KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed University Established Under Section 3 of UGC Act 1956)
Coimbatore - 641021.
(For the candidates admitted from 2016 onwards)
DEPARTMENT OF COMPUTER SCIECE

Semester – V

16CSU501A **INFORMATION SECURITY** **4H – 4C**

SCOPE

This course provides an overview of Information Security and Assurance. Students will be exposed to the spectrum of security activities, methods, methodologies, and procedures with emphasis on practical aspects of Information Security.

COURSE OBJECTTIVE

1. To provide an understanding of principal concepts, major issues, technologies and basic approaches in information security.
2. Develop a basic understanding of cryptography, how it has evolved and some key encryption techniques used today.
3. Develop an understanding of security policies (such as authentication, integrity and confidentiality), as well as protocols to implement such policies in the form of message exchanges.

COURSE OUTCOME

1. To master information security governance, and related legal and regulatory issues
2. To be familiar with how threats to an organization are discovered and analyzed.
3. To be familiar with advanced security issues and technologies

Unit I

Introduction : Security, Attacks, Computer Criminals, Security Services, Security Mechanisms.

Cryptography : **Substitution ciphers, Transpositions Cipher, Confusion, diffusion.**

Unit – II

Symmetric, Asymmetric Encryption. DES Modes of DES, Uses of Encryption, Hash function, key exchange, Digital Signatures, Digital Certificates

Unit – III

Program Security: Secure programs, Non malicious Program errors, Malicious codes virus, Trap doors, Salami attacks, Covert channels, Control against program. **Threats:** Protection in OS: Memory and Address Protection, ACSUess control, File Protection, User Authentication**.**

Unit – IV

Database Security: Requirements, Reliability, Integrity, Sensitive data, Inference, Multilevel Security**. Security in Networks :** Threats in Networks, Security Controls, firewalls, Intrusion detection systems, Secure e-mails

Unit V

Administrating Security

Security Planning, Risk Analysis, Organisational Security Policy, Physical Security. Ethical issues in Security: Protecting Programs and data. Information and law.

Suggested Readings

1. Pfleeger, C. P., & Pfleeger, S. L.(2006). Security in Computing. New Delhi: Prentice Hall of India
2. Stallings, W. (2010). Network Security Essentials: Applications and Standards(4th ed.).

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

LECTURE PLAN
DEPARTMENT OF COMPUTER SCIENCE

STAFF NAME: Dr.S.Hemalatha
SUBJECT NAME: Information Security SUB.CODE:16CSU501A
SEMESTER: V CLASS: III B.Sc (CS)

S.No	Lecture Duration Period	Topics to be Covered	Support Material/Page Nos
		UNIT-I	
1.	1	**Introduction –** Computer Security	T1:31-33,
2.	1	**Attacks :** • Vulnerabilities, Threats, Attacks and Controls	T1:35-38 R2:21 -22
3.	1	**Attacks:** • Passive Attacks • Active Attacks	T2 :26 -29
4.	1	Computer Criminals	T1:51 - 52
5.	1	Security Services, Security Mechanisms	T2 :29-31, W1
6.	1	**Cryptography** - Substitution Cipher	T1:74-84
7.	1	Transpositions Cipher	T1:85-89
8.	1	Confusion and Diffusion	T1:93 - 94
9.	1	Recapitulation and Discussion of Important Questions	
Total No Of Periods Planned For Unit 1 :9			
		UNIT-II	
1.	1	**Encryption Algorithms:** Introduction	W3
2.	1	Symmetric and Asymmetric Encryption Systems	T1:92 -93 T2: 57- 59 R1: 259 -264
3.	1	Stream and Block Ciphers	
4.	1	**Data Encryption Standard** • Overview of the DES Algorithms	T1:98-99 T2: 48- 51, W2

5.	1	**Data Encryption Standard** • Double and Triple DES • Security of the DES	T1:99-102 T2: 52- 53
6.	1	Uses of Encryption , Hash functions	T1:109 , T1:109-110 R1:279
7.	1	Key Exchange , Digital Signatures	T1:110-112 R1:270 T1:112-114 T2: 92, R1:280
8.	1	Digital Certificates Certificates to Authenticate an identity	T1:114-117 T2: 93 T1:117-121
9.	1	Recapitulation and Discussion of Important Questions	
Total No Of Periods Planned For Unit II :9			
		UNIT-III	
1.	1	**Program Security :** Secure Programs Non malicious Program Errors	T1:128-133 T1:133-141
2.	1	**Malicious Code Virus**	T1:143-158
3.	1	Trapdoors**,** Salami Attack , Covert Channels	T1:171-180 R2:24 T1:180-190
4.	1	Control Against Program Threats	T1: 190-211
5.	1	**Protection in OS:** Memory and Address Protection	T1: 223-234, W4
6.	1	Control of Access	T1:234-245 R1:218 -220
7.	1	File Protection Mechanisms	T1:245-249
8.	1	User Authentication	T1:249-264
9.	1	Recapitulation and Discussion of Important Questions	
Total No Of Periods Planned For Unit III :9			
		UNIT-IV	
1.	1	**Database Security:** Security Requirements	T1:354-359
2.	1	Reliability and Integrity Sensitive Data	T1:359-365 T1: 365-371

3.	1	Inference Multilevel Security	T1: 371-361 T1:386-396
4.	1	Security in Networks	T1: 426-469,
5.	1	Security Controls	T1:470-500
6.	1	Firewalls	T1:504-514 R1:285 - 310
7.	1	Intrusion Detection Systems	T1:514-520, W5 R1:319 - 322
8.	1	Secure E-mail	T1:520-526
9.	1	Recapitulation and Discussion of Important Questions	
Total No Of Periods Planned For Unit IV:9			
		UNIT-V	
1.	1	**Administrating Security:** Security Planning • Contents of a Security Plan	T1:538 - 547
2.	1	**Administrating Security:** Security Planning • Business Continuity Plans	T1:548 - 554
3.	1	**Risk Analysis**	T1:554 - 577
4.	1	**Organizational Security Policies**	T1:577 - 586
5.	1	**Physical Security:** • Natural Disasters	T1:586 - 591
6.	1	**Physical Security:** • Interception of Sensitive Information	T1:591 - 596
7.	1	Ethical Issues in Security: Protecting Programs and Data	T1:677 – 692,W6
8.	1	Information and Law	T1:693 - 699
9.	1	Recapitulation and Discussion of important Questions	
10.	1	Discussion of Previous ESE Question Papers.	
11.	1	Discussion of Previous ESE Question Papers.	
12.	1	Discussion of Previous ESE Question Papers.	

Total No of Periods planned for Unit V: 12
Total Planned Hours:-48

Text Book:

1. Pfleeger, C. P., & Pfleeger, S. L. (2006). "Security in Computing". New Delhi: Prentice Hall of India
2. Stallings, W. (2010). "Network Security Essentials: Applications and Standards" (4th ed.).

Reference Books:

R1 - Kizza, Joseph Migga, "Computer Network Security", @ 2005 Springer Science+BusinessMedia, Inc.

R2 - John E. Canavan, "Fundamentals of Network Security", 2001 ARTECH HOUSE, INC.

Websites

1. https://www.rbc.com/privacysecurity/ca/security-mechanisms.html
2. https://www.tutorialspoint.com/cryptography/data_encryption_standard.htm
3. http://www.networksorcery.com/enp/data/encryption.htm
4. http://www.ques10.com/p/3569/various-ways-of-memory-and-address-protection-1/
5. https://www.tutorialspoint.com/information_security_cyber_law/network_security.htm
6. http://www.brainkart.com/article/Ethical-Issues-in-Computer-ecurity_9739/

UNIT – 1

Introduction : Security, Attacks, Computer Criminals, Security Services, Security Mechanisms.
Cryptography : Substitution ciphers, Transpositions Cipher, Confusion, diffusion.

1.1 Introduction

Security is a continuous process of protecting an object from attack. That object may be a person, an organization such as a business, or property such as a computer system or a file. When we consider a computer system, for example, its security involves the security of all its resources such as its physical hardware components such as readers, printers, the CPU, the monitors, and others.

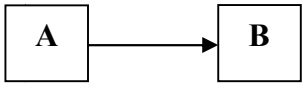
Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction. **Computer security** is the generic name for the collection of tools designed to protect the processed and stored data and to thwart hackers. **Network security** is the generic name for the collection of tools designed to protect data during their transmission.

The differences among **information security, computer security** and **network security** lie primarily in the approach to the subject, the methodologies used and the areas of concentration. **Information security** is concerned with the confidentiality, integrity and availability of data regardless of the form the data may take: electronic, print, or other forms. **Computer security** can focus on ensuring the availability and correct operation of a computer system without concern for the information stored or processed by the computer. **Network security** focuses on protecting data during their transmission.

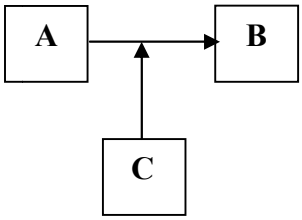
Information security = confidentiality + integrity + availability + authentication

Principles of Information Security:

1. **Confidentiality**: The principle of confidentiality specifies that only the sender and the intendedrecipient should be able to access the content of the message.



2. **Integrity**: The confidential information sent by A to B which is accessed by C without thepermission or knowledge of A and B.



3.**Availability**: It means that assets are accessible to authorized parties at appropriate times.

4.**Authentication:** This mechanism helps in establishing proof of identification.

1.2 Security Attacks:

Security attacks are those attacks on information and data to steal, delete or misuse them. These attacks are taking advantage of the weaknesses of either information technology.

Vulnerabilities, Threats, Attacks, and Controls

A computer-based system has three separate but valuable components: hardware, software, and data. Each of these assets offers value to different members of the community affected by the system.To analyze security, we can brainstorm about the ways in which the system or its information can experience some kind of loss or harm. For example, we can identify data whose format or contents should be protected in some way. We want our security system to make sure that no data are disclosed to unauthorized parties. Neither do we want the data to be modified in illegitimate ways. At the same time, we must ensure that legitimate users have access to the data. In this way, we can identify weaknesses in the system.

