN packets

e 22.3 *TCP SYNs and matching FINs*

- Figure 22.3(b) shows two horizontal timelines — the top line shows the times of SYN packet arrivals.
- The bottom line shows the corresponding FIN arrivals.
- Time is slotted into fixed-length observation intervals," T1, , during which we record the number of SYN arrivals.
- The corresponding observation intervals for FINs, T1', T2', ... are shifted to the right by the average duration, of a TCP connection.
- To construct an anomaly detection system, we define the following variables as
- ***Si# of SYN packet arrivals in the i-th observation interval***
- ***Fi = # of FIN packet arrivals in the i-th observation interval***
- ***Di = normalized difference between # of SYN and FIN packets in the i-th observation interval, i.e.,***

$$D_i = \frac{S_i - F_i}{F_i}$$

$\mathcal{T} \equiv$ threshold for detection

Consider the *time series*,

$$D_1, D_2, D_3, \ldots$$

- The different algorithms that attempt to detect the onset of a SYN Flood Attack by monitoring the above series.

1. ***Algorithm 1. Raise an alert if the most recently computed detection variable Di exceeds the threshold, i.e., D, >T1***

- Figure 22.4(a) shows D versus time with the threshold set at T1 = 90.
- Some of the problems with this approach are as follows:

(i) The IDS may raise many ***false alarms*** since it bases its decision on point values.

(ii) A modest spike in *D* at just one point is very unlikely to result in memory exhaustion but it does cause the IDS to raise an alarm.

(iii) The *cumulative* effect of the attack packets across the interval will cripple the system but this algorithm will not raise an alarm.

2. ***Algorithm 2 : Raise an alert if the "smoothed average" of the previous values of D exceeds the threshold.***

- This approach uses the well-known technique of ***exponential smoothing.***
- The decision variable at the end of the i-th observation interval is the smoothed average, Si computed using :

$$S_i = \alpha D_i + (1 - \alpha) S_{i-1} \qquad 0 < \alpha < 1 \text{ and } S_0 = 0$$

$$S_i = \alpha D_i + \alpha(1 - \alpha) D_{i-1} + \alpha(1 - \alpha)^2 D_{i-2} \ldots$$

3. ***Algorithm 3. Define a modified cumulative sum of previous values of D. Raise an alert if this value exceeds a threshold.***

- During normal operation, the number of FINs will balance out the number of SYNs and hence Di will be close to 0.
- Let u be an upper bound on the mean of Di during normal operations.
- Let Di' be a shifted versionof Di, i.e.,
- Di' = Di — u.
- The decision variable, Mi used here is defined as
- Mi = (Mi-1 + D'i) where M0=0;

22.4.3 IP Traceback

- There are two principal approaches to IP traceback :
- ***packet marking:***the packet keeps track of the routers it has visited
- ***packet logging:***each router keeps track of the packets passing through it.
- hybrid approaches using a combination of packet marking and packet logging have been proposed.

Probabilistic Packet Marking

- Consider, for a moment that every intermediate router were to ***append its 32-bit IP address*** to each packet it forwards.
- A packet on the Internet traverses about 10 hops on the average, so an extra 40 bytes would be needed to keep track of its path from source to destination.
- This is an unacceptable per-packet overhead.
- Instead, existing but infrequently used fields in the IP header are used to keep track of the routers visited.
- The IP header has a ***16-bit ID field.***
- This field provides support for packet fragmentation and re-assembly.
- Different networks have different restrictions on the size of the datagrams they can carry.

- They may split a datagram into two or more fragments and send each fragment separately.
- The router at the destination end has the responsibility for reassembling the fragments
- To create the original packet.
- All the fragments carry the same number in their ID fields, so they can be identified for re-assembly.
- On the assumption that the ID field is often unused, traceback schemes employing PPM use the ID field to store partial information on intermediate routers.
- But, given that the length of each IP address is 4 bytes, how can a packet store router address information in a 16-bit ID field?
- The answer lies in computing a global fingerprint for each router — this is, say, 16 or fewer bits . of the hash of a router's IP address.
- An intermediate router writes its fingerprint value into the ID field of a packet with probability p.
- Note that it could over-write a previously written fingerprint of a router closer to the source of the attack.
- To identify the perpetrator of the attack, the ingress router at the victim end will need to collect a sufficient number of packets that are all part of the same flooding attack.
- We assume that each ingress router has a map of all upstream routers from it.

Packet Logging

- ***Each router attempts to keep track of every packet that passes through it.***
- Packet logging makes use of the idea of a packet fingerprint or digest.
- This is computed using a well-designed ***hash function*** — one that distributes the hash values uniformly across all possible hash inputs.
- An interesting feature of packet logging is that it can help track even a single rogue packet.
- First, assume that each router stores each packet received by it in the last 5 minutes.
- Suppose the victim wishes to obtain the exact path followed by a packet received by it.
- The idea is that the victim's ingress router, A, queries each of its adjacent routers whether they have seen the packet.

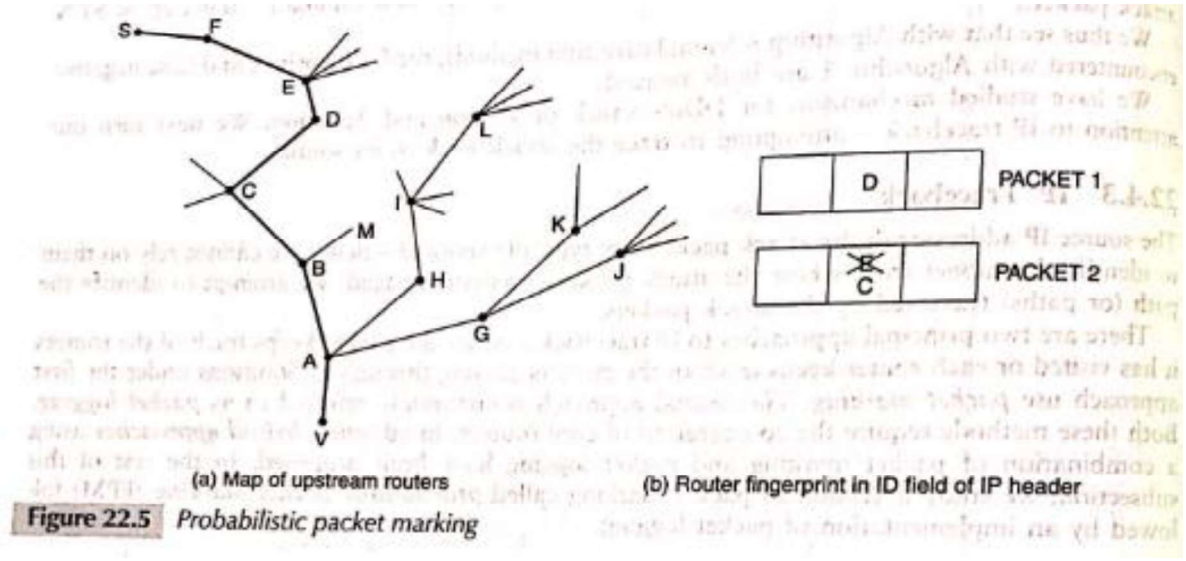


Figure 22.5 *Probabilistic packet marking*

- In Fig. 22.5(a), A would query B, H, and G.
- The router that responds positively, say B, then queries its neighbours, C and M.
- The one that responds positively then contacts its neighbours and so on until the source of the packet is traced.
- The storage requirements can be further reduced by the use of a space-efficient data structure called the Bloom Filter.
- Let n be the maximum number of packets to be stored in a router in a given interval, say 7 minutes.
- Each time an element has to be inserted, one or more hash functions on that element need to be computed.
- Let k be the number of distinct and independent hash functions used. k is a design parameter.
- The output of each hash function returns a w-bit quantity.
- The Bloom Filter is basically a bit array.
- Let $m = 2^w$ be the size of this array.
- ***Packet "Insertion."***:When a packet enters the router, the k hashes are computed on its content.
- To speed up the computation, the hashes are only computed on the invariant parts of the IP header and a small part of the payload, say 10 bytes.
- Suppose the k hash computations yield the values i1, i2, i3,...ik.
- These k hash outputs are used as indices into the bit array.
- To "insert" a packet, the bits in those positions are all set to 1. (If one or more of them were already set, they remain set.)
- ***Packet Presence Check:*** To check if a packet, P, is present in the Bloom Filter, compute the k hashes on it as done during packet insertion.

- Suppose the k hash computations yield the values i1, i2, i3, ... ik. Then, check whether each of the elements of the Bloom Filter are set. If even one of these elements = 0, P has not been encountered by this router.
- We next derive an expression for the probability of a false positive.

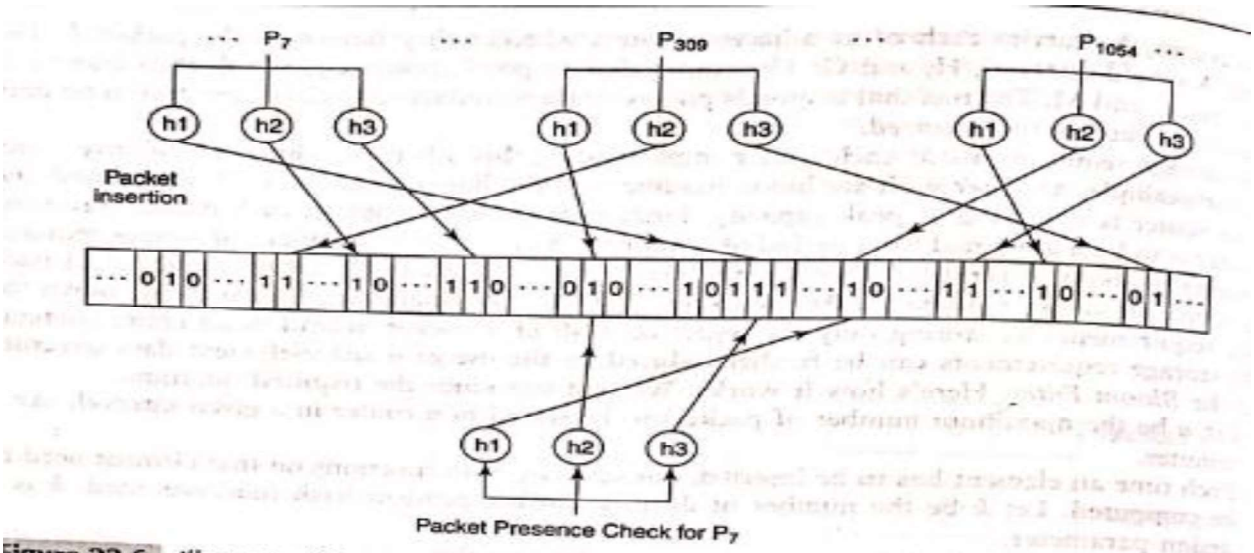


Figure 22.6 *Illustrating false positive in a Bloom Filter*

Probability [a packet hashes to i_1] = $\frac{1}{m}$

Probability [a packet does not hash to i_1]

$$= \left(1 - \frac{1}{m}\right)$$

Probability [none of the n packets hash to i_1 with any of the k hash functions]

$$= \left(1 - \frac{1}{m}\right)^{kn}$$

Probability [at least one of the n packets hashes to i_1 with at least one of the k hash functions]

$$= 1 - \left(1 - \frac{1}{m}\right)^{kn}$$

and at least one of the n packets hash to i_k with at least one of the k hash functions]

$$= \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k$$

It turns out that a reasonably acceptable false positive probability of 1% is achievable with $k = 3$ and $m = 12 \times n$. This translates to a storage cost of only 12 bits per packet at the expense of performing three hash computations per packet. This is considerably less than storing each IP datagram (about 500 bytes on average) or even storing just a 32-bit hash of a packet.

Virus, Worms and Malware

19.2 VIRUS AND WORM FEATURES

19.2.1 Virus Characteristics

- When a virus-infected program is run, the virus code is executed first.
- One of the first tasks of virus code is to seek other programs not yet infected and then pass on the infection to one or more of them.
- A truly malicious virus may then perform actions such as deleting certain files.
- An innocuous virus may attempt something benign like printing a "hello world" message.
- Execution of the virus code is usually followed by execution of the host's original program.
- All the virus code need not be located at the start of the infected file.
- In some cases, virus code is both prepended and appended to the host file.
- Virus code could be split into several segments and interspersed throughout the infected file using JUMP statements at the end of each virus segment.
- In most of these cases, the size of the infected program is larger than the original host program. This helps anti-virus software to detect infected code.
- To evade detection, some viruses modify the file service interrupt handler that returns attributes of files. By so doing, the service handler may be programmed to return the uninfected length of the file.
- Another technique is to use compression so that the length of an infected file remains the same as the length of its original version. The virus writer includes a compression routine in the viral code. To infect another file, the virus first compresses that file and then prepends the virus code to the compressed file.
- The infected file must be uncompressed just prior to execution.

- One of the characteristic features of many viruses is the set of system calls they make. System calls are used by application programs to request services of the operating system.
- They are made to read/write files, spawn new processes, establish TCP connections, etc. Some viruses make calls to copy their own code to other files, create/modify entries in the Windows registry, or search for e-mail. Such "suspicious" calls are often used to distinguish malicious from benign code.

19.2.2 Worm Characteristics

- Classes and features
- Worms are most commonly classified based on their vector of propogation.
- The main categories include:
  1. ***Internet scanning worms***
  2. ***E mail worms***
  3. ***P2P wOrms***
  4. ***Web worms***
  5. ***Mobile worms***

- ***Enhanced Targeting***
  - ✓ The most important attribute of a Worm is that it spreads its infection to other computers.
  - ✓ Many target selection strategies have been proposed and implemented.
  - ✓ Worms that spread through e-mail, for example, have an easy way to figure out their targets.
  - ✓ All they need to do is look into their victim's mailbox or e-mail address book to find a set of targets.
  - ✓ A mobile worm obtains phone numbers of its potential victims from the phone book in the cellphone hosting the worm.
  - ✓ Some web worms use search engines to harvest URLs of potentially vulnerable targets.
  - ✓ Internet scanning worms, on the other hand, scan the IP address space for vulnerable machines.
  - ✓ The most straightforward approach is random scanning — choosing IP addresses at random. This was adopted by Code Red Version-I. However, Code Red Version-II adopted localized scanning.
  - ✓ Over 80% of the rime, it attempted to connect to victims with whom it shared the network address (most significant 8 or 16 bits of the IP address). This strategy was more successful since hosts in the same network are likely to be closer and be running the same soft-ware.
  - ✓ Worms like Nimda, unleashed in September 2001, spread aggressively thanks to its five different vectors of propagation. Propagation through HTTP and e-

mail were particularly successful in penetrating the perimeter of the enterprise. Once inside, it exploited the Windows file-sharing feature to spread within the enterprise.