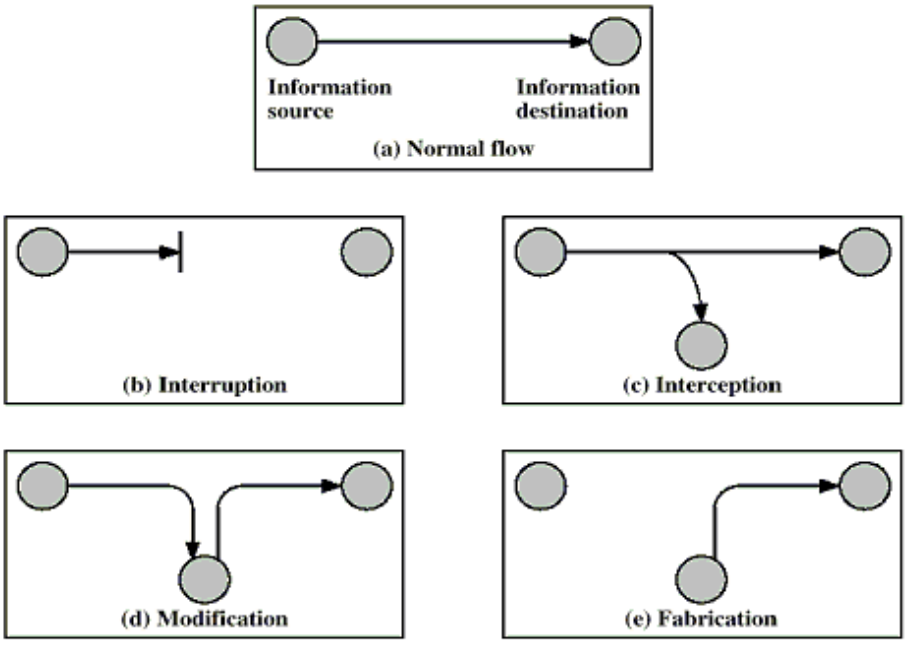
Vulnerability is a weakness in the security system, for example, in procedures, design, or implementation that might be exploited to cause loss or harm. For instance, a particular system may be vulnerable to unauthorized data manipulation because the system does not verify a user's identity before allowing data access.

A threat to a computing system is a set of circumstances that has the potential to cause loss or harm.

An attackcan also be launched by another system, as when one system sends an overwhelmingset of messages to another, virtually shutting down the second system's ability tofunction. Unfortunately, we have seen this type of attack frequently, as denial-of serviceattacks flood servers with more messages than they can handle.

To address these problems, we use a control as a protective measure. Thatis, a control is an action, device, procedure, or technique that removes or reduces vulnerability.

A threat is blocked by control of vulnerability.We can view any threat as being one of four kinds: interruption, interception, modification, and fabrication. Each threat exploits vulnerabilities of the assets in computing systems; the threats are illustrated in Figure.



- In an interruption, an asset of the system becomes lost, unavailable, or unusable.An example is malicious destruction of a hardware device, erasure of aprogram or data file, or malfunction of an operating system file manager so thatit cannot find a particular disk file.
- An interception means that some unauthorized party has gained access to an asset. The outside party can be a person, a program, or a computing system. Examples of this type of failure are illicit copying of program or data files, or wiretapping to obtain data in a network.

Although a loss may be discovered fairly quickly, a silent interceptor may leave no traces by which the interception can be readily detected.

- If an unauthorized party not only accesses but tampers with an asset, the threat is a modification. For example, someone might change the values in a database, alter a program so that it performs an additional computation, or modify data being transmitted electronically. It is even possible to modify hardware. Some cases of modification can be detected with simple measures, but other, more subtle, changes may be almost impossible to detect.
- Finally, an unauthorized party might create a fabrication of counterfeit objects on a computing system. The intruder may insert spurious transactions to a network communication system or add records to an existing database. Sometimes these additions can be detected as forgeries, but if skillfully done, they are virtually indistinguishable from the real thing.

Method, Opportunity, and Motive (MOM)

A malicious attacker must have three things:

• **method**: the skills, knowledge, tools, and other things with which to be able to pull off the attack

• **opportunity**: the time and access to accomplish the attack

• **motive**: a reason to want to perform. this attack against this system.

Types of Attacks:

- Active Attacks
- Passive Attacks

Active Attacks

Active attacks are highly malicious in nature, often locking out users, destroying memory or files, or forcefully gaining access to a targeted system or network.

Examples: Viruses, worms, malware, Denial of Service attacks, and password crackers.

Viruses: pieces of code that attach to host programs and propagate when an infected program executes.

Worms: particular to networked computers, carry out pre-programmed attacks to jump across the network.

Malware - Malicious software, commonly known as malware, is any software that brings harm to a computer system.

DoS - A denial-of-service attack is a security event that occurs when an attacker takes action that prevents legitimate users from accessing targeted computer systems, devices or other network resources

Password Crackers - A password cracker is an application program that is used to identify an unknown or forgotten password to a computer or network resources.

Passive Attacks

A passive attack is an information security event or incident based on monitoring or scanning communications, information flows or systems. In some cases, passive attacks are difficult to detect because they simply monitor as opposed to trying to break into a system. The following are illustrative examples.

- **Tapping** - Monitoring unencrypted communications such as emails or telephone calls.
- **Encryption** - Intercepting encrypted information flows and trying to break the encryption.
- **Scanning**- Scanning a device connected to the internet for vulnerabilities such as open ports or a weak operating system version.
- **Traffic Analysis** - Monitoring internet traffic to build data such as who is visiting what website.

Difference between Active & Passive Attacks:

S.No	Active Attacks	Passive Attacks
1.	Access & Modify information	Access Information
2.	System is harmed	No harm to system
3.	Easy to detect than prevent	Difficult to detect than prevent
4.	Masquerading, Repudiation, Denial of Service	Traffic Analysis

1.3 Computer Criminals

Computer crime, or cybercrime, is defined as any criminal activity in which computers, or a computer network, are the method or source of a crime. This encompasses a wide variety of crimes, from hacking into databases and stealing sensitive information to using computers to set up illegal activities.

Types of computer crime

1. Hacking- Currently defined as to gain illegal or unauthorized access to a file, computer or network.

2. Identity Theft - Various crimes in which a criminal or large group uses the identity of an unknowing, innocent person.

- **Phishing:** E-mail fishing for personal and financial information disguised as legitimate business e-mail.
- **Pharming**: False websites that fish for personal and financial information by planting false URLs in Domain Name Users.

3. Credit Card Fraud- A wide-ranging term for theft and fraud committed using a credit card or any similar payment mechanism as a fraudulent source of funds in a transaction.

4. Forgery - The process of making, adapting, or imitating objects, statistics, or documents, with the intent to deceive.

- **Digital Forgery** - New technologies are used to create fake checks, passports, visas, birth certificates with little skill or investments.

5. Scams- A confidence game or other fraudulent scheme, especially for making a quick profit, to cheat or swindle.

- **Auctions**: Some sellers do not sell items or send inferior products.
- **Stock Fraud**: Common method is to buy a stock low, send out email urging others to buy and then, sell when the price goes up.
- **Click Fraud**: Repeated clicking on an ad to either increase a site's revenue or to use up a competitors advertising budget.

1.4 Security Services

A service that enhances the security of data processing systems and information transfers. A security service makes use of one or more security mechanisms.

Classification of Security Services

1. Confidentiality (privacy)
2. Authentication (who created or sent the data)
3. Integrity (has not been altered)
4. Non-repudiation (the order is final)
5. Access control (prevent misuse of resources)
6. Availability (permanence, non-erasure)
   - Denial of Service Attacks
   - Virus that deletes files

1. **Data Confidentiality**: The protection of data from unauthorized disclosure.
   - Connection Confidentiality:The protection of all user data on a connection.
   - **Connectionless Confidentiality:** The protection of all user data in a single data block
   - **Selective-Field Confidentiality:** The confidentiality of selected fields within the userData on a connection or in a single data block.
   - **Traffic Flow Confidentiality:** The protection of the information that might bederived from observation of traffic flows.

2. **Authentication:** The assurance that the communicating entity is the one that itlaims to be.

- **Peer Entity Authentication**: Used in association with a logical connection toprovide confidence in the identity of the entities connected.
- **Data Origin Authentication:** In a connectionless transfer, provides assurance thatthe source of received data is as claimed.

3. **Data Integrity:** The assurance that data received are exactly as sent by anauthorized entity (i.e., contain no modification, insertion, deletion,or replay).

- **Connection Integrity with Recovery:** Provides for the integrity of all user data on aconnection and detects any modification,insertion, deletion, or replay of any datawithin an entire data sequence, with recoveryattempted.
- **Connection Integrity without Recovery**: As above, but provides only detectionwithout recovery.
- **Selective-Field Connection Integrity**: Provides for the integrity of selected fieldswithin the user data of a data block transferredover a connection and takes the form ofdetermination of whether the selected fieldshave been modified, inserted, deleted, orreplayed.
- **Connectionless Integrity:** Provides for the integrity of a single connectionless datablock and may take the form of detection of datamodification. Additionally, a limited form of replaydetection may be provided.
- **Selective-Field Connectionless Integrity**: Provides for the integrity of selectedfields within a single connectionless data block; takes the form of determination ofwhether the selected fields have beenmodified.

4. **Nonrepudiation:** Provides protection against denial by one of the entities involvedin a communication of having participated in all or part of thecommunication.

- Nonrepudiation, Origin: Proof that the message was sent by the specified party.
- Nonrepudiation, Destination: Proof that the message was received by the specified party

5. **Access Control:** The prevention of unauthorized use of a resource (i.e., this servicecontrols who can have access to a resource, under what conditionsaccess can occur, and what those accessing the resource are allowed to do).

6.**Availability**: It is ability to use information orresources desired

1.5 Security Mechanism - A mechanism that is designed todetect, prevent, or recover from a security attack.

- Secure Socket Layer (SSL) Encryption
- Authentication
- Firewalls
- Computer Anti-Virus Protection
- Data Integrity
- Ensuring Your Online Safety

1. Secure Socket Layer (SSL) Encryption: When you successfully login to Online Banking or another secure RBC website using an authentic user ID and password, our web servers will establish a secure socket layer (SSL) connection with your computer. This allows you to communicate with us privately and prevents other computers from seeing anything that you are transacting – so you can conduct online business with us safely. SSL provides 128-bit encrypted security so that sensitive information sent over the Internet during online transactions remains confidential.