- ***Enhanced Speed***
  - ✓ To enhance the infection rate, some worms are designed to spawn multiple threads.
  - ✓ Each thread is responsible for setting up connections to a different subset of hosts, thus increasing the rate at which infection is spread.
  - ✓ Some worms reduce infection latency by targeting a buffer overflow vulnerability on an application that employs UDP rather than TCP.
  - ✓ TCP connection establishment involves a three-way handshake and is time-consuming.
  - ✓ UDP, by contrast, is connectionless.
  - ✓ This sharply reduces infection latency.
  - ✓ A steep increase in the number of infected machines at the very outset of a worm epidemic has a multiplicative effect on spreading rate.
  - ✓ For this purpose, the attacker could create one or more hit-lists carrying addresses of several thousand vulnerable machines.
  - ✓ The first worms to be let loose could carry one such list.
  - ✓ As a worm infects each new machine, it splits its list between itself and the machine it has just infected.
  - ✓ Given that most of the machines on the hit-lists are vulnerable, the worm spreads rapidly during the initial stage of the epidemic. Thereafter, the infected machines could spread the infection using random scanning or some other spreading method.
- ***Enhanced Capabilities***
  - ✓ Most worms (and viruses) have unique and distinct signatures — a pattern of bits, usually assembly language code, which appears in all instances of the worm.
  - ✓ Worm and virus signatures are the key to detecting them. However, there are sophisticated code obfuscation techniques to evade detection.
  - ✓ One such technique is the use of encryption for disguising worm code.
  - ✓ Different instances of the worm may use different keys for encryption. Thus, they might fail to match any existing worm signatures. Such worms are said to be polymorphic.
  - ✓ A polymorphic worm would have to be decrypted before being executed. This suggests that a decryptor routine "in the clear" would have to be part of the worm code.
  - ✓ Decryptors themselves may be very simple, involving XOR operations or trivial shift-based substitutions. However, detecting a worm on the assumption that the decryptor routine is invariant would not always succeed.

- ✓ Figure 19.1 shows two versions of assembly code that look different but perform the same function.
- ✓ The second version is inefficient with spurious instructions.
- ✓ The second version also has a spurious branch instruction to confuse worm code detection software that relies on control flow analysis.
- ✓ Worms that have multiple such versions with or without relying on encryption are referred to as metamorphic worms.

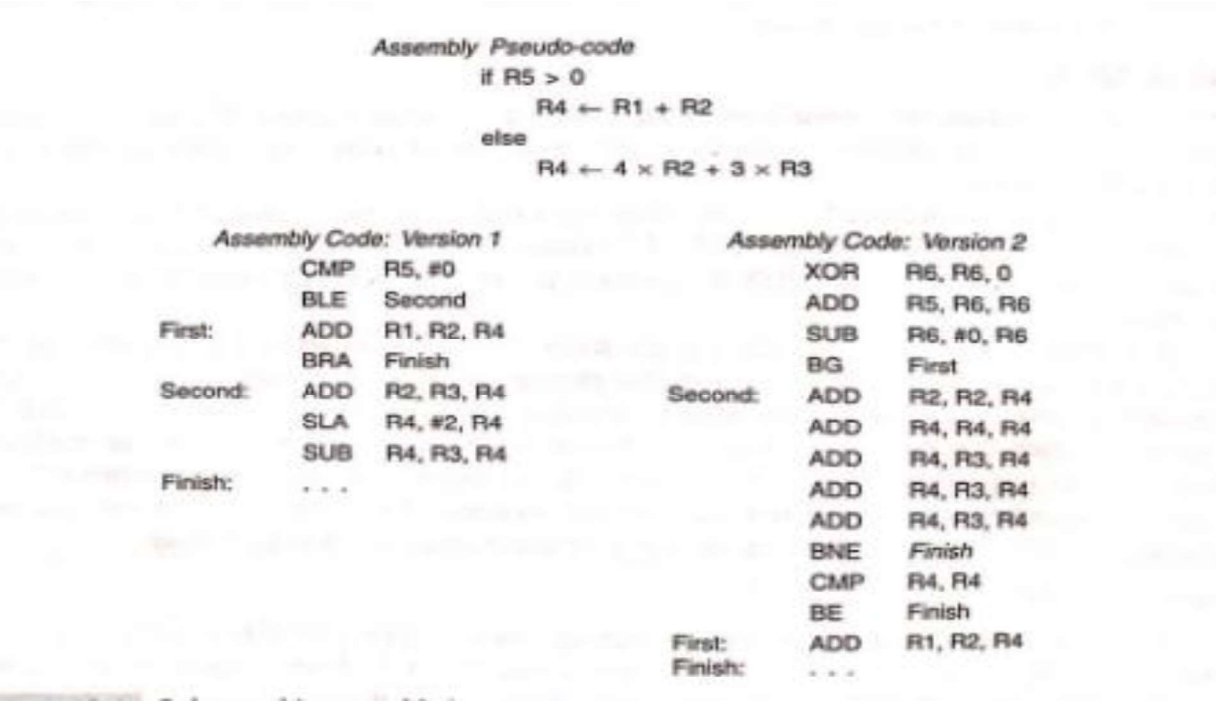


Figure 19.1 Polymorphic assembly language versions of same pseudo-code

- ➢ ***Enhanced Destructive Power***
  - ✓ It is estimated that worms such as Code Red and Nimda caused billions of dollars in damage.
  - ✓ Analysts estimate costs based on lost productivity, clean-up costs, system downtime which affects business and revenues.
  - ✓ Fast-spreading worms also caused severe network congestion problems disrupting normal Internet traffic and contributing to system dos time.
  - ✓ Nevertheless, most worms thus far have been relatively benign.
  - ✓ Some worms contributed atta packets to a DDoS attack or caused website defacement.
  - ✓ The Witty worm which appeared in Mar 2004, however, was qualitatively different. It was the first worm to carry a destructive payload. deleted a random section of the victim's hard disk leading to a system crash

19.3INTERNET SCANNING WORMS

- ✓ One characteristic of Internet scanning worms is that they are self-activated.
- ✓ The ability to spread without human intervention distinguishes them from most types of e-mail, P2P, and web worms.
- ✓ This category of worms is so called because they scan the Internet looking for vulnerable machines.
- ✓ The vulnerability could be a buffer overflow problem in a commonly used service provided by a particular version of an OS.
- ✓ The worm communicates with and delivers its malicious payload to the victim using standard transport protocols such as TCP or UDP.
- ✓ Once installed on the victim, it could erase local files, steal secrets, or deface webpages, but above all it seeks new victims to infect.

19.3.1 Case Studies: Code Red and Slammer

Code Red

- ➢ One of the best known examples of Internet scanning worms is the Code Red worm,
- ➢ It all started on June 18, 2001, when a buffer overflow vulnerability was discovered in the Microsoft IIS Weh Server.
- ➢ A patch for this vulnerability was developed a few days later.
- ➢ It is estimated that there were several million IIS servers in active deployment.
- ➢ Even assuming that a large percentage of these were patched, that still left plenty of room for the spread of the worm, which was unleashed on July 12, 2001.
- ➢ The worm itself was carried in HTTP request messages targeted at IIS servers.
- ➢ The first version of the worm used a random number generator to generate new addresses of machines to infect. However, the same seed was used for the random number generator in every instance of the worm.

Slammer

- ➢ The SQL Slammer was launched on 25 January, 2003, and targeted a buffer overflow vulnerability on the Microsoft SQL server 2000.
- ➢ The worm sent packets on UDP port 1434 — the database software's resolution service.
- ➢ It used simple random scanning to propagate.
- ➢ Slammer's payload was a mere 384 bytes in length — far smaller than the 4 kb payload of Code Red. Also, UDP, being a connectionless protocol, there is no overhead of connection establishment.

Worm Propagation Models

Simple Epidemic Model

- The Simple Epidemic Model used to study the spread of infectious diseases among humans is an appropriate starting point.
- The model assumes that there are only two types of entities in the population.
- Either an individual is susceptible or he is infected.
- An infected individual can infect a susceptible person.
- Once infected, a person remains infected and does not recover.
- Let N be the size of the total population.

Let N be the size of the total population. Let $I(t)$ be the number of infected individuals at time t. The number of susceptibles at time t is then $N - I(t)$. β is the initial infection rate, i.e., each infected person attempts to pass on the infection to β susceptibles in 1 time unit. The following differential equation captures the number of infected persons at time t.

$$dI = \beta I(t)\left(1 - \frac{I(t)}{N}\right)dt \tag{19.1}$$

or

$$\beta\, dt = \left(\frac{dI(t)}{I(t)\left(1 - \frac{I(t)}{N}\right)}\right) \tag{19.2}$$

In an infinite population, each infected person infects $\beta\, dt$ susceptibles in time interval dt. However, in a finite population of size N, the probability that the target of an infective is already infected is $\frac{I(t)}{N}$. Such targets do not add to the population of newly infected. The factor $\left(1 - \frac{I(t)}{N}\right)$ in the above equations ensures that only previously uninfected entities are added to the count of the freshly infected in time interval dt.

Integrating both sides of Eq. (19.2) yields

$$I(t) = \frac{I_0 N}{I_0 + (N - I_0)e^{-\beta t}} \tag{19.3}$$

Kermack—McKendrick Model

- The Kermack—McKendrick (K—M) model more accurately models the spread of human infectious disease by considering three (instead of two) categories of people:
  - those who are susceptible (state S)
  - those who are infectious (state I) and
  - those who are neither, i.e. individuals who are cured or those who have succumbed to the disease (terminal T).
- Initially, all individuals in the population are susceptible.
- It is possible to go from state S to I but not vice versa .

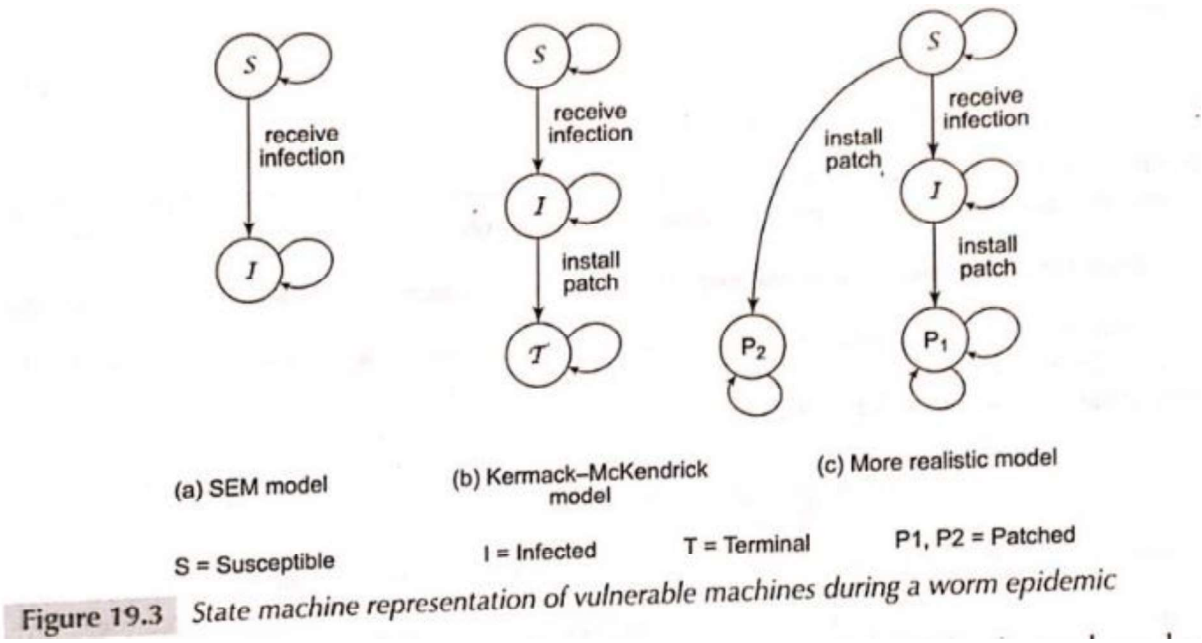


Figure 19.3 *State machine representation of vulnerable machines during a worm epidemic*

19.4 Topological worms

19.4.1 Email worms

- Email worm propagates through infected e mail.
- The victim receives email that comes from trusted or familiar source .
- The victim sees an innocent text file attached to the email.
- In reality, the file named such as loveletter.text.vbs contains Visual Basic script.
- By clicking on this attachment, the embedded VB script executes, sending a copy of itself to every person in the victim's contact list.
- Many e-mail worms exploit the fact that documents created by certain word processors embed software macros in them.
- The macros execute when the document is opened
- For example, Melissa was a macro worm (or "macro virus") that propagated by sending copies of itself to the first 50 persons in the victim's address book.
- One of the best-known e-mail worms of more recent vintage is Sobig, which was let loose in 2003. It spread by communicating malicious c-mail or copying itself to an open network share.
- There were several versions of SoBig. One version could update itself by downloading code from certain websites.
- The URLs of these sites were contained in a file that itself was downloadable from geocitics.com this site allows users to host their own free webpages besides providing tools in support of building dynamic webpages).
- Some of the malicious code received , installed a keystroke logger and stole passwords from its victims.

19.4.2 P2P Worms

- A P2P network is a massively distributed system of computers where each peer or node plays the role of both client and server.
- They are used principally for sharing files, which may contain songs, images, videos, etc.
- Each peer maintains within itself a shared folder of files that it is willing to share with others.
- Users do not download files from a central server but from their peers located across the globe.
- They are immensely popular as evidenced by the fact that a very large proportion of Internet traffic is comprised of P2P packets.
- To see how P2P worms spread, it is important to understand how a P2P network operates.
- Most P2P networks use an overlay network, which is a logical network of peers.
- Two peers are said to be neighbors at any given point of time if there is an active TCP connection between them.
- Here are potential ways in which P2P worms may spread:

1. ***One of the simplest is for a malicious peer to respond positively to any query.***

- ✓ If the requester then chooses to download the file from the malicious peer, the latter sends it an infected file whose name is changed to match that of the requested file.
- ✓ The infected file contains a worm which passes on its infection to the requester.
- ✓ Once infected, the requester mimics the behaviour of the malicious peer thus helping to propagate the infection.
- ✓ Alternatively, various "popular" files stored in the shared folder of a peer may be infected.
- ✓ When any of them is downloaded, the infection spreads to the shared folder of the requesting peer.