2. Authentication To protect our users, we provide secure private websites for any business that users conduct with us. Users login to these sites using a valid client number or username and a password. Users are required to create their own passwords, which should be kept strictly confidential so that no one else can login to their accounts.

3. Firewalls: We use a multi-layered infrastructure of firewalls to block unauthorized access by individuals or networks to our information servers.

4. Computer Anti-Virus Protection: We are continuously updating our anti-virus protection. This ensures we maintain the latest in anti-virus software to detect and prevent viruses from entering our computer network systems.

5. Data Integrity The information you send to one of our secure private websites is automatically verified to ensure it is not altered during information transfers. Our systems detect if data was added or deleted after you send information. If any tampering has occurred, the connection is dropped and the invalid information transfer is not processed.

6. Ensuring Your Online Safety

- Find out how these security mechanisms safeguard our communications with youand learn how RBC helps to protect you against fraud

1.6 Substitution ciphers

Substitution cipher, data encryption scheme in which units of the plaintext (generally single letters or pairs of letters of ordinary text) are replaced with other symbols or groups of symbols. There are several types of substitution cryptosystems:

- Monoalphabetic substitution involves replacing each letter in the message with another letter of the alphabet
- Polyalphabetic substitution involves using a series of monoalphabetic ciphers that are periodically reused
- Homophonic substitution makes it possible to have each letter of the plaintext message correspond to a possible group of other characters
- Polygraphic substitution involves replacing a group of characters in the message with another group of characters

1. Caesar Cipher

It is a mono-alphabetic cipher wherein each letter of the plaintext is substituted by another letter to form the ciphertext. It is a simplest form of substitution cipher scheme.

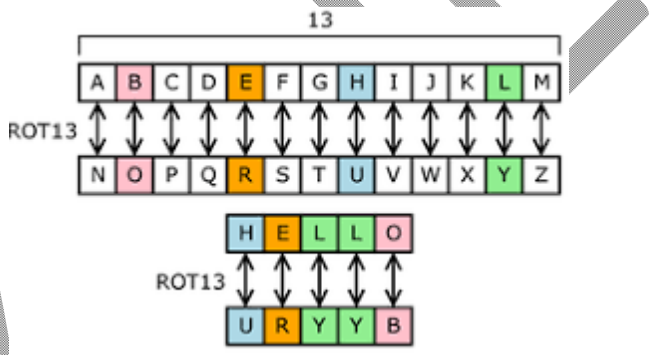
This cryptosystem is generally referred to as the Shift Cipher. The concept is to replace each alphabet by another alphabet which is 'shifted' by some fixed number between 0 and 25.

For this type of scheme, both sender and receiver agree on a 'secret shift number' for shifting the alphabet. This number which is between 0 and 25 becomes the key of encryption.

The name 'Caesar Cipher' is occasionally used to describe the Shift Cipher when the 'shift of three' is used.

Process of Shift Cipher

- In order to encrypt a plaintext letter, the sender positions the sliding ruler underneath the first set of plaintext letters and slides it to LEFT by the number of positions of the secret shift.
- The plaintext letter is then encrypted to the ciphertext letter on the sliding ruler underneath. The result of this process is depicted in the following illustration for an agreed shift of three positions. In this case, the plaintext 'tutorial' is encrypted to the ciphertext 'WXWRULDO'. Here is the ciphertext alphabet for a Shift of 3 –



2. Play fair ciphers

The Playfair Cipher is a manual symmetric encryption cipher invented in 1854 by Charles .In a playfair cipher the message is split into digraphs, pairs of two letters.

In playfair cipher, initially a key table is created. The key table is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table as we need only 25 alphabets instead of 26. If the plaintext contains J, then it is replaced by I.

The sender and the receiver deicide on a particular key, say 'tutorials'. In a key table, the first characters (going left to right) in the table is the phrase, excluding the duplicate letters. The rest of the table will be filled with the remaining letters of the alphabet, in natural order. The key table works out to be –

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

Process of Playfair Cipher

- First, a plaintext message is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter. Let us say we want to encrypt the message "hide money". It will be written as −

  HI DE MO NE YZ

- The rules of encryption are −
  - If both the letters are in the same column, take the letter below each one (going back to the top if at the bottom)

T	U	O	R	I	
A	L	S	B	C	
D	E	F	G	H	'H' and 'I' are in same column, hence take letter below them to replace. HI → QC
K	M	N	P	Q	
V	W	X	Y	Z	

If both letters are in the same row, take the letter to the right of each one (going back to the left if at the farthest right)

T	U	O	R	I	
A	L	S	B	C	
D	E	F	G	H	'D' and 'E' are in same row, hence take letter to the right of them to replace. DE → EF
K	M	N	P	Q	
V	W	X	Y	Z	

If neither of the preceding two rules are true, form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

T	U	O	R	I	'M' and 'O' nor on same column or same row, hence form rectangle as shown, and replace letter by picking up opposite corner letter on same row MO -> NU
A	L	S	B	C	
D	E	F	G	H	
K	M	N	P	Q	
V	W	X	Y	Z	

3. **Vigenere Cipher**

This scheme of cipher uses a text string (say, a word) as a key, which is then used for doing a number of shifts on the plaintext.

For example, let's assume the key is 'point'. Each alphabet of the key is converted to its respective numeric value: In this case,

p → 16, o → 15, i → 9, n → 14, and t → 20.

Thus, the key is: 16 15 9 14 20.

Process of Vigenere Cipher

- The sender and the receiver decide on a key. Say 'point' is the key. Numeric representation of this key is '16 15 9 14 20'.
- The sender wants to encrypt the message, say 'attack from south east'. He will arrange plaintext and numeric key as follows −

a	t	t	a	c	k	f	r	o	m	s	o	u	t	h	e	a	s	t
16	15	9	14	20	16	15	9	14	20	16	15	9	14	20	16	15	9	14

He now shifts each plaintext alphabet by the number written below it to create ciphertext as shown below

a	t	t	a	c	k	f	r	o	m	s	o	u	t	h	e	a	s	t
16	15	9	14	20	16	15	9	14	20	16	15	9	14	20	16	15	9	14
Q	I	C	O	W	A	U	A	C	G	I	D	D	H	B	U	P	B	H

- Here, each plaintext character has been shifted by a different amount – and that amount is determined by the key. The key must be less than or equal to the size of the message.
- For decryption, the receiver uses the same key and shifts received ciphertext in reverse order to obtain the plaintext.

Q	I	C	O	W	A	U	A	C	G	I	D	D	H	B	U	P	B	H
16	15	9	14	20	16	15	9	14	20	16	15	9	14	20	16	15	9	14
a	t	t	a	c	k	f	r	o	m	s	o	u	t	h	e	a	s	t

Note: It is significantly more secure than a regular Caesar Cipher.

There are two special cases of Vigenere cipher −

- The keyword length is same as plaintect message. This case is called Vernam Cipher. It is more secure than typical Vigenere cipher.
- Vigenere cipher becomes a cryptosystem with perfect secrecy, which is called One-time pad.

4.**One-Time Pad**

The circumstances are −

- The length of the keyword is same as the length of the plaintext.
- The keyword is a randomly generated string of alphabets.
- The keyword is used only once.

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

To encrypt a letter, we write the key underneath the plaintext. We take the plaintext letter at the top and the key letter on the left. The cross section of those two letters is the ciphertext. In the first letter of our example below, the crossing between the plaintext T and key X is ciphertext Q.

```
Plaintext : T H I S   I S   S E C R E T
OTP-Key   : X V H E   U W   N O P G D Z
-------------------------------------
Ciphertext: Q C P W   C O   F S R X H S
```

In groups : QCPWC OFSRX HS

to decrypt a letter, we take the key letter on the left and find the ciphertext letter in that row. The plaintext letter is at the top of the column where you found ciphertext letter. In our example, we takethe X row, find the Q in that row and see the plain T on top of that column. As a mnemonic we can consider the column header as plaintext, the row header as key and the square field as ciphertext.

1.7 Transposition Cipher

It is another type of cipher where the order of the alphabets in the plaintext is rearranged to create the ciphertext. The actual plaintext alphabets are not replaced.

An example is a 'simple columnar transposition' cipher where the plaintext is written horizontally with a certain alphabet width. Then the ciphertext is read vertically as shown.

For example, the plaintext is "golden statue is in eleventh cave" and the secret random key chosen is "five". We arrange this text horizontally in table with number of column equal to key value. The resulting text is shown below.

g	o	l	d	e
n	s	t	a	t
u	e	i	s	i
n	e	l	e	v
e	n	t	h	c
a	v	e		

The ciphertext is obtained by reading column vertically downward from first to last column. The ciphertext is 'gnuneaoseenvltiltedasehetivc'.

To decrypt, the receiver prepares similar table. The number of columns is equal to key number. The number of rows is obtained by dividing number of total ciphertext alphabets by key value and rounding of the quotient to next integer value.

The receiver then writes the received ciphertext vertically down and from left to right column. To obtain the text, he reads horizontally left to right and from top to bottom row.

Rail fence techniques

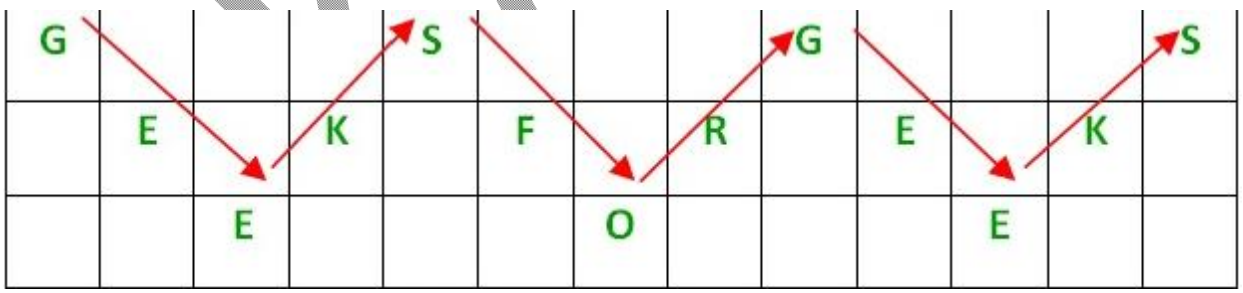
In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.