2. ***Peers in a given P2P network run the same P2P protocol.***

- ✓ There are usually few different implementations of this protocol leading to little software diversity.
- ✓ An exploitable buffer overflow vulnerability in one popular implementation is a familiar starting point.
- ✓ This, coupled with the fact that a peer maintains a list of neighbours, implies that a worm has ready targets and does not need to perform random scanning as in the case of Internet scanning worms.
- ✓ The first type of worm is said to be passive since it propagates only when requested to download a file.
- ✓ The second type of worm is active since it propagates on its own without receiving requests from its peers.

- ✓ One aspect of P2P worms is that they may result in no apparent traffic anomaly, so an intrusion detection system monitoring network traffic is unlikely to raise an alert.

19.5 WEB WORMS AND CASE STUDY

- Web worms differ from malware such as the Internet scanning worms in several ways.
- Many web worms are executed in browsers which run on diverse hardware/OS plat forms.
- Web worms are written in a high-level language making it easy to perform complex operations but difficult to execute low-level operations.
- On the other hand, many other worms are written in assembly language.
- One type of web worm is the XSS worm — so called since it exploits cross-site scripting vulnerabilities in web servers
- The first step in creating an XSS worm is to inject attack code into a vulnerable web server.
- When a user accesses the infected website through his/her browser, the malicious code (usually Javascript) is downloaded on to the browser.
- As in any XSS vulnerability, malicious code executes on the browser.
- Given that a key function of a worm is to propagate, the challenging question then is "How does an XSS worm propagate?"
- A partial answer to this question may be found through our next case study of the Samy worm.

XSS Worm Case Study

- The XSS worm, Samy was unleashed in October 2005.
- Authored by SamyKamkar, it infected the social networking site, Myspace.
- Social networking sites typically allow users to create and edit their profiles (Fig. 19.5), which are stored on the site and are accessible to other members of the social networking group.
- A user profile may contain information about him including his hobbies, photographs, etc.
- A user profile also contains a list of the user's friends with hyperlinks to their profiles.
- Samy added a bunch of carefully crafted Javascript to his profile.
- When a visitor to Samy's website, say V1, downloaded Samy's profile on to his browser, the Javascript in Samy's profile executed.
- This caused Samy to be added as a friend in V1's profile and also to include the message "but most of all, Samy is my hero.
- " Within 20 hours of the first visit to Samy's profile, Samy had been added as a friend to more than a million user profiles.
- This rate of spread was even faster than that of Code Red.
- How did the worm spread and why did it spread so fast?

- The malicious Javascript uploaded itself on to V1's profile on the MySpace server, thus infecting it.
- This is done by an HTTP Post-request sent from the browser to the server. However, that would cause the screen to freeze between sending the request and receiving the HTTP response from the server.
- To ensure that the viewer had a normal screen experience, Samy'sJavascript created an XMLHttpRequest object which was used to send the malicious Javascript to the Myspace server. Unlike the regular HTTP request, the message from an XMLHttpRequest object is asynchronous nd runs in the background.

19.6 MOBILE MALWARE

19.6.1 Introduction

- New—generation smartphones combine the functionality of a cellphone and a lose-end PC.
- They may be used for storing confidential documents, communicating via e-mail/SMS/MMS, and taking photographs.
- They support feature-rich applications that run on top of a complete OS.
- The most common OS on smartphones is the Symbian followed by Windows Mobile, Linux, and recently the Mac OS X (on the iPhone).
- They provide a rich set of APIs to access the phone book and other files, send SMS/MMS messages, etc. Unfortunately, these very APIs can also be used by malware to, for example, read a confidential document on the smartphone and ship it to the attacker as an MMS attachment.

19.6.2 Bluetooth

- Bluetooth is both a communication technology and a protocol stack.
- As a communication technology,
  - ✓ It supports short-range wireless communication —
  - ✓ A maximum of between 10 and 100 meters between devices.
  - ✓ Bluetooth uses 2.4 ghz shortwave radio technology.
  - ✓ Bluetooth is a complex, multi-layered protocol.

Discovery and User Authorization

- Besides the vulnerabilities mentioned above, social engineering is a key factor in the spread of mobile malware,
- Many PC users do not hesitate to click hot links in their e-mail or open attachments even when the sender of the e-mail is unknown.
- This behavior carries over to the mobile world where many users have no hesitation in accepting a file from an unknown source.

- The combination of Bluetooth implementation vulnerabilities, rich feature set of the smartphone, and unthinking user behavior has exposed the smartphone to various strains of malware.
- To investigate the spread of malware in smartphones, it is necessary to understand the basics of how smart phones exchange files using Bluetooth.
- To discover other Bluetooth-enabled devices in its neighborhood, a device initiates an inquiry procedure, which includes broadcasting an inquiry request.
- All devices in the range of the initiator that are in discoverable mode respond sending their bluetooth device address (BD_ADDR).
- This is a 48-bit MAC address — the first 24 bits identify the device manufacturer/model and the last 24 bits specify a particular instance of that model.
- Bluetooth is a connection-oriented protocol.
- A device, A, can set up a connection to any other device B. But to set up such a connection, A should know the BD_ADDR of B.
- One way of obtaining B'sBD_ADDR is through the discovery procedure.
- A large percentage of users keep their phones in discoverable mode, so this is an attractive way of harvesting device addresses especially In crowded areas such as railway stations or malls. However, it should be noted that even with the phone in non-discoverable mode, there are a number of brute force techniques to extract its BD_ADDR.
- Knowing B's BD_ADDR, A could attempt to exchange files with it using the OBEX (object exchange) protocol.
- A session protocol that resembles HTTP, OBEX is used to transfer images, business cards, and other files between Bluetooth devices.
- An attacker, A, could use OBEX Push totransfer a file containing malicious code to user, B.
- User authorization is usually required before a file can be accepted by his smartphone. Each user selects a PIN which varies between 4 and 16 characters long but 4 characters are typically used.
- The smartphone usually prompts a user to enter his/her PIN as a way to confirm whether an external file, for example, should be accepted.
- Some OS versions accept file transfers without user authorization. And some smartphones allow users to disable the "Authorization Required" option for file transfers.

Link Level Security

- The level of security provided by user authorization alone is generally inadequate.
- Another level of security provided by Bluetooth is link-level authentication and encryption.
- For this purpose, both sides compute a common secret called the link key.

- Bluetooth uses a procedure called **pairing** wherein this key is computed by two participating devices.
- Pairing is preceded by discovery/inquiry and paging.
- The latter is a procedure whereby the discovering device, A, establishes a connection with the discovered device, B.
- Computing the Link Key
    1. The first step in deriving the common link key between A and B is to compute an initialization key, K_{init}.
        - ✓ This is a function of the BD_ADDR of B and a nonce, IN_RAND, generated by A as shown in Fig. 19.7(a).
        - ✓ Before the pairing procedure, the owners of A and B must agree in an off-line manner on a temporary PIN to be used specifically as part of the pairing procedure.
        - ✓ Both users then type in the temporary PIN. K_{init} is also a function of this temporary PIN agreed to by both parties. K_{init} is computed from IN_RAND, BD_ADDR0, and the PIN using an algorithm, E22, based on a block cipher called SAFER+.

    2. To compute the link key,
        - ✓ A and B each generate a random number (LK_RAND_A and LK_RAND_B, respectively).
        - ✓ Each party then performs an XOR of its random number with and they each transmit this across.
        - ✓ Each side recovers the other party's random number by performing an XOR of the received value with K_{init}, (see Fig. 19.7(6)).
        - ✓ Now, each side has LK_RAND_A, LK_RAND_B, and the two device addresses, BD_ADDR_A and BD_ADDR_B.
        - ✓ They then perform identical operations on these to obtain the link key, K_{AB} as shown in Fig. 19.7(a). The operations involve the use of an algorithm, E21, which, like E22, is based on the cipher, SAFER+ .
        - ✓ Thereafter, each device stores the pair, BD_ADDR, of the other device and the newly computed link key in a database. Each device maintains such a database of BD_ADDR, link key pairs, one pair per device it is paired up with.

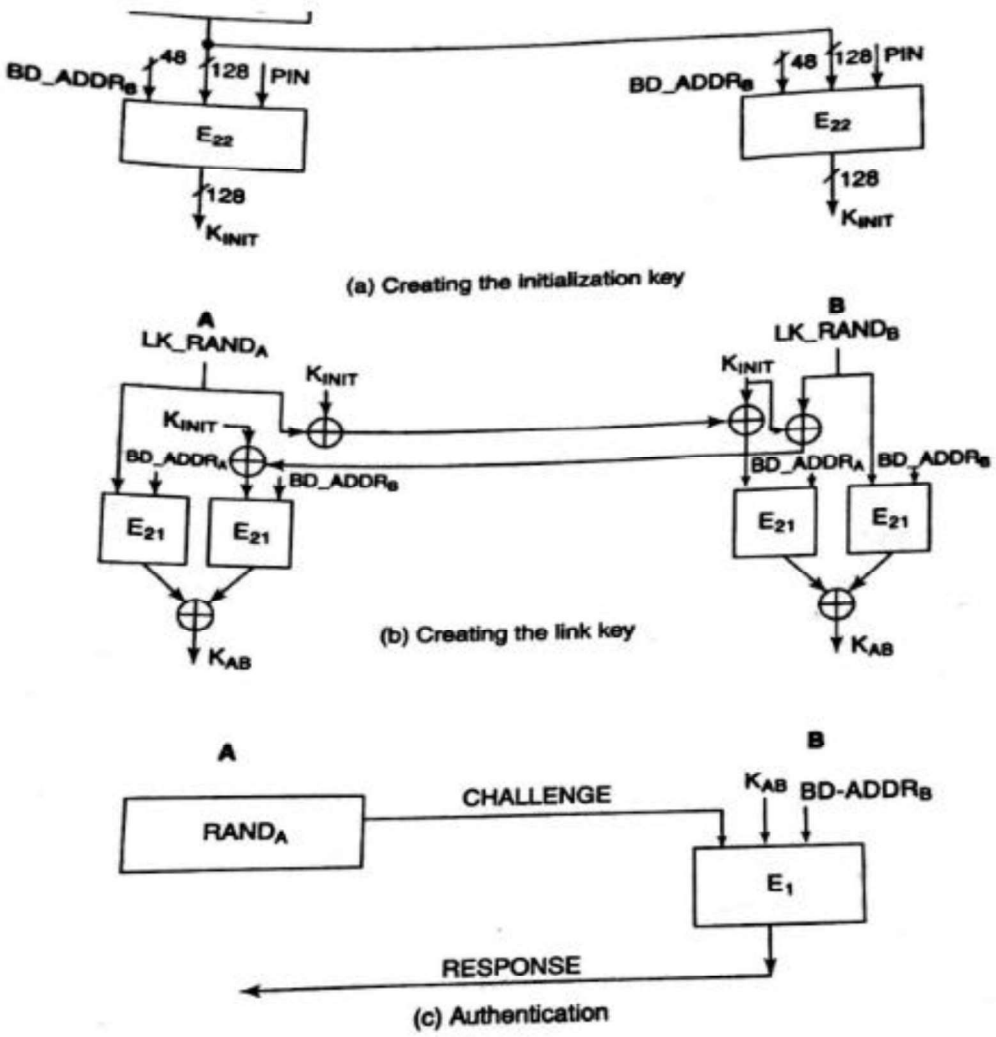
Using the Link Key

- The link key is used for both authentication and encryption.
- Suppose A and B were already paired.
- Let KA, be their common link key.
- Now suppose A wishes to authenticate B.

- For this purpose, it generates a random number, $RAND_A$ (a challenge) and sends it to B.
- The response computed by B is E1(KAB,RANDA,BD_ADDRB).

Hacking the link key

- It is possible to launch a dictionary attack by sniffing each message involved in pairing and authentication.
- These attacks enable an eavesdropper to obtain the link key KAB.



ure 19.7 *Generation and use of the Link key*

19.6.3 Examples

- Cabir was one of the earliest proof-of concept worms that targeted the Symbian Series 60 OS.
- Unleashed in June 2004, it was authored by the International Virus writing group 29A.

- The worm attempts to discover other Bluetooth-enabled phones set in discoverable mode.
- When it finds such a phone, it sends the worm payload in a SIS file.
- The receiver needs to accept and install the file.
- Its payload was mostly benign typically displaying "Caribe" on the screen. However, the continuous scanning for new victims by an infected phone depletes battery power.
- Commwarrior, which appeared in March 2005, was the first worm to spread through, both Bluetooth and MMS.
- Like Cabir, it targeted Symbian smartphones.

19.7 BOTNETS

19.7.1 Basics

- A botnet is an army of compromised computers or bots connected to the Internet and remotely controlled by a ***"botmaster."***
- The earliest botnets were a collection of zombies that participated in DDoS attacks.
- The emergence of botnets is closely linked to the motive of financial gain that is behind many recent cyber attacks.
- They are often used to send spam mail on behalf of third parties,
- For example, Bot programs , may contain keyloggers and other forms of spyware that capture sensitive personal information such as passwords and credit card numbers and send these to the botmaster.
- Botnets have also been used as an extortion tool — "Pay up or your website will be bombarded by a DDoS attack".
- How does a computer become a bot?
- Bots are created in ways similar to many of the traditional trojan/worm/virus infections.
- A common vector of propagation is e-mail that contains an infected attachment.
- Another is through downloading a malicious webpage containing scripts that exploit vulnerabilities in certain browsers or application software.
- A bot infection may also be propagated by bots themselves by scanning the Internet for vulnerable machines.
- Finally, open file shares and IRC (Internet Relay Chat) multicast messages have also been widely used to spread infections.
- One important difference between a bot and a computer infected by a traditional worm/virus/ Trojan is that a bot needs to communicate with specific nodes in the bonet to receive fresh commands.
- A bot may be ordered to send spam or to "Launch a DDoS attack on site abc.com beginning 14:00 hours on 01-12-10." Some of the nodes in the botnet play the role of Command and Control (C&C) servers. They receive commands from the botmaster and disseminate these to the rest of the bots.