When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus the alphabets of the message are written in a zig-zag manner.

After each alphabet has been written, the individual rows are are combined to obtain the cipher-text.

For example, if the message is "GeeksforGeeks" and the number of rails = 3 then cipher is prepared as:



Decryption

The number of columns in rail fence cipher remains equal to the length of plain-text message. And the key corresponds to the number of rails.

Hence, rail matrix can be constructed accordingly. Once we've got the matrix we can figure-out the spots where texts should be placed (using the same way of moving diagonally up and down alternatively).

Then, we fill the cipher-text row wise. After filling it, we traverse the matrix in zig-zag manner to obtain the original text.

Implementation:

Let cipher-text = "GsGsekfrekeoe" , and Key = 3

Number of columns in matrix = len(cipher-text) = 12

Number of rows = key = 3

Row Transposition ciphers

- In general write message in a number of columns and then use some rule to read off from these columns
- key could be a series of number being the order to: read off the cipher; or write in the plain-text

Plain: THESIMPLESTPOSSIBLETRANSPOSITIONSXX

Key(R 2 5 4 1 3

Key (W): 4 1 5 3 2

T H E S I S T I E H

M P L E S E M S L P

T P O S S S T S O P

I B L E T E I T L B

R A N S P S R P N A

O S I T I T O I I S

O N S X X X O X S N

Cipher: STIEH EMSLP STSOP EITLB SRPNA TOIIS XOXSN

Confusion and Diffusion

cryptography gave two propertiesthat a good cryptosystem should have to hinder statistical analysis namely,diffusion and confusion.

Diffusion means that if we change a character of the plaintext, then severalcharacters of the ciphertext should change, and similarly, if we change acharacter of the ciphertext, then several characters of the plaintext shouldchange. We saw that the Hill cipher has this property. This means thatfrequency statistics of letters, [digraphs], etc. in the plaintext are diffusedover several characters in the ciphertext, which means that much moreciphertext is needed to do a meaningful statistical attack.

Confusion means that the key does not relate in a simple way to theciphertext. In particular, each character of the ciphertext should depend onseveral parts of the key. For example, suppose we have a Hill cipher with a matrix, and suppose we have a plaintext-ciphertext pair of lengthwith which we are able to solve for the encryption matrix. If we change onecharacter of the ciphertext, one column of the matrix can change completely.Of course, it would be more desirable to have the entire key change. When asituation like that happens, the cryptanalyst would probably need to solve forthe entire key simultaneously, rather than piece by piece.

POSSIBLE QUESTIONS

PART A

Q.NO 1 TO 20 (MULTIPLE CHOICE QUESTIONS)

PART B (2 MARKS)

1. Define Security
2. What is the need for information security?
3. Define Security Attacks
4. What are the characteristics of information?
5. Define Cryptography.
6. Differentiate between attack and threat.
7. What ate the threats to information security
8. What is the main drawback of the one time pad
9. What is encryption
10. List out the techniques used in substitution ciphers

PART C (6 MARKS)

1. Explain the components of information system.
2. Summarize the steps performed in playfair techniques.
3. With suitable sketched, explain the working of transpositions ciphers
4. Consider the following:

   Plaintext: “PROTOCOL”

   Secret key: “NETWORK”
5. What is the corresponding cipher text using play fair cipher method?
6. Explain various types of attack on computer system
7. What is security mechanism? List and explain various security mechanisms.
8. Explain Vigenere cipher with suitable example

KARPAGAM ACADEMY OF HIGHER EDUCTATION

DEPARTMENT OF COMPUTER SCIENCE

III B.Sc CS

INFORMATION SECURITY[16CSU501A]

UNIT 1

S.NO	Question	Option 1	Option 2	Option 3	Option 4	Answer
1	CMAC stands for	cipher based mask authentication code	cipher based message architecture code	common based message architecture	cipher based message authentication code	cipher based message authentication code
2	_____ is the study of techniques for ensuring the secrecy and/or authenticity of information	cryptology	network security	computer security	internet	cryptology
3	_____ which deals with the defeating such techniques to recover information	computer security	network security	cryptanalysis	internet	cryptanalysis
4	_____ area covers the use of cryptographic algorithms in network protocol and network application	computer security	network security	internet	mobile security	network security
5	_____ term to refer to the security of computers against intruders.	internet	network security	computer security	mobile security	computer security
6	the generic name for the collection of tools designed to protect data and to thwart hackers is_____	computer security	network security	internet	mobile security	computer security
7	_____ measures are needed to protect data during their transmission	mobile security	computer security	internet	network security	network security
8	Rail fence cipher technique takes in the form of	zigzag model	straight	downward direction	upward direction	zigzag model
9	CERT stands for_____	computer electric response team	computer emergency response team	computer electric reply team	computer electric reply task	computer emergency response team
10	the _____ security architecture is useful to managers as a way of organizing the task of providing security	OSI	SIO	OSII	OSA	OSI
11	the OSI architecture focuses on security _____,_____ and _____	internet,base,security	net,mech,none	attack,internet,services	attack,mechanism,service	attack,mechanism,service
12	Any action that compromises the security of information owned by an organization is known as_____	Security attack	Security Service	Security mechanism	passive attacks	Security attack

13	. _____is a mechanism that is designed to detect,prevent or recover from a security attack	Security service	Security attack	Security mechanism	passive attacks	Security mechanism
14	_____ is a service that enhances the security of the data processing systems and the information transfers of an organization	Security mechanism	Security service	Security attack	active attacks	Security service
15	_____ are in the nature of eavesdropping on,or monitoring of, transmissions	mobile attack	active attacks	internet attacks	passive attacks	passive attacks
16	_____ are very difficult to detect because they do not involve any alteration of the data	active attacks	passive attacks	internet attacks	mobile attack	passive attacks
17	_____involves some modification of the data stream or the creation of a false stream.	active attacks	passive attacks	internet attacks	mobile attack	active attacks
18	the terms _____ and _____ were introduced by claude shannon to capture the two basic building blocks for any cryptographic system	diffusion confusion	diffusion	permutation	none	diffusion confusion
19	In _____ the statistical structure of the plain text is dissipated into long range statistics of the ciphertext.	diffusion	substitution	permutation	confusion	diffusion
20	the ____ prevents or inhibits the normal use or management of communication facilities.	masquerade	denial of service	Reply	X.802	denial of service
21	_____ seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible again	permutation	substitution	confusion	diffusion	confussion
22	A_____ is performed on the left half of the data	substitution	permutation	confusion	transformation	substitution
23	._____requires the access to information reduces may be controlled by or for the target system	authentication	Non repudiation	Access control	confidentiality	Access control
24	_____ requires that neither the sender nor the receiver of the message be able to deny the transmission	Non repudiation	Authentication	Access control	Access control	Non repudiation
25	_____requires that computer system assets be available to authorized parties when needed.	Non repudiation	Authentication	Availability	Access control	Availability
26	_____ is the protection of transmitted data from passive attacks	Authentication	confidentiality	Non repudiation	Access control	confidentiality
27	_____ provides for the corroboration of the source of a data unit	data origin authentication	peer entity authentication	transformation	Access control	data origin authentication
28	_____ exploit service flaws in computer to inhibit use by legitimate users	service threats	information threats	internet	network security	service threats

29	_____ threats intercepts or modify data on behalf of users who should not have access to that data	internet	service threats	information access	network security	information access
30	_____ and _____ are two examples of software attacks	viruses and worms	service	internet	network	viruses and worms
31	_____ encryption is a form of cryptosystem in which encryption and decryption are performed using the same key	symmetric	asymmetric	service	network	symmetric
32	_____encryption transforms plaintext into ciphertext using a secret key and an encryption algorithms	service	asymmetric	symmetric	network	symmetric
33	_____ involves trying all possible keys	brute force	symmetric	asymmetric	network	brute force
34	_____ techniques map plaintext elements into ciphertext elements	transposition	substitution	service	network	substitution
35	_____ techniques systematically transpose the positions of plaintext elements	transposition	substitution	service	network	transposition
36	_____machines are sophisticated precomputer hardware devices that use substitution techniques	computer	rotor	network	embedded	rotor
37	_____ is a technique for hiding a secret message within a larger one	steganography	encryption	decryption	transposition	steganography
38	An original message is known as_____	ciphertext	input	plaintext	output	plaintext
39	the coded message is called the_____	plaintext	ciphertext	input	output	ciphertext
40	the process of converting from plain text into ciphertext is known as_____	enciphering	deciphering	substitution	transposition	enciphering
41	restoring the plaintext from the ciphertext is_____	deciphering	enciphering	substitution	transposition	deciphering
42	the _____ key is also input to the encryption algorithm	secret	plain	cipher	keyword	secret
43	_____is essentially the encryption algorithm run in reverse	substitution	encryption	decryption	transposition	decryption
44	the ____ algorithm performs various substition and transformation on the plaintext	transposition	encryption	substitution	decryption	encryption
45	_____is the scrambled message produced as output	plaintext	ciphertext	input	output	ciphertext