19.7.2 Case Study: **The Storm Botnet**

- The Storm botnet was first detected in January 2007.
- Its other names are Peacomm, Nuwar, and Zhelatin.
- Bots in the Storm botnet are infected in stages.
- The most common vectors for propagating the primary infection appear to be e-mail or infected websites. E-mail was sent with sensational subject lines like "230 die as Storm batters Europe."
- Likewise, users were lured into downloading free but infected files from websites containing music of various pop artists.
- The primary infection instructed the victim to join the Storm hornet embedded in the Overnet P2P network.
- Once part of the botnet, the bat was programmed to receive the second and subsequent injections of malicious code. One of the injections instructed the bat to propagate e-mail viruses. Another injection received some days later instructed the bat to launch a DDoS attacks on a target specified by the botmaster.

WEB SECURITY

Web services

- W3c defines a web service as:
- A software system identified by a URI whose public interfaces and bindings are defined and described by XML.
- The entities involved are:
- Providers register or publish their services in a public registry.
- Requesters discover services by querying the registry for services that match certain criteria.
- Once a requester has identified a provider whose services it needs ,it binds to and invokes the service of that provider.

Entities involved in a web service

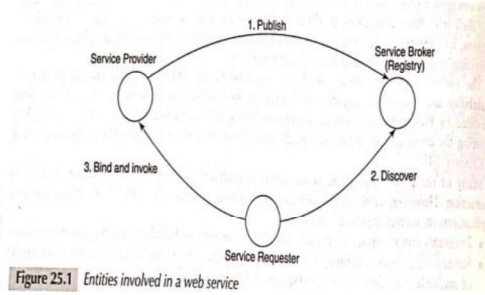


Figure 25.1 *Entities involved in a web service*

XML

- **XML is a meta markup language for text documents / textual data.**
- **XML allows to define languages („applications“) to represent text documents / textual data.**
- Easy to understand for human users
- Very expressive (semantics along with the data)
- Well structured, easy to read and write from programs

Possible Advantages of Using XML

- Truly Portable Data
- Easily readable by human users
- Very expressive (semantics near data)
- Very flexible and customizable (no finite tag set)
- Easy to use from programs (libs available)
- Easy to convert into other representations (XML transformation languages)
- Many additional standards and tools
- Widely used and supported

5

XML Documents

What's in an XML document?

- Elements
- Attributes
- plus some other details

6

Elements in XML Documents

- (Freely definable) **tags**: `article`, `title`, `author`
  - with start tag: `<article>` etc.
  - and end tag: `</article>` etc.
- **Elements**: `<article> ... </article>`
- Elements have a **name** (`article`) and a **content** (`...`)
- Elements may be nested.
- Elements may be empty: `<this_is_empty/>`
- Each XML document has exactly one root element and forms a tree.

7

XML by Example

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in 10 Years</title>
</article>
```

8

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

9

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Freely definable tags

10

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Start Tag

End Tag

Element

Content of the Element (Subelements and/or Text)

11

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Attributes with **name** and **value**

12

Elements vs. Attributes

Elements may have **attributes** (in the start tag) that have a **name** and a **value**, e.g. `<section number="1">`.

What is the difference between elements and attributes?

- Only one attribute with a given name per element (but an arbitrary number of subelements)
- Attributes have no structure, simply strings (while elements can have subelements)

As a *rule of thumb*:

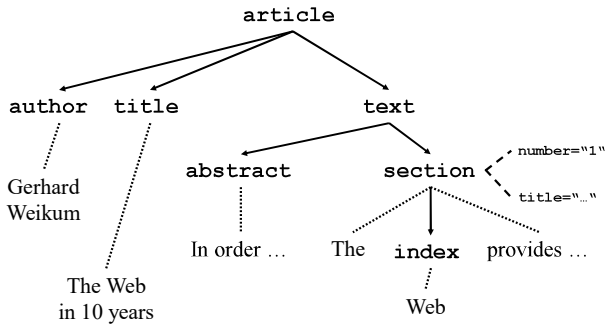
- Content into elements
- Metadata into attributes

Example:

```
<person born="1912-06-23" died="1954-06-07">
Alan Turing</person> proved that…
```

13

XML Documents as Ordered Trees



14

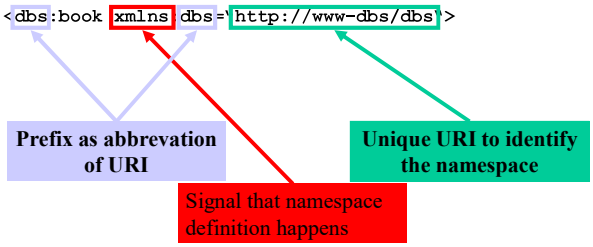
Well-Formed XML Documents

A **well-formed** document must adher to, among others, the following rules:

- Every start tag has a matching end tag.
- Elements may nest, but must not overlap.
- There must be exactly one root element.
- Attribute values must be quoted.
- An element may not have two attributes with the same name.
- Comments and processing instructions may not appear inside tags.

15

Namespace Syntax



16

3.2 XML Schema Basics

- XML Schema is an XML application
- Provides simple types (string, integer, dateTime, duration, language, …)
- Allows defining possible values for elements
- Allows defining types derived from existing types
- Allows defining complex types
- Allows posing constraints on the occurrence of elements
- Allows forcing uniqueness and foreign keys

A Simple XML Document

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Attributes with **name** and **value**

18

What is SOAP?

- SIMPLE OBJECT ACCESS PROTOCOL
- For exchanging structured information over internet.
- SOAP can be used over any transport protocol such as TCP, HTTP, SMTP
- SOAP defines a model for processing individual, one-way messages
- The mapping between soap message and an underlying transport protocol is referred as SOAP binding.
- Soap may run on top of http ot SMTP.

19

SOAP Message Format

- SOAP message consists of three parts:
  - SOAP Envelope
  - SOAP Header (optional)
  - SOAP Body

SOAP Header:

- The Header element is a generic container for control information
- Header blocks should contain information that influences payload processing
- Header is optional

SOAP Body:

- The Body element represents the message payload

20

```
<?xml version="1.0"?>
<soap:Envelope
 xmlns:soap = "http://www.w3.org/2001/12/soap-envelope . . .>
  <soap:Body xmlns:X="http://www.stockQuote.com/price">
    <X:GetPrice xmlns:X = "http://www. . . " >
      <X:StockName> MyStartUp </X:StockName>
    </X:GetPrice>
  </soap:Body>
</soap:Envelope>
```

(a) SOAP message in HTTP POST request


```
<?xml version="1.0"?>
<soap:Envelope
 xmlns:soap = "http://www.w3.org/2001/12/soap-envelope . . . >
  <soap:Body xmlns:X="http://www.stockQuote.com/price">
    <X:GetPriceResponse xmlns:X = "http://www. . . " >
      <X:Price> 3847 </X:Price>
    </X:GetPriceResponse>
  </soap:Body>
</soap:Envelope>
```

(b) SOAP message in HTTP response

WSDL

- Web services Definition langauge
- "An XML format for describing network services as a set
- Contains information where the service is located, what the service does, and how to invoke the service such as types ,messages,operation,port types,and bindings.
- Operation:abstract definition of a n action.
- Message: abstract definition of data beingexchanged as a part of operation.

WSDL

```
<message name="message1">
   <part name=" . . . " type=" . . . "/>
</message>

<message name="message2">
   <part name=" . . . " type=" . . . "/>
</message>

<portType name="portType1">
  <operation name=" . . . ">
      <input message="message1"/>
      <output message="message2"/>
  </operation>
</portType>
```

Port type that includes one operation comprising two messages

UDDI

- UNIVERSAL DESCRIPTION DISCOVERY AND INTEGRATION
- IS a registry or catalogue that allows businesses across the globe to list themselves in internet.
- Clients query the registry services
- Response :access to WSDL.

25

WS security

- Token Types
- XML Encryption
- XML signature

26

Token types

- Web security addresses basic problems in securing messages used in web services.
- Its main functions are:
  - It defines XML elements that are used to communicate *security tokens* (defined below) in the header of a SOAP message.
  - Together with the XML Encryption Standard it defines the syntax and processing rules used to *encrypt* one or more parts of a SOAP message.
  - Together with the XML Signature Standard, it defines the syntax and processing rules used to create and represent a *digital signature* on one or more parts of a SOAP message.

27

Token types

SHA-1 *(n, t, pw)*

The user name, hash, nonce, and timestamp are sent in a <UsernameToken>

```
< UsernameToken >
        < Username > John < /Username >
        < Password Type = "PasswordDigest" >
                4u%h&+q:L
        < /Password >
        < Nonce > . . . < /Nonce >
        < Created > . . . </Created >
< /UsernameToken>
```

28

Token types

```
< Security >
...
    < BinarySecurityToken
        ValueType = " ... X509v3"
        EncodingType = " ... Base64Binary" >
                Lp9tba4Pc7G ...
    < / BinarySecurityToken >
...
< /Security >
```

29

XML Encryption

- It defines XML elements for representing encrypted data and keys used for encryption.
- Encryption at different levels of granularity
- An entire document
- A complete XML element within a document
- Content of an XML element.

30

XML Encryption

- <EncryptedData> element contains:
  - A Type attribute – indicates the type of the information encrypted
  - Information about the algorithm used for encryption
  - <CipherData> A Reference to the cipher, or the cipher itself
  - <EncryptedKey> - used for encrypting pre shared key or session key.

31

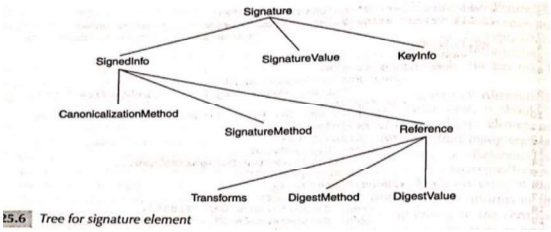
XML Encryption

```
< S11:Body wsu:Id = "#Id1" >
    <xenc:EncryptedData
    Type = "http://www.w3.org/2001/04/xmlenc#Element"
          Id="Id2">
      <xenc:EncryptionMethod
            Algorithm = "http://www.w3.org/xmlenc#aes256-cbc"/>
      <xenc:CipherData>
          <xenc:CipherValue>
                tdaqUsjXipJ09jlkjh5oinlkdsn...
          </xenc:CipherValue>
      </xenc:CipherData>
</xenc:EncryptedData>
```

32

XML signature

- Specifies syntax for signatures and signature keys.
- Canonical form: transformed to
- Xml Signature is included in header of soap message.



25.6 *Tree for signature element*

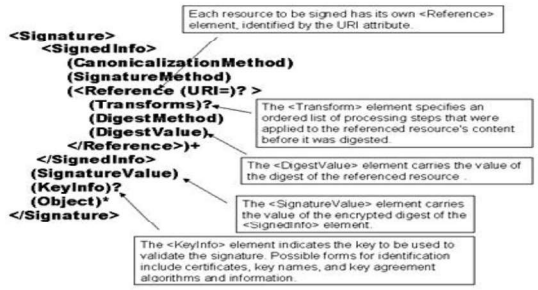
XML signature

- Signed Info

<SignedInfo>

Within this element is included information about the *canonicalization algorithm* and *signature algorithm* employed. An example of the signature algorithm is RSA-SHA1, i.e., the use of SHA-1 to perform the hash followed by an RSA private key operation on the hash value. A "signature" in the form of a Message Authentication Code (MAC) is also supported.

- Signature Value
  - Contains digital signature value.
  - On Entire signature element
- Key Info
  - the key required to verify digital signature at the receiver end.




```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod
      Algorithm="http://www.w3.org/2001/10/xmlexcc14n#" />
    <SignatureMethod
      Algorithm="http://www.w3.org/2000/xmldsig#rsasha1" />

    <Reference URI="#Id1">
      <Transforms>
        <TransformAlgorithm =
          "http://www.w3.org/2001/xmlexcc14n#" />
      </Transforms>
      <DigestMethodAlgorithm=
          "http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>  Yhsl . . . pKl  </DigestValue>
    </Reference>
```

SAML:security assertions markup language

- SAML is the XML based standard created to enable portable identities and the assertions these identities want to make.
- Defines a standard message exchange protocol.
  Specifies how it is transported.
- Single Sign-On (SSO)
- Portable Trust" - a user, whose identity is established and verified in one domain, can invoke services in another domain

37

SAML Assertions

- SAML defines three types of assertions.

1. Authentication
2. Authorization
3. Attributes

38

SAML

- Assertion is a claim, statement, or declaration of fact made by SAML authority
- Types of assertions:
- Authentication - the subject was authenticated by a particular means at a particular time
- Authorization - the subject was granted or denied access to a specified resource
- Attributes -the subject is associated with the supplied attribute.

39

Common Elements

- <Issuer> - the issuer name [Required]
- <ds:Signature> - an XML signature for integrity protection and authentication of the issuer [Optional]
- <Subject> - the subject of the statements in the assertion [Optional]
- <Conditions> - must be evaluated when using assertions [Optional]

40

SAML code