46	_____ types of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext	transposition	substitution	cryptanalysis	steganography	cryptanalysis
47	A ____ techniques is one in which the letter of plaintext are replaced by other letters or by numbers or symbols	substitution	transposition	cryptanalysis	steganography	substitution
48	the _____cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet	caesar	monoalphabetic cipher	playfair cipher	hill ciphers	caesar
49	_____,the relative frequency of the letters can be determined and compared to a standard frequency distribution for english	playfair cipher	hill ciphers	monoalphabetic cipher	caesar	monoalphabetic cipher
50	____ treats diagrams in the plaintext as single units and translates these units into ciphertext diagrams	playfair cipher	monoalphabetic cipher	hill ciphers	caesar	playfair cipher
51	_____ takes m successive plaintext letters and substitutes for them m ciphertext letters	playfair cipher	monoalphabetic cipher	caesar	hill ciphers	hill ciphers
52	_____produces random output that bears no statistical relationship to the plaintext	one time pad	vigenere	rail fence	caesar	one time pad
53	_____ which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows	one time pad	rail fence	vigenere	caesar	rail fence
54	_____selected letters of printed or typewritten text are overwritten in pencil	pin punctures	invisible ink	character marking	type writer correction ribbon	character marking
55	_____a number of substances can be used for writing but leave no visible trace until heat	character marking	invisible ink	pin punctures	type writer correction ribbon	invisible ink
56	small _____ on selected letters are ordinarily not visible unless the paper is held up in front of a light	pin punctures	type writer correction ribbon	character marking	pin punctures	pin punctures
57	_____used between lines typed with a black ribbon	character marking	pin punctures	invisible ink	type writer correction ribbon	type writer correction ribbon
58	the machines consists of a set of independently rotating cylinders through which_____ can flow	electrical pulse	waves	rail fence	none	electrical pulse
59	_____,in which a keyword is concatenated with the plaintext itself to provide a running key	autokey system	random key system	rail fence	none	autokey system
60	the best known and one of the simplest such algorithm is referred to as the _____ cipher	vigenere	rail fence	one time pad	caesar	vigenere

Unit II

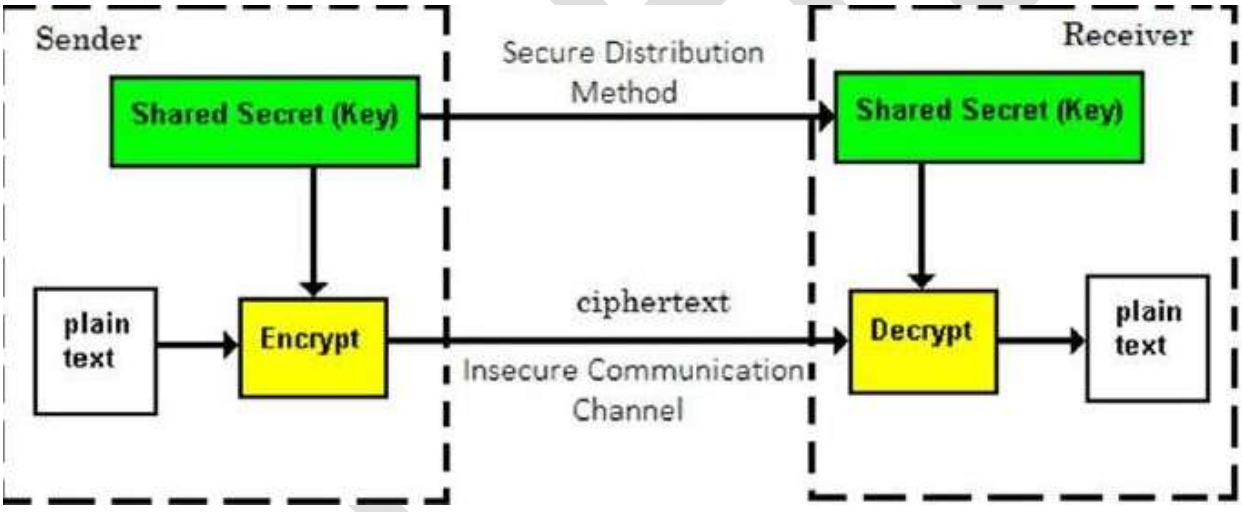
Symmetric, Asymmetric Encryption. DES Modes of DES, Uses of Encryption, Hash function, key exchange, Digital Signatures, Digital Certificates

1.1 Symmetric Key Encryption

The encryption process where same keys are used for encrypting and decrypting the information is known as Symmetric Key Encryption.

The study of symmetric cryptosystems is referred to as symmetric cryptography. Symmetric cryptosystems are also sometimes referred to as secret key cryptosystems.

A few well-known examples of symmetric key encryption methods are − Digital Encryption Standard (DES), Triple-DES (3DES), IDEA, and BLOWFISH.



all cryptosystems employed symmetric key encryption. Even today, its relevance is very high and it is being used extensively in many cryptosystems. It is very unlikely that this encryption will fade away, as it has certain advantages over asymmetric key encryption.

The salient features of cryptosystem based on symmetric key encryption are −

- Persons using symmetric key encryption must share a common key prior to exchange of information.
- Keys are recommended to be changed regularly to prevent any attack on the system.
- A robust mechanism needs to exist to exchange the key between the communicating parties. As keys are required to be changed regularly, this mechanism becomes expensive and cumbersome.
- In a group of n people, to enable two-party communication between any two persons, the number of keys required for group is $n \times (n – 1)/2$.
- Length of Key (number of bits) in this encryption is smaller and hence, process of encryption-decryption is faster than asymmetric key encryption.
- Processing power of computer system required to run symmetric algorithm is less.

Challenge of Symmetric Key Cryptosystem

There are two restrictive challenges of employing symmetric key cryptography.

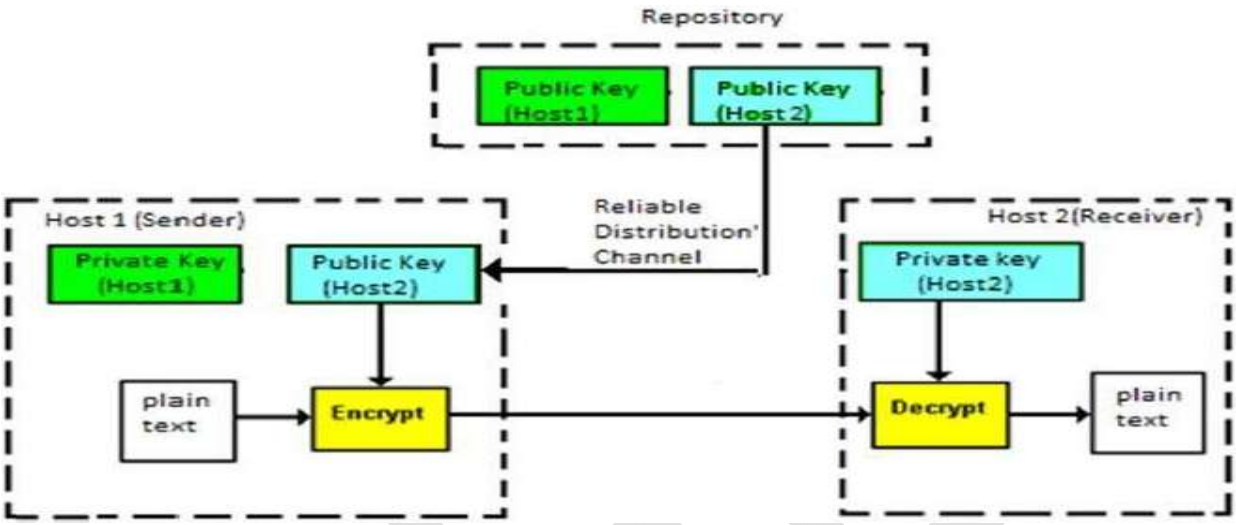
- **Key establishment** − Before any communication, both the sender and the receiver need to agree on a secret symmetric key. It requires a secure key establishment mechanism in place.
- **Trust Issue** − Since the sender and the receiver use the same symmetric key, there is an implicit requirement that the sender and the receiver 'trust' each other. For example, it may happen that the receiver has lost the key to an attacker and the sender is not informed.

These two challenges are highly restraining for modern day communication. Today, people need to exchange information with non-familiar and non-trusted parties. For example, a communication between online seller and customer. These limitations of symmetric key encryption gave rise to asymmetric key encryption schemes.

Asymmetric Key Encryption

The encryption process where different keys are used for encrypting and decrypting the information is known as Asymmetric Key Encryption. Though the keys are different, they are

mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible. The process is depicted in the following illustration –



Asymmetric Key Encryption is used for the necessity of pre-shared secret key between communicating persons.

The salient features of this encryption scheme are as follows −

- Every user in this system needs to have a pair of dissimilar keys, private key and public key. These keys are mathematically related − when one key is used for encryption, the other can decrypt the ciphertext back to the original plaintext.
- It requires putting the public key in public repository and the private key as a well-guarded secret. Hence, this scheme of encryption is also called Public Key Encryption.
- Though public and private keys of the user are related, it is computationally not feasible to find one from another. This is a strength of this scheme.
- When Host1 needs to send data to Host2, he obtains the public key of Host2 from repository, encrypts the data, and transmits.
- Host2 uses his private key to extract the plaintext.
- Length of Keys (number of bits) in this encryption is large and hence, the process of encryption-decryption is slower than symmetric key encryption.

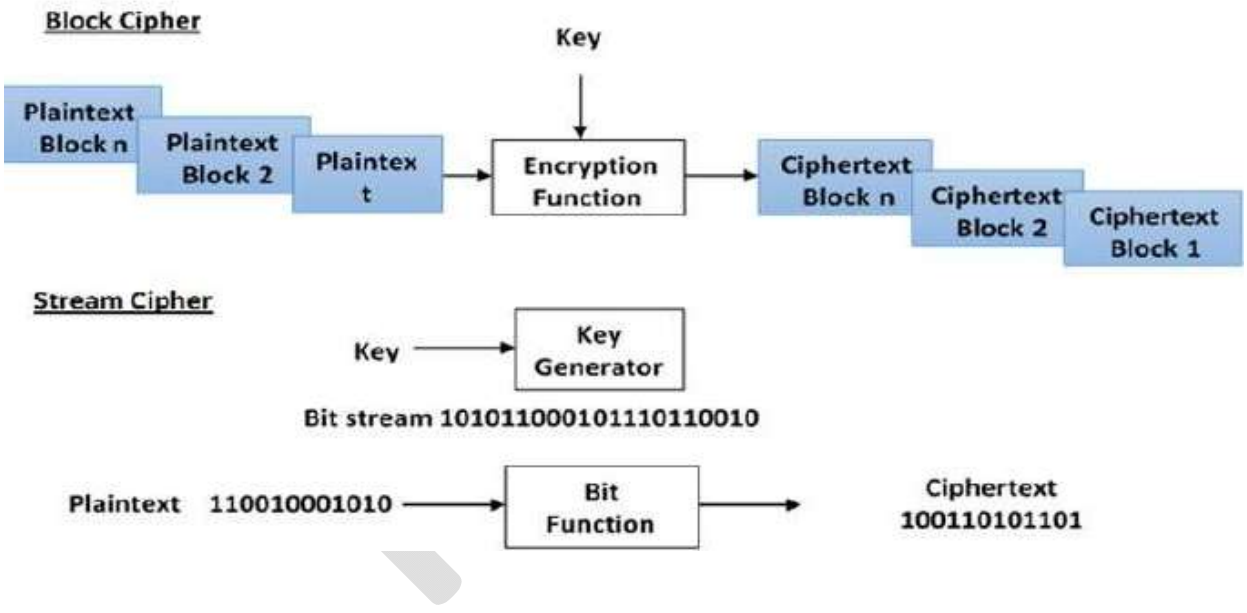
- Processing power of computer system required to run asymmetric algorithm is higher.