```
1   <saml:Assertion  xmlns:saml = . . .     Version  =  "2.0"
                                            IssueInstant  =  "2010-02-01T08:25:15Z">
2           <saml:Issuer  Format= . . . :entity>
                          http://www.admin.iitb.ac.in
3           </saml:Issuer>
4           <saml:Subject>
5                 <saml:NameID   Format = " . . . :emailAddress">
                          rajeshX@cse.iitb.ac.in
6                 </saml:NameID>
7           </saml:Subject>
8           <saml:Conditions
9                 NotBefore  =  "2010-02-01T08:26:00Z"
10                NotOnOrAfter  =  "2010-02-02T10:30:00Z"
11          </saml:Conditions>
12          <saml:AuthnStatement   AuthnInstant  =  "2010-02-01T08:25:15Z"
                                                    SessionIndex  =  "1234">
13                <saml:AuthnContext>
14                      <saml:AuthnContextClassRef>
                              . . . :PasswordProtectedTransport
15                      </saml:AuthnContextClassRef>
16                </saml:AuthnContext>
17          </saml:AuthnStatement>
18  </saml:Assertion>
```

41

Example of SAML

Now suppose that Sandeep is a gold customer of SmartTravels – a status conferred on all customers of SmartTravels who have done business in excess of Rs. 4,00,000 over the last 4 years. SmartTravels has business relationships with several airlines, including Jet Air, which provide varying discounts to all of its gold customers. How is Jet Air expected to know that Sandeep is a gold customer of SmartTravels and is eligible for the discounted price?

For the sake of completeness, we enumerate all the steps involved in Sandeep's transaction.

1. Sandeep logs in to the SmartTravels website and is *authenticated*. He indicates the destination city, date and time of travel, and price of ticket he is willing to pay.
2. SmartTravels determines that Sandeep is a gold customer and presents a list of airlines that satisfy Sandeep's requirements.
3. Sandeep clicks on the airline of interest, say JetAir.
4. SmartTravels creates *SAML assertions* indicating that
   a. Sandeep has been authenticated using a login name–password mechanism (authentication assertion)
   b. Sandeep is a gold customer (attribute assertion)
5. SmartTravels creates an HTML form with two hidden inputs. The first, named *SAMLResponse*, contains the signed SAML assertion. The second hidden input, called RelayState, contains the *URL of the resource* required by Sandeep. The relevant portion of the form is

42

Other standards

- WS trust
- WS security policy

43

WS trust

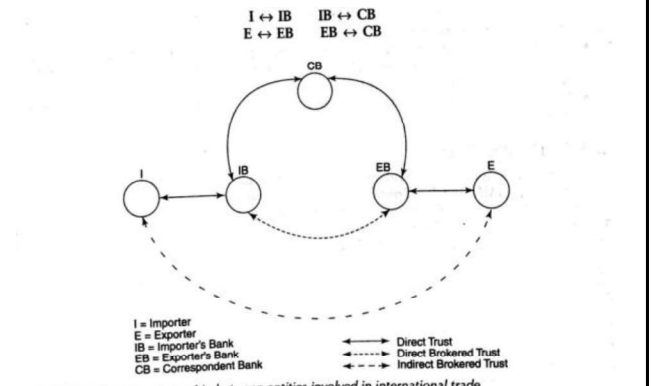


Figure 25.9 *Trust relationship between entities involved in international trade*

44

WS trust could be used in the following manner:

- I authenticates himself to IB.
- Since IB and CB trust each other, IB requests CB to issue a security token to be used by I.
- CB creates a security token and includes information such as the maximum credit amount extended to I. CB acts as a Security Token Service Provider. CB communicates this token to IB who forwards it to I.
- I dispatches the token to E.
- E requests EB to validate the token. EB may be able to validate the token on its own. If not, it sends it to CB for validation.
- The success or failure of the validation process is communicated by CB to E through EB.

45

Web security policy

- Web security policy enables a web sevice to specify the security tokens it will accept for authentication and access control.
- Returning to the example of Fig. 25.9, the policies regarding authentication laid out by relevant entities may be as follows:
- An importer must authenticate himself to his local bank via a login name and password.
- A local bank must authenticate itself to the global bank using a challenge-response protocol in conjunction with a digital certificate.
- An importer must authenticate himself to the given exporter through a SAML token signed by a global bank.

46

MODULE 5

27.1 IT ACT: AIM AND OBJECTIVES

Q.Explain the aims and objectives of IT act 2000.

- The information technology act ,2000 is an important law related to Indian cyber law.
- The act strives to achieve the following objectives:
    1. To give legal recognition to ***transactions*** done by electronic way or by use of the internet.
    2. To grant legal recognition to ***digital signature*** for accepting any agreement via computer.
    3. To provide facility of ***filling documents online.***
    4. To authorize any undertaking to ***store their data in electronic storage***.
    5. To prevent ***cyber crime*** by imposing high penalty for such crimes and protect privacy of internet users.
    6. To keep legal recognition for keeping ***books of account by bankers*** and undertaking in electronic form.

27.2 SCOPE OF THE ACT

- The act attempts to address the following issues:

    1.legal recognition of electronic documents.

    2. legal recognition of digital signatures.

    3. offences.

    4.justice dispensation for cyber crimes.

Q.Define the following terms wrt to IT act 2000(any terminologies can be asked)

27.3 MAJOR CONCEPTS

1. ***"Access"*** *:* means ***gaining entry into, instructing or communicating*** with the logical, arithmetical, or memory function resources of a computer, computer system or computer network;
2. ***"Addressee"*** :means a person who is intended by the originator to ***receive the electronic record*** but does not include any intermediary;
3. ***"Adjudicating officer" :*** means an adjudicating officer appointed .
4. ***"Affixing digital signature"*** *:* adoption of any methodology or procedure by a person for the purpose of authenticating an electronic record by means of digital signature;

5. ***"Appropriate Government"*** : means as respects any matter,—
   relating to any State law enacted under List III of the Seventh Schedule to the Constitution,
6. ***"Asymmetric crypto system"*** means a system of a ***secure key pair*** consisting of a private key for creating a digital signature and a public key to verify the digital signature;
7. ***"Certifying Authority"*** means a person who has been ***granted a license to issue a Digital Signature Certificate*** under section 24;
8. ***"Certification practice statement"*** means a statement issued by a Certifying Authority to specify the practices that the Certifying Authority employs in issuing Digital Signature Certificates;
9. ***"computer"*** means any electronic magnetic, optical or other high- speed data processing device or system which performs logical, arithmetic, and memory functions by manipulations of electronic, magnetic or optical impulses, and includes all input, output, processing, storage, computer software, or communication facilities which are connected or related to the computer in a computer system or computer network;

   10."Computer network" means the interconnection of one or more computers through—
   - the use of satellite, microwave, terrestrial line or other communication media; and
   - terminals or a complex consisting of two or more interconnected computers whether or not the interconnection is continuously maintained;

11. ***"computer system"*** means a device or collection of devices, including input and output support devices and excluding calculators which are not programmable and capable of being used in conjunction with external files, which contain computer programmes, electronic instructions, input data and output data, that performs logic, arithmetic, data storage and retrieval, communication control and other functions;

12."data" means a representation of information, knowledge, facts, concepts or instructions which are being prepared or have been prepared in a formalised manner, and is intended to be processed, is being processed or has been processed in a computer system or computer network, and may be in any form (including computer printouts magnetic or optical storage media, punched cards, punched tapes) or stored internally in the memory of the computer;

13."***digital signature"*** means ***authentication of any electronic record*** by a subscriber by means of an electronic method or procedure in accordance with the provisions of section 3;

14. ***"electronic form"*** with reference to information means any information generated, sent, received or stored in media, magnetic, optical, computer memory, micro film, computer generated micro fiche or similar device;

15. ***"Electronic Gazette"*** means the Official Gazette published in the electronic form;

1. ***electronic record"*** means data, record or data generated, image or sound stored, received or sent in an electronic form or micro film or computer generated micro fiche;

17. ***"information"*** includes data, text, images, sound, voice, codes, computer programmes, software and databases or micro film or computer generated micro fiche:

18. ***"intermediary"*** with respect to any particular electronic message means any person who on behalf of another person receives, stores or transmits that message or provides any service with respect to that message;

19. ***"key pair",*** in an asymmetric crypto system, means a private key and its mathematically related public key, which are so related that the public key can verify a digital signature created by the private key;

Q Explain any important provisions of IT act 2000.
Q Explain any important provisions of IT act 2000 with regard to:
1.digital signature,.
2. Legal recognition of electronic records.
3. Legal recognition of digital signatures

27.4 IMPORTANT PROVISIONS

27.4.1 DIGITAL SIGNATURE :AUTHENTICATION OF ELECTRONIC RECORDS.

1. Any subscriber may authenticate an electronic record by ***affixing his digital signature.***
2. The authentication of the electronic record shall be effected by the use of ***asymmetric crypto system and hash function*** which envelop and transform the initial electronic record into another electronic record.

   Explanation: "hash function" means an algorithm mapping or translation of one sequence of bits into another, generally smaller, set known'as ***"hash result"*** such that an electronic record yields the same hash result every time the algorithm is executed with the same electronic record as its input making it computationally infeasible—

   - ✓ to derive or reconstruct the original electronic record from the hash result produced by the algorithm;

- ✓ that two electronic records can produce the same hash result using the algorithm.

3. Any person by the use of a ***public key of the subscriber*** can verify the electronic record.
4. The private key and the ***public key are unique to the subscriber*** and constitute a functioning key pair.

27.4.2 ELECTRONIC GOVERNANCE : Legal recognition of electronic records.

Where any law provides that information or any other matter shall be in writing or in the typewritten or printed form, then, notwithstanding anything contained in such law, such requirement shall be deemed to have been satisfied if such information or matter is—

(a) rendered or made available in an electronic form; and

(b) accessible so as to be usable for a subsequent reference.

27.4.3 Legal recognition of digital signatures

- Where any law provides that information or any other matter shall be authenticated by affixing the signature or any document shall be signed or bear the signature of any person (then, notwithstanding anything contained in such law, such requirement shall be deemed to have been satisfied, if such information or matter is authenticated by means of digital signature affixed in such manner as may be prescribed by the Central Government.

27.4.4 Use of electronic records and digital signatures in Government and its agencies.

1. Where any law provides for—

- ✓ the filing of any form. application or any other document with any office, authority, body or agency owned or controlled by the appropriate Government in a particular manner;
- ✓ the issue or grant of any licence, permit, sanction or approval by whatever name called in a particular manner;
- ✓ the receipt or payment of money in a particular manner, then, notwithstanding anything contained in any other law for the time being in force, such requirement shall be deemed to have been satisfied if such filing, issue, grant, receipt or payment, as the case may be, is effected by means of such electronic form as may be prescribed by the appropriate

Government.

2. The appropriate Government may, for the purposes of sub-section (1), by rules, prescribe—
   - ✓ the manner and format in which such electronic records shall be filed, created or issued;
   - ✓ the manner or method of payment of any fee or charges for filing, creation or issue any electronic record under clause *(a)*.

27.4. 6 Retention of electronic records.

1. Where any law provides that documents, records or information shall be retained for any specific period, then, that requirement shall be deemed to have been satisfied if such documents, records or information are retained in the electronic form, if—
   (a) the information contained therein remains accessible so as to be usable for a subsequent reference;
   (b) the electronic record is retained in the format in which it was originally generated, sent or received or in a format which can be demonstrated to represent accurately the information originally generated, sent or received;
   (c) the details which will facilitate the identification of the origin, destination, date and time of despatch or receipt of such electronic record are available in the electronic record:

27.4.7 Publication of rule, regulation, etc., in Electronic Gazette.

- Where any law provides that any rule, regulation, order, bye-law, notification or any other matter shall be published in the Official Gazette, then, such requirement shall be deemed to have been satisfied if such rule, regulation, order, bye-law, notification or any other matter is published in the Official Gazette or Electronic Gazette:
- Provided that where any rule, regulation, order, bye-law, notification or any other matter is published in the Official Gazette or Electronic Gazette, the date of publication shall be deemed to be the date of the Gazette which was first published in any form.

27.4.7 Power to make rules by Central Government in respect of digital signature.

- The Central Government may, for the purposes of this Act, by rules, prescribe—
- the type of digital signature;

- the manner and format in which the digital signature shall be affixed;
- the manner or procedure which facilitates identification of the person affixing the digital signature;
- control processes and procedures to ensure adequate integrity, security and confidentiality of electronic records or payments; and
- any other matter which is necessary to give legal effect to digital sigtures.

Q. Explain Attribution,Acknowledgement, And Despatch Of Electronic Records.

27.5 ATTRIBUTION,ACKNOWLEDGEMENT, AND DESPATCH OF ELECTRONIC RECORDS

27.5.1 Attribution of electronic records.

An electronic record shall be attributed to the originator—

1. if it was sent by the ***originator*** himself;
2. by a person who had the ***authority to act on behalf of the originator*** in respect of that electronic record;
3. by an information system programmed by or on behalf of the originator to operate automatically.

27.5.2 Acknowledgment of receipt.

1. Where the originator has not agreed with the addressee that the acknowledgment of receipt of electronic record be given in a particular form or by a particular method, an acknowledgment may be given by—

   (a) any communication by the addressee, automated or otherwise; or

   (b) any conduct of the addressee, sufficient to indicate to the originator that the electronic record has been received.

2. Where the originator has stipulated that the electronic record shall be binding only on receipt of an acknowledgment of such electronic record by him, then unless acknowledgment has been so received, the electronic record shall be deemed to have been never sent by the originator.
3. Where the originator has not stipulated that the electronic record shall be binding only on receipt of such acknowledgment, and the acknowledgment has not been received by the originator within the time specified or agreed or, if no time has been specified or agreed to

within a reasonable time, then the originator may give notice to the addressee stating that no acknowledgment has been received by him and specifying a reasonable time by which the acknowledgment must be received by him and if no acknowledgment is received within the aforesaid time limit he may after giving notice to the addressee, treat the electronic record as though it has never been sent.

27.5 .3 Time and place of despatch and receipt of electronic record.

1. UNLESS as otherwise agreed to between the originator and the addressee, the dispatch of an electronic record occurs when it enters a computer resource outside the control of the originator.
2. UNLESS as otherwise agreed between the originator and the addressee, the time of receipt of an electronic record shall be determined as follows, namely :—
   - ✓ if the addressee has designated a computer resource for the purpose of receiving electronic records,—
   - ✓ receipt occurs at the time when the electronic, record enters the designated computer resource;
   - ✓ if the electronic record is sent to a computer resource of the addressee that is not the designated computer resource, receipt occurs at the time when the electronic record is retrieved by the addressee;
3. UNLESS as otherwise agreed to between the originator and the addressee, an electronic record is deemed to be dispatched at the place where the originator has his place of business, and is deemed to be received at the place where the addressee has his place of business.
4. if the originator or the addressee has more than one place of business, the principal place of business, shall be the place of business;
5. if the originator or the addressee does not have a place of business, his usual place of residence shall be deemed to be the place of business;
6. "usual place of residence", in relation to a body corporate, means the place where it is registered.

Q.Explain the importance of securing electronic records and digital signature(27.6.1,27.6.2)