Block Ciphers

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

Stream Ciphers

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



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

Block Cipher

A block cipher takes a block of plaintext bits and generates a block of ciphertext bits, generally of same size. The size of block is fixed in the given scheme. The choice of block size

does not directly affect to the strength of encryption scheme. The strength of cipher depends up on the key length.

Block Size

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

Avoid very small block size − Say a block size is m bits. Then the possible plaintext bits combinations are then 2m. If the attacker discovers the plain text blocks corresponding to some previously sent ciphertext blocks, then the attacker can launch a type of 'dictionary attack' by building up a dictionary of plaintext/ciphertext pairs sent using that encryption key. A larger block size makes attack harder as the dictionary needs to be larger.

Do not have very large block size − With very large block size, the cipher becomes inefficient to operate. Such plaintexts will need to be padded before being encrypted.

Multiples of 8 bit − A preferred block size is a multiple of 8 as it is easy for implementation as most computer processor handle data in multiple of 8 bits

Padding in Block Cipher

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

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

Block Cipher Schemes

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

- **Digital Encryption Standard (DES)** − The popular block cipher of the 1990s. It is now considered as a 'broken' block cipher, due primarily to its small key size.

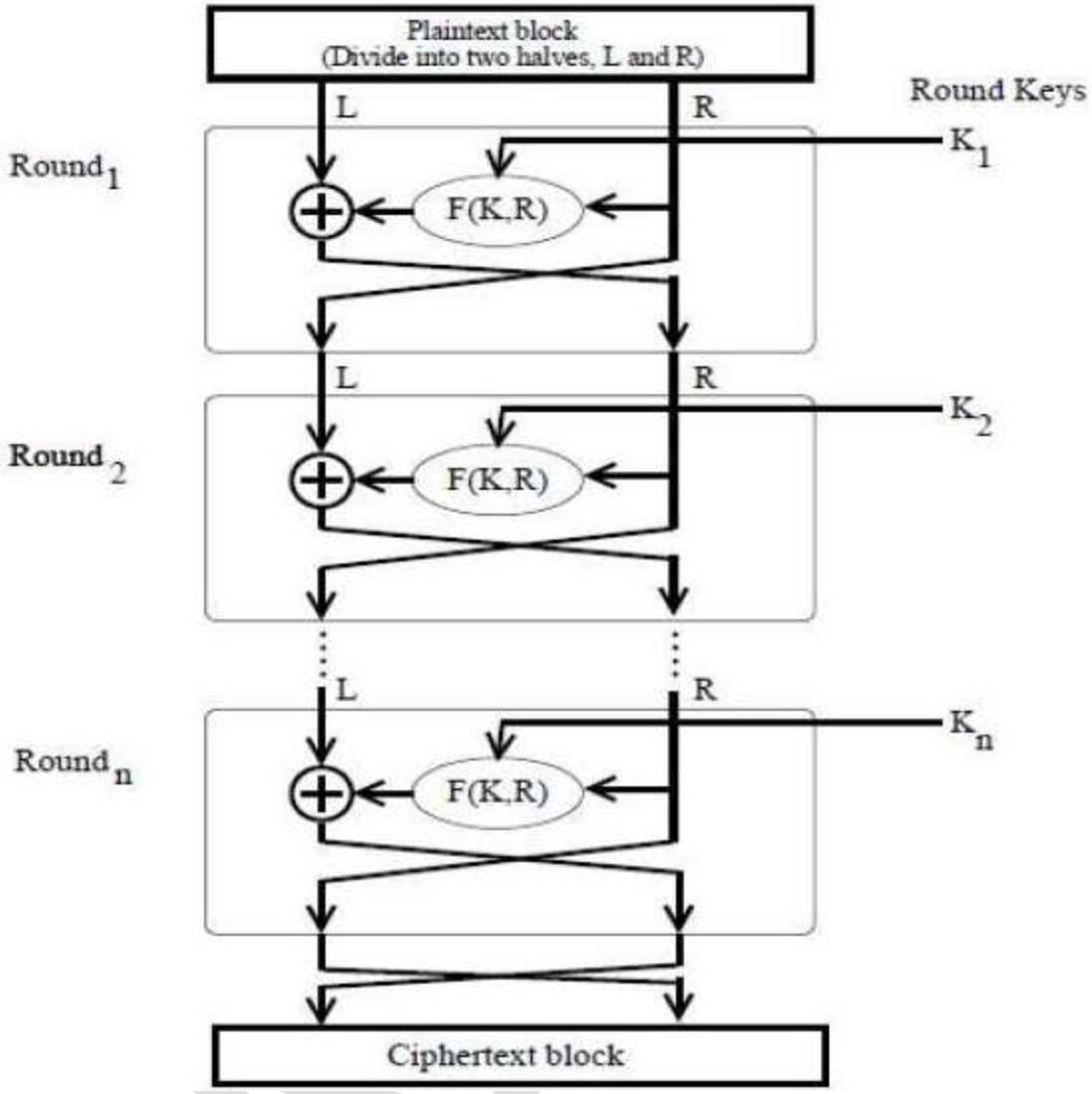
- **Triple DES** − It is a variant scheme based on repeated DES applications. It is still a respected block ciphers but inefficient compared to the new faster block ciphers available.
- **Advanced Encryption Standard (AES)** − It is a relatively new block cipher based on the encryption algorithm Rijndael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

Feistel Cipher is not a specific scheme of block cipher. It is a design model from which many different block ciphers are derived. DES is just one example of a Feistel Cipher. A cryptographic system based on Feistel cipher structure uses the same algorithm for both encryption and decryption.

Encryption Process

The encryption process uses the Feistel structure consisting multiple rounds of processing of the plaintext, each round consisting of a "substitution" step followed by a permutation step.

.



The input block to each round is divided into two halves that can be denoted as L and R for the left half and the right half.

In each round, the right half of the block, R, goes through unchanged. But the left half, L, goes through an operation that depends on R and the encryption key. First, we apply an encrypting function 'f' that takes two input − the key K and R. The function produces the output f(R,K). Then, we XOR the output of the mathematical function with L.

In real implementation of the Feistel Cipher, such as DES, instead of using the whole encryption key during each round, a round-dependent key (a subkey) is derived from the

encryption key. This means that each round uses a different key, although all these subkeys are related to the original key.

The permutation step at the end of each round swaps the modified L and unmodified R. Therefore, the L for the next round would be R of the current round. And R for the next round be the output L of the current round.

Above substitution and permutation steps form a 'round'. The number of rounds are specified by the algorithm design.

Once the last round is completed then the two sub blocks, 'R' and 'L' are concatenated in this order to form the ciphertext block.

The difficult part of designing a Feistel Cipher is selection of round function 'f'. In order to be unbreakable scheme, this function needs to have several important properties that are beyond the scope of our discussion.

Decryption Process

The process of decryption in Feistel cipher is almost similar. Instead of starting with a block of plaintext, the ciphertext block is fed into the start of the Feistel structure and then the process thereafter is exactly the same as described in the given illustration.

The process is said to be almost similar and not exactly same. In the case of decryption, the only difference is that the subkeys used in encryption are used in the reverse order.

The final swapping of 'L' and 'R' in last step of the Feistel Cipher is essential. If these are not swapped then the resulting ciphertext could not be decrypted using the same algorithm.

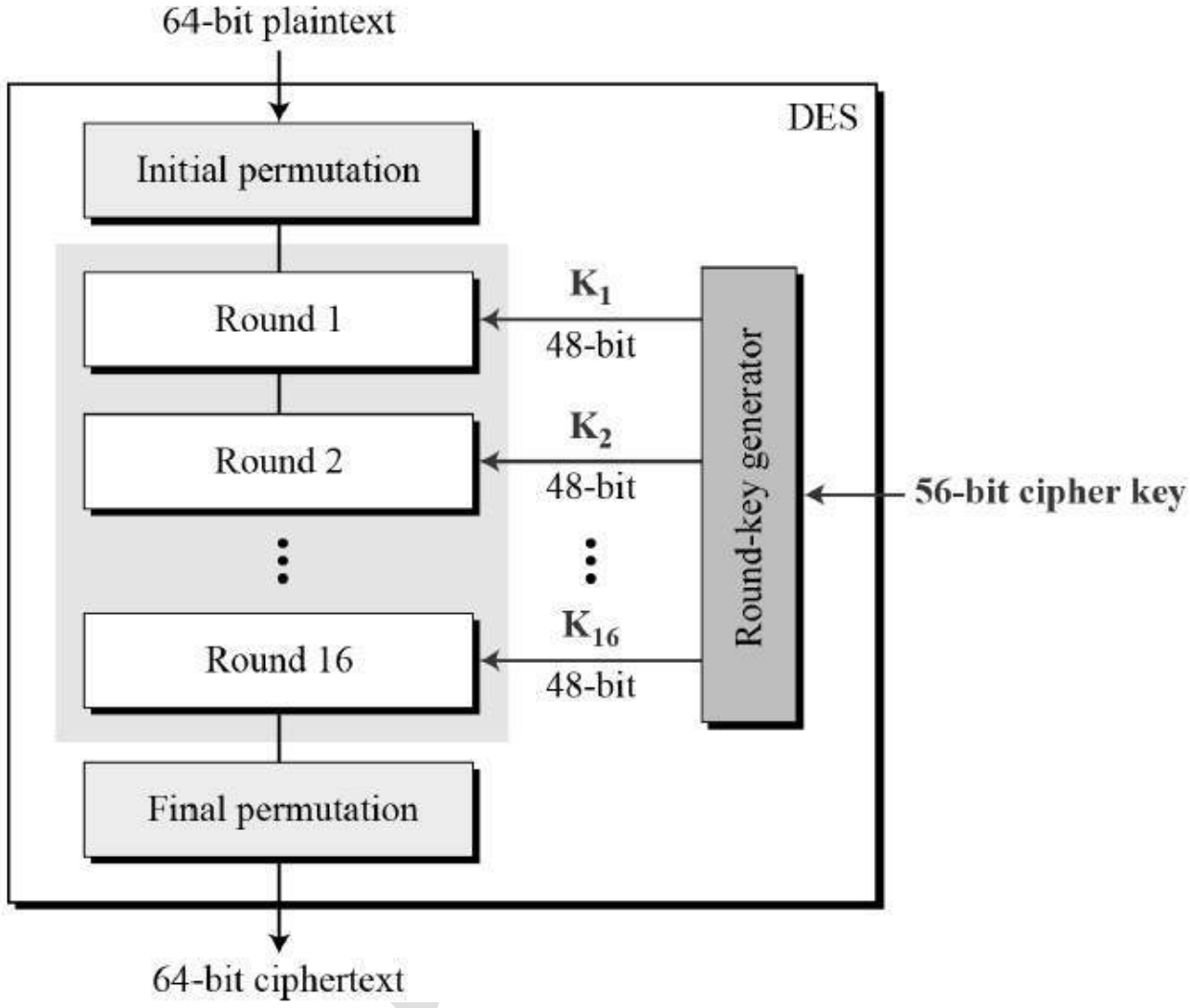
Number of Rounds

The number of rounds used in a Feistel Cipher depends on desired security from the system. More number of rounds provide more secure system. But at the same time, more rounds mean the inefficient slow encryption and decryption processes. Number of rounds in the systems thus depend upon efficiency–security tradeoff.

Data Encryption Standard

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

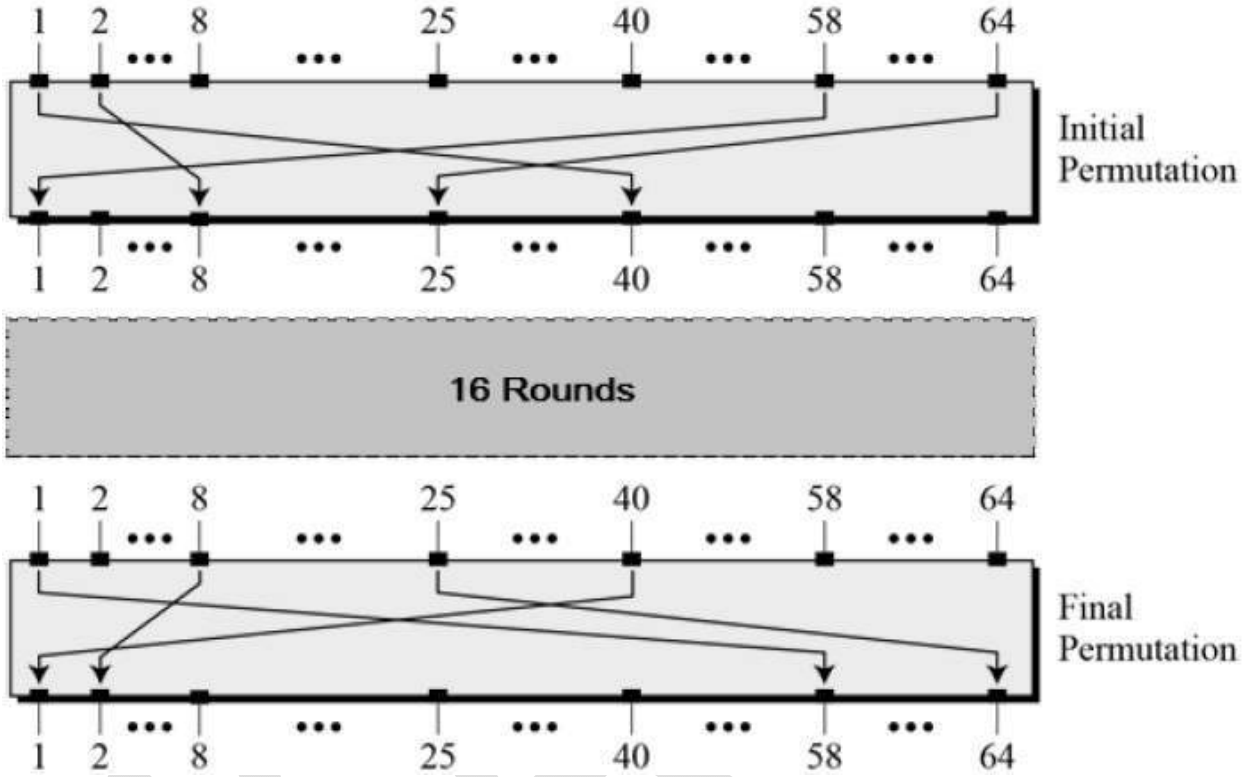
DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure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



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

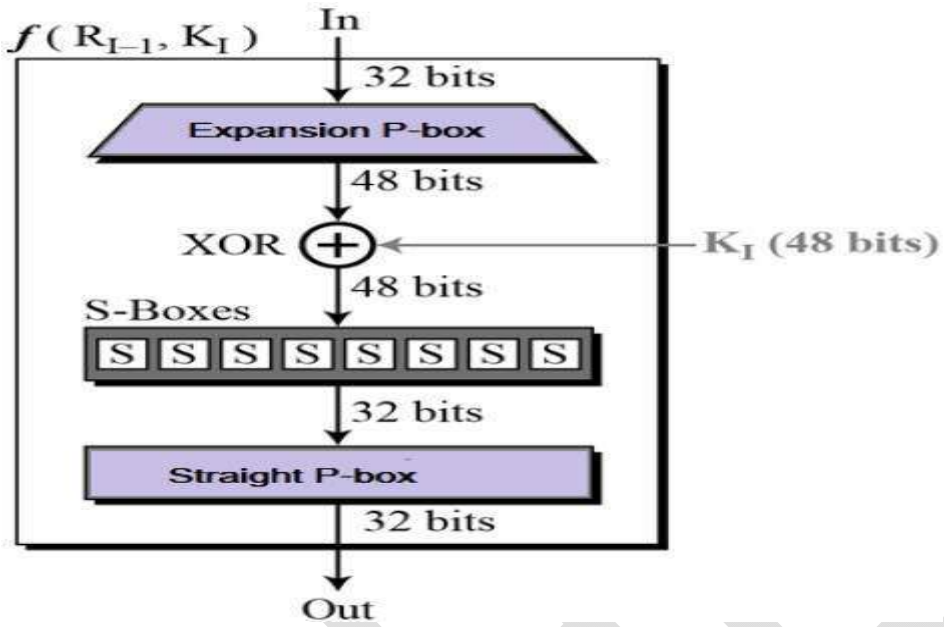
- Round function
- Key schedule
- Any additional processing − Initial and final permutation
- Initial and Final Permutation

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

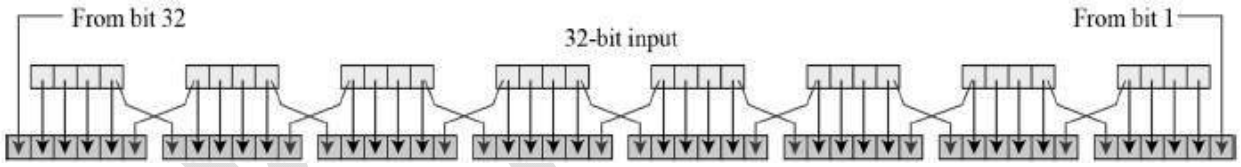


Round Function

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



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

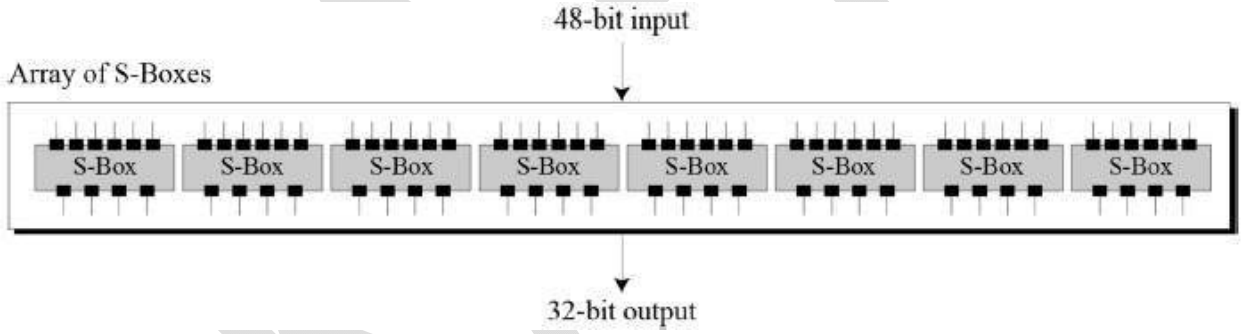


The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown –

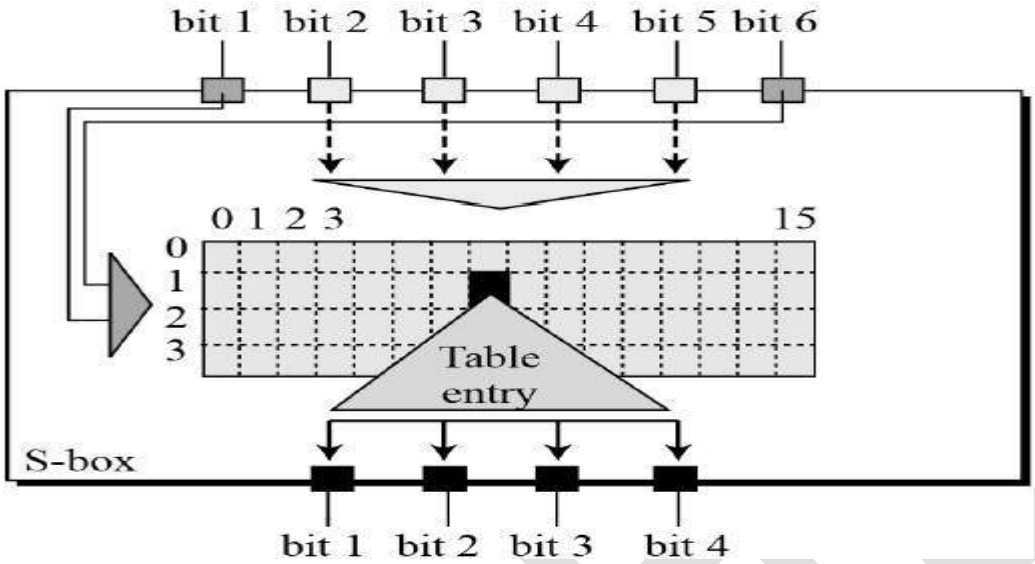
32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

XOR (Whitener). − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.