Q.mention the security procedures followed with reference to securing electronic records and digital signature(27.6.3)

27. 6 SECURE ELECTRONIC RECORDS AND SECURE DIGITAL SIGNATURES

27.6.1 Secure electronic record.

- Where any security procedure has been applied to an electronic record at a specific point of time. then such record shall be deemed to be a secure electronic record from such point of time to the time of verification.

27.6.2 Secure digital signature.

- If, by application of a security procedure agreed to by the parties concerned, it can be verified that a digital signature, at the time it was affixed, was—
  - ✓ unique to the subscriber affixing it;
  - ✓ capable of identifying such subscriber;
  - ✓ created in a manner or using a means under the exclusive control of the subscriber and is linked to the electronic record to which it relates in such a manner that if the electronic record was altered the digital signature would be invalidated, then such digital signature shall be deemed to be a secure digital signature.

27.6.3 Security procedure.

- The Central Government for the purposes of this Act prescribe the security procedure having regard to commercial circumstances prevailing at the time when the procedure was used, including—
  - ✓ the nature of the transaction;
  - ✓ the level of sophistication of the parties with reference to their technological capacity;
  - ✓ the volume of similar transactions engaged in by other parties;
  - ✓ the availability of alternatives offered to but rejected by any party;
  - ✓ the cost of alternative procedures; and
  - ✓ the procedures in general use for similar types of transactions or communications.

27.7 REGULATION OF CERTIFYING AUTHORITIES

1. The Central Government may, by notification in the Official Gazette, appoint a Controller of Certifying Authorities for the purposes of this Act and may also by the same or subsequent notification appoint such number of Deputy Controllers and Assistant Controllers as it deems fit.
2. The Controller shall discharge his functions under this Act subject to the general control and directions of the Central Government.
3. The Deputy Controllers and Assistant Controllers shall perform the functions assigned to them by the Controller under the general superintendence and control of the Controller.
4. The qualifications, experience and terms and conditions of service of Controller, Deputy Controllers and Assistant Controllers shall be such as may be prescribed by the Central Government.
5. The Head Office and Branch Office of the office of the Controller shall be at such places as the Central Government may specify, and these may be established at such places as the Central Government may think fit.
6. There shall be a seal of the Office of the Controller.

Q. who is a controller ?outline his functions.(27.7.1)

Q. outline the controllers powers.(27.7.2 to 27.7. 12)

27.7.1 Functions of Controller.

- The Controller may perform all or any of the following functions, namely:—
    1. exercising supervision over the activities of the Certifying Authorities;
    2. certifying public keys of the Certifying Authorities;
    3. laying down the standards to be maintained by the Certifying Authorities;
    4. specifying the qualifications and experience which employees of the Certifying Authorities should possess;
    5. specifying the conditions subject to which the Certifying Authorities shall conduct their business;
    6. specifying the contents of written, printed or visual materials and advertisements that may be distributed or used in respect of a Digital Signature Certificate and

the public key;

7. specifying the form and content of a Digital Signature Certificate and the key,
8. specifying the form and manner in which accounts shall be maintained by the Certifying Authorities;
9. specifying the terms and conditions subject to which auditors may be appointed and the remuneration to be paid to them;
10. facilitating the establishment of any electronic system by a Certifying Authority either solely or jointly with other Certifying Authorities and regulation of such systems;
11. specifying the manner in which the Certifying Authorities shall conduct their dealings with the subscribers;
12. resolving any conflict of interests between the Certifying Authorities and the subscribers;
13. laying down the duties of the Certifying Authorities;
14. maintaining a data base containing the disclosure record of every Certifying Authority containing such particulars as may be specified by regulations, which shall be accessible to public.

27.7.2 Recognition of foreign Certifying Authorities.

1. the Controller may with the previous approval of the Central Government, and by notification in the Official Gazette, recognise any foreign Certifying Authority as a Certifying Authority for the purposes of this Act.
2. Where any Certifying Authority is recognised under sub-section (1), the Digital Signature Certificate issued by such Certifying Authority shall be valid for the purposes of this Act.
3. The Controller may, if he is satisfied that any Certifying Authority has contravened any of the conditions and restrictions subject to which it was granted recognition under sub-section (1) he may, for reasons to be recorded in writing, by notification in the Official Gazette, revoke such recognition.

27.7.3 Controller to act as repository.

1. The **Controller** shall be the repository of all Digital Signature Certificates issued under this Act.

2. The Controller shall—
   - ✓ make use of hardware, software and procedures that are secure intrusion and misuse;
   - ✓ observe such other standards as may be prescribed by the Central Government,

   to ensure that the secrecy and security of the digital signatures are assured.

3. The Controller shall maintain a computerised data base of all public keys in such a manner that such data base and the public keys are available to any member of the public.

27.7.4 Licence to issue Digital Signature Certificates.

1. Any person may make an application, to the Controller, for a licence to issue Digital Signature Certificates.
2. No licence shall be issued under sub-section (1), unless the applicant fulfills such requirements with respect to qualification, expertise, manpower, financial resources and other infrastructure facilities, which are necessary to issue Digital Signature Certificates as may be prescribed by the Central Government
3. A licence granted under this section shall—

   (a) be valid for such period as may be prescribed by the Central Government;

   (b) not be transferable or heritable;

   (c) be subject to such terms and conditions as may be specified by the regulations.

27.7.5 Application for licence

1. Every application for issue of a licence shall be in such form as may be prescribed by the Central Government.
2. Every application for issue of a licence shall be accompanied by—
   - ✓ a certification practice statement;
   - ✓ a statement including the procedures with respect to identification of the applicant;
   - ✓ payment of such fees, not exceeding twenty-five thousand rupees as may be prescribed by the Central Government;
   - ✓ such other documents, as may be prescribed by the Central Government.

27.7.6 Renewal of licence.

- An application for renewal of a licence shall be—
- ✓ in the required form;
- ✓ accompanied by such fees, not exceeding five thousand rupees, as may be prescribed by the Central Government and shall be made not less than forty-five days before the date of expiry of the period of validity of the licence.

27.7.7 Procedure for grant or rejection of licence.

- The Controller may, on receipt of an application under sub-section (1) of section 21, after considering the documents accompanying the application and such other factors, as he deems fit, grant the licence or reject the application:

27.7.8 Suspension of licence.

- The Controller may, if he is satisfied after making such inquiry, as he may think fit, that a Certifying Authority has,—

  (a) made a statement in, or in relation to, the application for the issue or renewal of the licence, which is incorrect or false in material particulars;

  (b) failed to comply with the terms and conditions subject to which the licence was granted;

27.7.9 Notice of suspension or revocation of licence.

1. Where the licence of the Certifying Authority is suspended or revoked, the Controller shall publish notice of such suspension or revocation, as the case may be, in the database maintained by him.
2. Where one or more repositories are specified, the Controller shall publish notices of such suspension or revocation, as the case may be, in all such repositories:
3. Provided that the data base containing the notice of such suspension or revocation, as the case may be, shall be made available through a web site which shall be accessible round the clock:

27.7.10 Power to delegate.

- The Controller may, in writing, authorise the Deputy Controller, Assistant Controller or

any officer to exercise any of the powers of the Controller under this Chapter.

27.7.11 Power to investigate contraventions.

1. The Controller or any officer authorized by him in this behalf shall take up for investigation any contravention of the provisions of this Act, rules or regulations made thereunder.
2. The Controller or any officer authorized by him in this behalf shall exercise the like powers which are conferred on Income-tax authorities under Chapter XIII of the Income-tax Act, 1961 and shall exercise such powers, subject to such limitations laid down under that Act.

27.7.12 Access to computers and data.

1. Without prejudice to the provisions of sub-section (1) of section 69, the Controller or any person authorized by him shall, if he has reasonable cause to suspect that any contravention of the provisions of this Act, rules or regulations made there under has been committed, have access to any computer system, any apparatus, data or any other material connected with such system, for the purpose of searching or causing a search to be made for obtaining any information or data contained in or available to such computer system.
2. For the purposes of sub-section (1), the Controller or any person authorised by him may, by order, direct any person incharge of, or otherwise concerned with the operation of, the computer system, data apparatus or material, to provide him with such reasonable technical and other assistance as he may consider necessary.

27.7.13 Certifying Authority to follow certain procedures.

Every Certifying Authority shall, —

(a) make use of hardware, software and procedures that are secure from intrusion and misuse;

(b) provide a reasonable level of reliability in its services which are reasonably suited to the performance of intended functions;

(c) adhere to security procedures to ensure that the secrecy and privacy of the digital signatures are assured; and

(d) observe such other standards as may be specified by regulations.

27.7.14 Certifying Authority to ensure compliance of the Act, etc.

- ✓ Every Certifying Authority shall ensure that every person employed or otherwise engaged by it complies, in the course of his employment or engagement, with the provisions of this

Act, rules, regulations and orders made thereunder.

27.7.15 Display of licence.

- ✓ Every Certifying Authority shall display its licence at a conspicuous place of the premises in which it carries on its business.

27.7.16 Surrender of licence.

1. Every Certifying Authority whose licence is suspended or revoked shall immediately after such suspension or revocation, surrender the licence to the Controller.
2. Where any Certifying Authority fails to surrender a licence under sub-section (1), the person in whose favour a licence is issued, shall be guilty of an offence and shall be punished with imprisonment which may extend up to six months or a fine which may extend up to ten thousand rupees or with both.

27.7. 17 Disclosure.

1. Every Certifying Authority shall disclose in the manner specified by regulations—
   - ✓ its Digital Signature Certificate which contains the public key corresponding to the private key used by that Certifying Authority to digitally sign another Digital Signature Certificate;
   - ✓ any certification practice statement relevant thereto;
   - ✓ notice of the revocation or suspension of its Certifying Authority certificate, if any; and
   - ✓ any other fact that materially and adversely affects either the reliability of a Digital Signature Certificate, which that Authority has issued, or the Authority's ability to perform its services.
2. Where in the opinion of the Certifying Authority any event has occurred or any situation has arisen which may materially and adversely affect the integrity of its computer system or the conditions subject to which a Digital Signature Certificate was granted, then, the Certifying Authority shall—

   (a) use reasonable efforts to notify any person who is likely to be affected by that occurrence; or

   (b) act in accordance with the procedure specified in its certification practice

statement to deal with such event or situation.

Q Describe the role of certifying authority with regard to issuing digital certificate and Representation upon issuance,suspension and revocation .(27.8)

27.8 DIGITAL SIGNATURE CERTIFICATE

27.8.1 Certifying Authority to issue Digital Signature Certificate.

1. Any person may make an application to the Certifying Authority for the issue of a Digital Signature Certificate in such form as may be prescribed by the Central Government
2. Every such application shall be accompanied by such fee not exceeding twenty- five thousand rupees as may be prescribed by the Central Government, to be paid to the Certifying Authority:
3. Every such application shall be accompanied by a certification practice statement or where there is no such statement, a statement containing such particulars, as may be specified by regulations.
4. On receipt of an application under sub-section (1), the Certifying Authority may, after consideration of the certification practice statement or the other statement under sub- section (3)
5. and after making such enquiries as it may deem fit, grant the Digital Signature Certificate or for reasons to be recorded in writing, reject the application:

   Provided that no Digital Signature Certificate shall be granted unless the Certifying Authority is satisfied that—

   - ✓ the applicant holds the private key corresponding to the public key to be listed in the Digital Signature Certificate;
   - ✓ the applicant holds a private key, which is capable of creating a digital signature;
   - ✓ the public key to be listed in the certificate can be used to verify a digital signature affixed by the private key held by the applicant:

27.8.2 Representations upon issuance of Digital Signature Certificate.

- A Certifying Authority while issuing a Digital Signature Certificate shall certify that--
  1. it has complied with the provisions of this Act and the rules and regulations made thereunder,

2. it has published the Digital Signature Certificate or otherwise made it available to such person relying on it and the subscriber has accepted it;
3. the subscriber holds the private key corresponding to the public key, listed in the Digital Signature Certificate;
4. the subscriber's public key and private key constitute a functioning key pair,
5. the information contained in the Digital Signature Certificate is accurate; and
6. it has no knowledge of any material fact, which if it had been included in the Digital Signature Certificate would adversely affect the reliability of the representations made in clauses (a) to (d).

27.8.3 Suspension of Digital Signature Certificate.

1. Subject to the provisions of sub-section (2), the Certifying Authority which has issued a Digital Signature Certificate may suspend such Digital Signature Certificate,—

(a) on receipt of a request to that effect from—

I. the subscriber listed in toe Digital Signature Certificate; or

II. any person duly authorised to act on behalf of that subscriber,

2. if it is of opinion that the Digital Signature Certificate should be suspended in public interest
3. A Digital Signature Certificate shall not be suspended for a period exceeding fifteen days unless the subscriber has been given an opportunity of being heard in the matter.
4. On suspension of a Digital Signature Certificate under this section, the Certifying Authority shall communicate the same to the subscriber.

27.8.3 Revocation of Digital Signature Certificate.

1. A Certifying Authority may revoke a Digital Signature Certificate issued by it—
   (a) where the subscriber or any other person authorised by him makes a request to that effect;
   (b) upon the death of the subscriber, or
   (c) upon the dissolution of the firm or winding up of the company where the

subscriber is a firm or a company.

2. Subject to the provisions of sub-section (3) and without prejudice to the provisions of sub-section (1), a Certifying Authority may revoke a Digital Signature Certificate which has been issued by it at any time, if it is of opinion that—
   (a) a material fact represented in the Digital Signature Certificate is false or has been concealed;
   (b) a requirement for issuance of the Digital Signature Certificate was not satisfied;
   (c) the Certifying Authority's private key or security system was compromised in a manner materially affecting the Digital Signature Certificate's reliability;
   (d) the subscriber has been declared insolvent **or** dead or where a subscriber is a firm or a company, which has been dissolved, wound-up **or** otherwise ceased to exist
3. A Digital Signature Certificate shall not be revoked unless the subscriber has been given an opportunity of being heard in the matter.
4. On revocation of a Digital Signature Certificate under this section, the Certifying Authority

27.8.5 Notice of suspension or revocation.

1. Where a Digital Signature Certificate is suspended or revoked under section 37 or section 38, the Certifying Authority shall publish a notice of such suspension or revocation, as the case may be, in the repository specified in the Digital Signature Certificate for publication of such notice.
2. Where one or more repositories are specified, the Certifying Authority shall publish notices of such suspension or revocation, as the case may he. in all such repositories.

Q. ***Describe the duties of subscriber under the section 40, 41, and 42 of IT act 2000(answer:27.9)***

27.9 DUTIES OF SUBSCRIBERS

27.9.1 Generating key pair.

- Where any Digital Signature Certificate, the public key of which corresponds to the private key of that subscriber which is to be listed in the Digital Signature Certificate has been accepted by a subscriber, then, the subscriber shall generate the key pair by applying the security procedure.

27.9.2 Acceptance of Digital Signature Certificate

1. A subscriber shall be deemed to have accepted a Digital Signature Certificate if he publishes or authorises the publication of a Digital Signature Certificate—

   (a) to one or more persons;

   (b) in a repository, or otherwise demonstrates his approval of the Digital Signature Certificate in any manner.

2. By accepting a Digital Signature Certificate the subscriber certifies to all who reasonably rely on the information contained in the Digital Signature Certificate that—

   (a) the subscriber holds the private key corresponding to the public key listed in the Digital Signature Certificate and is entitled to hold the same;

   (b) all representations made by the subscriber to the Certifying Authority and all material relevant to the information contained in the Digital Signature Certificate are true;

   (c) all information in the Digital Signature Certificate that is within the knowledge of the subscriber is true.

27.9.3 Control of private key.

1. Every subscriber shall exercise reasonable care to retain control of the private key corresponding to the public key listed in his Digital Signature Certificate and take all steps to prevent its disclosure to a person not authorised to affix the digital signature of the subscriber.
2. If the private key corresponding to the public key listed in the Digital Signature Certificate has been compromised, then, the subscriber shall communicate the same without any delay to the Certifying Authority in such manner as may be specified by .the regulations.

Discuss the penalties and adjudication under section 43 IT act 2000 for

- ***a) Damage to computer, computer system***
- ***b) Failure to protect data.***
- ***c) Failure to furnish information return***

27.10 PENALTIES AND ADJUD1CATION

27.10.1 Penalty for damage to computer, computer system, etc.

- If any person without permission of the owner or any other person who is incharge of a computer, computer system or computer network, —

1. accesses or secures access to such computer, computer system or computer network;
2. downloads, copies or extracts any data, computer data base or information from such computer, computer system or computer network including information or data held or stored in any removable storage medium;
3. introduces or causes to be introduced any computer contaminant or computer virus into any computer, computer system or computer network;
4. damages or causes to be damaged any computer, computer system or computer network, data, computer data base or any other programmes residing in such computer, computer system or computer network;
5. disrupts or causes disruption of any computer, computer system or computer network;
6. denies or causes the denial of access to any person authorised to access any computer, computer system or computer network by any means;
7. provides any assistance to any person to facilitate access to a computer, computer system or computer network in contravention of the provisions of this Act, rules or regulations made thereunder;
8. charges the services availed of by a person to the account of another person by tampering with or manipulating any computer, computer system, or computer network,

1. he shall be liable to pay damages by way of compensation not exceeding one crore rupees to the person so affected.

- ✓ **"computer contaminant" means any set of computer instructions that are designed— to modify, destroy, record, transmit data or programme residing within a computer, computer system or computer network; orby any means to usurp the normal operation of the computer, computer system, or computer network;**
- ✓ "computer data base" means a representation of information, knowledge, facts, concepts or instructions in text, image, audio, video that are being prepared or have been prepared in a formalised manner or have been produced by a computer, computer system or computer network and are intended for use in a computer, computer system or computer network;
- ✓ "computer virus" means any computer instruction, information, data or programme that

destroys, damages, degrades or adversely affects the performance of a computer resource or attaches itself to another computer resource and operates when a programme, data or instruction is executed or some other event takes place in that computer resource;

- ✓ "damage" means to destroy, alter, delete, add, modify or rearrange any computer resource by any means.

27.10.2 compensation for failure to protect data

- ✓ If a corporate handling any sensitive information in a computer resource owns, controls or operates in negligent in maintaining security which causes gain to other person.in such case the corporate shall be liable to pay damages to the aggrieved party.

27.10.3Penalty for failure to furnish information return, etc.

- ➢ If any person who is required under this Act or any rules or regulations made thereunder to—

1. furnish any document, return or report to the Controller or ?he Certifying Authority fails to furnish the same, he shall be liable to a penalty not exceeding one lakh and fifty thousand rupees for each such failure;
2. file any return or furnish any information, books or other documents within the time specified therefor in the regulations fails to file return or furnish the same within the time specified therefor in the regulations, he shall be liable to a penalty not exceeding five thousand rupees for every day during which such failure continues
3. maintain books of account or records, fails to maintain the same, he shall be liable to a penalty not exceeding ten thousand rupees for every day during which the failure continues.

27.10.2 Residuary penalty.

- ✓ The contravention of which no penalty has been separately provided, shall be liable to pay a compensation not exceeding twenty-five thousand rupees to the person affected by such contravention or a penalty not exceeding twenty-five thousand rupees.

27.10.4Power to adjudicate.

1. For the purpose of adjudging under this Chapter whether any person has committed a contravention of any of the provisions of this Act or of any rule, regulation, direction or order made thereunder the Central Government shall, subject to the provisions of sub-section (3), appoint any officer not below the rank of a Director to the Government of

India or an equivalent officer of a State Government to be an adjudicating officer'for holding an inquiry in the manner prescribed by the Central Government.

2. The adjudicating officer shall, after giving the person referred to in sub-section
    - a reasonable opportunity for making representation in the matter and if, on such inquiry, he is satisfied that the person has committed the contravention, he may impose such penalty or award such compensation as he thinks fit in accordance with the provisions of that section.
3. No person shall be appointed as an adjudicating officer unless he possesses such experience in the field of Information Technology and legal or judicial experience as may be prescribed by the Central Government.
4. Where more than one adjudicating officers are appointed, the Central Government shall specify by order the matters and places with respect to which such officers shall exercise their jurisdiction
5. Every adjudicating officer shall have the powers of a civil court which are conferred oh the Cyber Appellate Tribunal under sub-section (2) of section 58, and—

(a) all proceedings before it shall be deemed to be judicial proceedings within the meaning of sections 193 and 228 of the Indian Penal Code;

(b) shall be deemed to be a civil court for the purposes of sections 345 and 346 of the Code of Criminal Procedure, 1973.

27.10.6 Factors to be taken into account by the adjudicating officer.

✓ While adjudging the quantum of compensation under this Chapter, the adjudicating officer shall have due regard to the following factors, namely:—

1. the amount of gain of unfair advantage, wherever quantifiable, made as a result of the default
2. the amount of loss caused to any person as a result of the default;
3. the repetitive nature of the default.

27.11 THE CYBER REGULATIONS APPELLATE TRIBUNAL

27.11.1 Establishment of Cyber Appellate Tribunal.

1. The Central Government shall, by notification, establish one or more appellate tribunals to be known as the Cyber Regulations Appellate Tribunal.
2. The Central Government shall also specify, in the notification referred to in sub- section (1), the matters and places in relation to which the Cyber Appellate Tribunal may exercise jurisdiction.

27.11.2 Composition of Cyber Appellate Tribunal.

- ✓ A Cyber Appellate Tribunal shall consist of one person only (hereinafter referred to as the Residing Officer of the Cyber Appellate Tribunal) to be appointed, by notification, by the Central Government

27.11.3 Qualifications for appointment as Presiding Officer of the Cyber Appellate Tribunal.

- ✓ A person shall not be qualified for appointment as the Presiding Officer of a Cyber Appellate Tribunal unless he—

1. is, or has been. or is qualified to be, a Judge of a High Court; or
2. is or has been a member of the Indian Legal Service and is holding or has held a post in Grade I of that Service for at least three years.

27.11.4 Term of office

- ✓ The Presiding Officer of a Cyber Appellate Tribunal shall hold office for a term of five years from the date on which he enters upon his office or until he attains the age of sixty-five years, whichever is earlier.

27.11.5Salary, allowances and other terms and conditions of service of Presiding Officer.

- ✓ The salary and allowances payable to, and the other terms and conditions of service including pension, gratuity and other retirement benefits of. the Presiding Officer of a Cyber Appellate Tribunal shall be such as may be prescribed.

27.11.6 Filling up of vacancies.

- ✓ If, for reason other than temporary absence, any vacancy occurs in the office n the Presiding Officer of a Cyber Appellate Tribunal, then the Central Government shall appoint another person in accordance with the provisions of this Act to fill the vacancy and the proceedings may be continued before the Cyber Appellate Tribunal from the stage at which the vacancy is filled.

27.11.7 Resignation and removal.

1. The Presiding Officer of a Cyber Appellate Tribunal may, by notice in writing under his hand addressed to the Central Government, resign his office:
   Provided that the said Presiding Officer shall, unless he is permitted by the Central Government to relinquish his office sooner, continue to hold office until the expiry of three months from the date of receipt of such notice or until a person duly appointed as his successor enters upon his office or until the expiry of his term of office, whichever is the earliest.
2. The Presiding Officer of a Cyber Appellate Tribunal shall not be removed from his office except by an order by the Central Government on the ground of proved misbehaviour or incapacity after an inquiry made by a Judge of the Supreme Court in which the Presiding Officer concerned has been informed of the charges against him and given a reasonable opportunity of being heard in respect of these charges.
3. The Central Government may, by rules, regulate the procedure for the investigation of misbehaviour or incapacity of the aforesaid Presiding Officer.

27.11.8 Orders constituting Appellate Tribunal to be final and not to invalidate its proceedings.

- ✓ No order of the Central Government appointing any person as the Presiding Officer of a Cyber Appellate Tribunal shall be called in question in any manner and no act or proceeding before a Cyber Appellate Tribunal shall be called in question in any manner on the ground merely of any defect in the constitution of a Cyber Appellate Tribunal.

27.11.9 Staff of the Cyber Appellate Tribunal.

1. The Central Government shall provide the Cyber Appellate Tribunal with such officers and employees as that Government may think fit

2. The officers and employees of the Cyber Appellate Tribunal shall discharge their functions under general superintendence of the Presiding Officer.
3. The salaries, allowances and other conditions of service of the officers and employees or' the Cyber Appellate Tribunal shall be such as may be prescribed by the Central Government.

27.11.10 Appeal to Cyber Appellate Tribunal.

1. Save as provided in sub-section (2), any person aggrieved by an order made by Controller or an adjudicating officer under this Act may prefer an appeal to a Cyber Appellate Tribunal having jurisdiction in the matter.
2. No appeal shall lie to the Cyber Appellate Tribunal from an order made by an adjudicating officer with the consent of the parties.
3. Every appeal under sub-section (1) shall be filed within a period of tony-five days from the date on which a copy of the order made by the Controller or the adjudicating officer is received by the person aggrieved and it shall be in such form and be accompanied by such fee as may be prescribed:

✓ Provided that the Cyber Appellate Tribunal may entertain an appeal after the expiry of the said period of tony-five days if it is satisfied that there was sufficient cause for not filing it within that period

4. On receipt of an appeal under sub-section (1), the Cyber Appellate Tribunal may, after giving the parties to the appeal, an opportunity of being heard, pass such orders thereon as it thinks fit, confirming, modifying or setting aside the order appealed against.
5. The Cyber Appellate Tribunal shall send a copy of every order made by it to" the parties to the appeal and to the concerned Controller or adjudicating officer
6. The appeal filed before the Cyber Appellate Tribunal under sub-section (1) shall be dealt with by it as expeditiously as possible and endeavour shall be made by it to dispose of the appeal finally within six months from the date of receipt of the appeal.

2711.11.Procedure and powers of the Cyber Appellate Tribunal.

(1) The Cyber Appellate Tribunal shall not be bound by the procedure laid down by the Code of civil Procedure, 1908 but shall be guided by the principles of natural justice and, subject to the other provisions of this Act and of any rules, the Cyber Appellate Tribunal shall have powers to regulate its own procedure including the place at which it shall have its sittings.

(2) The Cyber Appellate Tribunal shall have, for the purposes of discharging its functions under this Act, the same powers as are vested in a civil court under the Code of Civil Procedure, 1908, while trying a suit, in respect of the following matters, namely:—

1. summoning and enforcing the attendance of any person and examining him on oath;
2. requiring the discovery and production of documents or other electronic records
3. receiving evidence on affidavits;
4. issuing commissions for the examination of witnesses or documents;
5. reviewing its decisions;
6. dismissing an application for default or deciding it *ex pane;*
7. any other matter which may be prescribed.

(3). Every proceeding before the Cyber Appellate Tribunal shall be deemed to be a judicial proceeding within the meaning of sections 193 and 228, and for the purposes of section 196 of the Indian Penal Code and the Cyber Appellate Tribunal shall be deemed to be a civil court for the purposes of section 195 and Chapter XXVI of the Code of Criminal Procedure, 1973.

27.11.12Right to legal representation.

- ✓ The appellant may either appear in person or authorise one or more legal practitioners or any of its officers to present his or its case before the Cyber Appellate Tribunal.

27.11.13 Limitation.

- ✓ The provisions of the Limitation Act, 1963, shall, as far as may be, apply to an appeal made to the Cyber Appellate Tribunal.

27.11.13 Civil court not to have jurisdiction.

- ✓ No court shall have jurisdiction to entertain any suit or proceeding in respect of any matter which an adjudicating officer appointed under this Act or the Cyber Appellate Tribunal constituted under this Act is empowered by or under this Act to determine and no injunction shall be granted by any court or other authority in respect of any action taken or to be taken in pursuance of any power conferred by or under this Act.

27.11.15 Appeal to High Court.

- ✓ Any person aggrieved by any decision or order of the Cyber Appellate Tribunal may file an appeal to the High Court within sixty days from the date of communication of the

decision or order of the Cyber Appellate Tribunal to him on any question of fact or law arising out of such order

- ✓ Provided that the High Court may, if it is satisfied that the appellant was prevented by sufficient cause from filing the appeal within the said period, allow it to be filed within a further period not exceeding sixty days.

27.11.16 Compounding of contraventions.

(1) Any contravention under this Chapter may, either before or after the institution of adjudication proceedings, be compounded by the Controller or such other officer as may be specially authorised by him in this behalf or by the adjudicating officer, as the case may be, subject to such conditions as the Controller or such other officer or the adjudicating officer may specify:

27.11.17 Recovery of penalty

- ✓ A penalty imposed under this Act, if it is not paid, shall be recovered as an arrear of land revenue and the licence or the Digital Signature Certificate, as the case may be, shall be suspended till the penalty is paid.

Q. Explain the offences and punishments ,penalties for offences under the IT act 2000

27.12 OFFENCES

27.12.1 Tampering with computer source documents.

- ➢ Whoever knowingly or intentionally conceals, destroys or alters or intentionally or knowingly causes another to conceal, destroy or alter any computer source code used for a computer, computer programme, computer system or computer network, when the computer source code is required to be kept or maintained by law for the time being in force, shall be punishable with imprisonment up to three years, or with fine which may extend up to two lakh rupees, or with both.

- ➢ Explanation.— "computer source code" means the listing of programmes, computer commands, design and layout and programme analysis of computer resource in any form.

27.12.2Hacking with computer system.

- ➢ Whoever with the intent to cause or knowing that he is likely to cause wrongful loss or

damage to the public or any person destroys or deletes or alters any information residing in a computer resource or diminishes its value or utility or affects it injuriously by any means, commits hack

- Whoever commits hacking shall be punished with imprisonment up to three years, or with fine which may extend up to two lakh rupees, or with both.

27.12.3punishment for receiving stolen computer resource or communication device

- ✓ Whoever dishonestly received or retains any stolen computer resource of communication device knowing 'on device or having reason to believe the same to be stolen computer resource or communication three ice, shall be punished with imprisonment of either description for a term which may extend to years or with fine which may extend to rupees one lakh or with both [Section 66B].

27.12.4 punishment for identity theft

- ✓ Whoever fraudulently or dishonestly make use of the electronic signature, password or any unique identification feature of any other person, shall be punished with imprisonment of either description for a term which may extend to three years and shall also be liable to fine which may extend to rupees one lakh [Section 66B].

27.12.5 Punishment for cheating by personation by using computer resource

- ✓ Whoever, by means for any communication device or computer resource cheats by personating, shall be punished with imprisonment of either description for a term which may extend to three years and shall also be liable to fine which may extend to one lakh rupees [Section 66D].

27.12.6Punishment for violation of privacy

- ✓ Whoever, intentionally or knowingly captures, publishes or transmits the image of a private area of any person without his or her consent, under circumstances violating the privacy of that person, shall be punished with imprisonment which may extend to three years or with fine not exceeding two lakh rupees, or with both [Section 66E].

27.12.7Punishment for cyber terrorism

1. Whoever,

(a) With intent to threaten the unity, integrity, security of sovereignty of India or to strike terror in the people or any section of the people by-

(i) denying or cause the denial of access to any person authorized to access computer resource; or

(ii) attempting to penetrate or access a computer resource without authorization or exceeding authorized access; or

(iii) introducing or causing to introduce any computer contaminant, and by means of such conduct causes or is likely to cause death or injuries to persons or damage to or destruction of property or disrupts or knowing that it is likely to cause damage or disruption of supplies or services essential to the life of the community or adversely affect the critical information infrastructure specified under Section 70; or

(b) knowingly or intentionally penetrates or accesses a computer resource without authorization or exceeding authorized access, and by means of such conduct obtains access to information, data or computer database that is restricted; or any restricted information, data or computer database, with reasons to believe that such information, data or computer database so obtained may be used to cause or likely to cause injury to the interests of the sovereignty and integrity of India, the security of the State, friendly relations with foreign States, public order, decency or morality, or in relation to contempt of court, defamation or incitement to an offence, or to the advantage of any foreign nation, group of individuals, or otherwise, commits the offence of cyber terrorism.

27.12.7Publishing of information which is obscene in electronic form.

- Whoever publishes or transmits or causes to be published in the electronic form, any material which is lascivious or appeals to the prurient interest or if its effect is such as to tend to deprave and corrupt persons who are likely, having regard to all relevant circumstances, to read, see or hear the matter contained or embodied in it, shall be punished on first conviction with imprisonment of either description for a term which may extend to five years and with fine which may extend to one lakh rupees and in the event of a second or subsequent conviction with imprisonment of either description for a term which may extend to ten years and also with fine which may extend to two lakh rupees.

27.12.9Power of Controller to give directions.

(l) The Controller may, by order, direct a Certifying Authority or any employee of such Authority to take such measures or cease carrying on such activities as specified in the order if those are necessary to ensure compliance with the provisions of this Act, rules or any regulations made thereunder.

(2) Any person who fails to comply with any order under sub-section (1) shall be guilty of an offence and shall be liable on conviction to imprisonment for a term not exceeding three years or to a Fine not exceeding two lakh rupees or to both.

27.12.11Protected system.

(1) The appropriate Government may, by notification in the Official Gazette, declare that any computer, computer system or computer network to be a protected system.

(2) The appropriate Government may, by order in writing, authorise the persons who are authorized to access protected systems notified under sub-section (1).

(3) Any person who secures access or attempts to secure access to a protected system in contravention of the provisions of this section shall be punished with imprisonment of either description for a term which may extend to ten years and shall also be liable to fine.

27.12.12Penalty for misrepresentation.

- ✓ Whoever makes any misrepresentation to, or suppresses any material fact from, the Controller or the Certifying Authority for obtaining any licence or Digital Signature Certificate, as the case may be. shall be punished with imprisonment for a term which may extend to two years, or with fine which may extend to one lakh rupees, or with both.

27.12.13Penalty for breach of confidentiality and privacy.

- ✓ Save as otherwise provided in this Act or any other law for the time being in force, any person who, in pursuance of any of the powers conferred under this Act, rules or regulations made thereunder, has secured access to any electronic record, book, register, correspondence, information, document or other material without the consent of the person concerned discloses such electronic record, book. register, correspondence, information, document or other material to any other person shall be punished with imprisonment for a term which may extend to two years, or with fine which may extend to one lakh rupees, or with both.

27.12.14 Penalty for publishing Digital Signature Certificate false in certain particulars.

(1) No person shall publish a Digital Signature Certificate or otherwise make it available to any other person with the knowledge that—

(a) the Certifying Authority listed in the certificate has not issued it; or

(b) the subscriber listed in the certificate has not accepted it; or

(c) the certificate has been revoked or suspended,

unless such publication is for the purpose of verifying a digital signature created prior to such suspension or revocation.

(2) Any person who contravenes the provisions of sub-section (1) shall be punished with imprisonment for a term which may extend to two years, or with fine which may extend to one lakh rupees, or with both.

27.12.15Publication for fraudulent purpose.

- ✓ Whoever knowingly creates, publishes or otherwise makes available a Digital Signature Certificate for any fraudulent or unlawful purpose shall be punished with imprisonment for a term which may extend to two years, or with fine which may extend to one lakh rupees, or with both.

27.12.16Act to apply for offence or contravention commited outside India.

(1) Subject to the provisions of sub-section (2), the provisions of this Act shall apply also to any offence or contravention committed outside India by any person irrespective of his nationality.

(2) For the purposes of sub-section (1), this Act shall apply to an offence or contravention committed outside India by any person if the act or conduct constituting the offence or contravention involves a computer, computer system or computer network located in India.

27.12.17Confiscation.

- ✓ Any computer, computer system, floppies, compact disks, tape drives or any other accessories related thereto, in respect of which any provision of this Act. rules, orders or regulations made thereunder has been or is being contravened, shall be liable to confiscation:

- ✓ where it is established to the satisfaction of the court adjudicating the confiscation that the person in whose possession, power or control of any such computer, computer system, floppies, compact disks, tape drives or any other accessories relating to is found is not responsible for the contravention of the provisions of this Act, rules, orders or regulations made thereunder, the court may, instead of making an order for confiscation of such computer, computer system, floppies, compact disks, tape drives or any other accessories related thereto, make such other order authorised by this Act against the person contravening of the provisions of this Act, rules, orders or regulations made thereunder as it may think fit.

27.12.18 Penalties or confiscation not to interfere with other punishments.

- ✓ No penalty imposed or confiscation made under this Act shall prevent the imposition of any other punishment to which the person affected thereby is liable under any other law for the time being in force.

27.12.19 Power to investigate offences.

- ✓ Notwithstanding anything contained in the Code of Criminal Procedure, 1973, a police officer not below the rank of Deputy Superintendent of Police shall investigate any offence under this Act.

27.13 NETWORK SERVICE PROVIDERS NOT TO BE LIABLE IN CERTAIN CASES

- ✓ For the removal of doubts, it is hereby declared that no person providing any service as a network service provider shall be liable under this Act, rules or regulations made thereunder for any third party information or data made available by him if he proves that the offence or contravention was committed without his knowledge or that he had exercised all due diligence to prevent the commission of such offence or contravention.

27.14 MISCELLANOUS PROVISIONS

27.14.1 Power of police officer and other officers to enter, search, etc.

1. Notwithstanding anything contained in the Code of Criminal Procedure, 1973, any police officer, not below the rank of a Deputy Superintendent of Police, or any other officer of the Central Government or a State Government authorised by the Central Government in this behalf may enter any public place and search and arrest without warrant any person

found therein who is reasonably suspected or having committed or of committing or of being about to commit any offence under this Act

***Explanation.—For* the purposes of this sub-section, the expression "public place" includes any public conveyance, any hotel, any shop or any other place intended for use by, or accessible to the public.**

2. Where any person is arrested under sub-section (1) by an officer other than a police officer, such officer shall, without unnecessary delay, take or send the person arrested before a magistrate having jurisdiction in the case or before the officer-in-charge of a police station.
3. The provisions of the Code of Criminal Procedure, 1973 shall, subject to the provisions of this section, apply, so far as may be, in relation to any entry, search or arrest, made under this section.

27.14.2 Act to have overriding effect.

- ✓ The provisions of this Act shall have effect notwithstanding anything inconsistent therewith contained in any other law for the time being in force.

27.14.3Controller, Deputy Controller and Assistant Controllers to be public servants.

- ✓ The Presiding Officer and other officers and employees of a Cyber Appellate Tribunal, the Controller, the Deputy Controller and the Assistant Controllers shall be deemed to be public servants within the meaning of section 21 of the Indian Penal Code.

27.14.4Power to give directions.

- ✓ The Central Government may give directions to any State Government as to the carrying into execution in the State of any of the provisions of this Act or of any rule, regulation or order made there under.

27.14.5 Protection of action taken in good faith.

- ✓ No suit, prosecution or other legal proceeding shall lie against the Central Government, the State Government, the Controller or any person acting on behalf of him, the Presiding Officer, adjudicating officers and the staff of the Cyber Appellate Tribunal for anything which is in good faith done or intended to be done in pursuance of this Act or any rule, regulation or order made thereunder.

27.14.6Offences by companies.

- ✓ Where a person committing a contravention of any of the provisions of this Act or of any rule, direction or order made thereunder is a company, every person who, at the time the contravention was committed, was in charge of, and was responsible to, the company for the conduct of business of the company as well as the company, shall be guilty of the contravention and shall be liable to be proceeded against and punished accordingly:
- ✓ Provided that nothing contained in this sub-section shall render any such person liable to punishment if he proves that the contravention took place without his knowledge or that he exercised all due diligence to prevent such contravention.
- ✓ Notwithstanding anything contained in sub-section (1), where a contravention of any of the provisions of this Act or of any rule, direction or order made thereunder has been committed by a company and it is proved that the contravention has taken place with the consent or connivance of, or is attributable to any neglect on the part of, any director, manager, secretary or other officer of the company, such director, manager, secretary or other officer shall also be deemed to be guilty of the contravention and shall be liable to be proceeded against and punished accordingly.

27.14.7Removal of difficulties.

(1) If any difficulty arises in giving effect to the provisions of this Act, the Central Government may, by order published in the Official Gazette, make such provisions not inconsistent with the provisions of this Act as appear to it to be necessary or expedient for removing the difficulty:

(2) Every order made under this section shall be laid, as soon as may be after it is made, before each House of Parliament.

27.14.8Constitution of Advisory Committee.

1. The Central Government shall, as soon as may be after the commencement of this Act, constitute a Committee called the Cyber Regulations Advisory Committee.
2. The Cyber Regulations Advisory Committee shall consist of a Chairperson and such number of other official and non-official members representing the interests principally affected or having special knowledge of the subject-matter as the Central Government may deem fit.
3. The Cyber Regulations Advisory Committee shall advise—

(a) the Central Government either generally as regards any rules or for any other purpose connected with this Act;

(b) the Controller in framing the regulations under this Act.

4. There shall be paid to the non-official members of such Committee such travelling and other allowances as the Central Government may fix.

27.14.9Special provisions as to evidence relating to electronic record.

- ✓ The contents of electronic records may be proved in accordance with the provisions of section 65B.

27.14.10 Admissibility of electronic records.

- ✓ Any information contained in an electronic record which is printed on a paper, stored, recorded or copied in optical or magnetic media produced by a computer (hereinafter referred to as the computer output) shall be deemed to be also a document, if the conditions mentioned in this section are satisfied in relation to the information and computer in question and shall be admissible in any proceedings, without further proof or production of the original, as evidence of any contents of the original or of any fact stated therein of which direct evidence would be admissible.

27.14.11Presumption as to electronic records and digital signatures.

1. In any proceedings involving a secure electronic record, the Court shall presume unless contrary is proved, that the secure electronic record has not been altered since the specific point of time to which the secure status relates
2. In any proceedings, involving secure digital signature, the Court shall presume unless the contrary is proved that—

(a) the secure digital signature is affixed by subscriber with the intention of signing or approving the electronic record;

(b) except in the case of a secure electronic record or a secure digital signature, nothing in this section shall create any presumption relating to authenticity and integrity of the electronic record or any digital signature.

27.14.12 Presumption as to Digital Signature Certificates.

- ✓ The Court shall presume, unless contrary is proved, that the information listed in a Digital Signature Certificate is correct, except

for information specified as subscriber information which has not been verified, if the certificate was accepted by the subscriber.".

27.14.13.1.1 Presumption as to electronic messages.

- ✓ The Court may presume that an electronic message forwarded by the originator through an electronic mail server to the addressee to whom the message purports to be addressed corresponds with the message as fed into his computer for transmission; but the Court shall not make any presumption as to the person by whom such message was sent.