Substitution Boxes. − The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration –



The S-box rule is illustrated below –



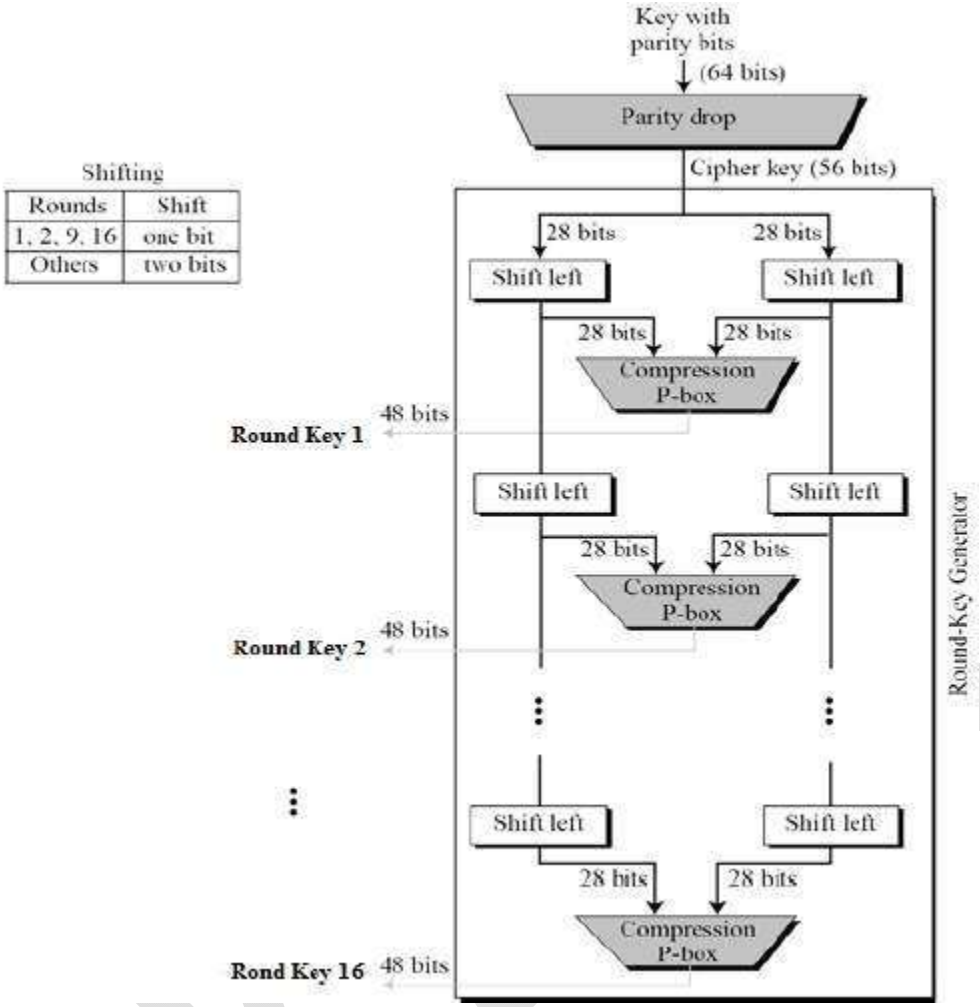
There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.

Straight Permutation − The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

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

Key Generation

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



Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

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

DES Analysis

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

- **Avalanche effect** − A small change in plaintext results in the very great change in the ciphertext.

- **Completeness** − Each bit of ciphertext depends on many bits of plaintext.

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

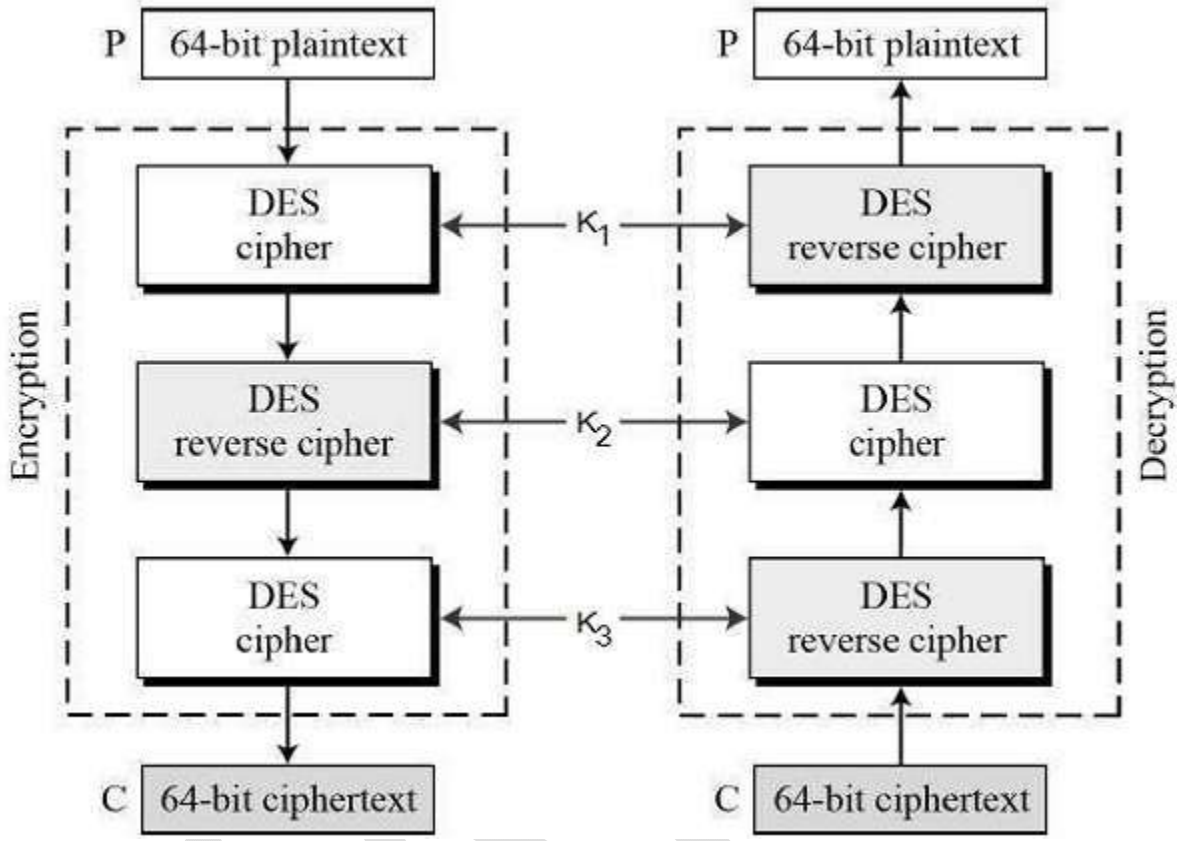
The speed of exhaustive key searches against DES after 1990 began to cause discomfort amongst users of DES. However, users did not want to replace DES as it takes an enormous amount of time and money to change encryption algorithms that are widely adopted and embedded in large security architectures.

The pragmatic approach was not to abandon the DES completely, but to change the manner in which DES is used. This led to the modified schemes of Triple DES (sometimes known as 3DES).

Incidentally, there are two variants of Triple DES known as 3-key Triple DES (3TDES) and 2-key Triple DES (2TDES).

3-KEY Triple DES

Before using 3TDES, user first generate and distribute a 3TDES key K, which consists of three different DES keys K1, K2 and K3. This means that the actual 3TDES key has length 3×56 = 168 bits. The encryption scheme is illustrated as follows –



The encryption-decryption process is as follows −

Encrypt the plaintext blocks using single DES with key K1.

- Now decrypt the output of step 1 using single DES with key K2.
- Finally, encrypt the output of step 2 using single DES with key K3.
- The output of step 3 is the ciphertext.

Decryption of a ciphertext is a reverse process. User first decrypt using K3, then encrypt with K2, and finally decrypt with K1.

Due to this design of Triple DES as an encrypt–decrypt–encrypt process, it is possible to use a 3TDES (hardware) implementation for single DES by setting K1, K2, and K3 to be the same value. This provides backwards compatibility with DES.

Second variant of Triple DES (2TDES) is identical to 3TDES except that K3is replaced by K1. In other words, user encrypts plaintext blocks with key K1, then decrypt with key K2, and finally encrypt with K1 again. Therefore, 2TDES has a key length of 112 bits.

Triple DES systems are significantly more secure than single DES, but these are clearly a much slower process than encryption using single DES.

Uses of Encryption

Encryption has long been used by militaries and governments to facilitate secret communication. Encryption can be used to protect data "at rest", such as information stored on computers and storage devices (e.g. USB flash drives).

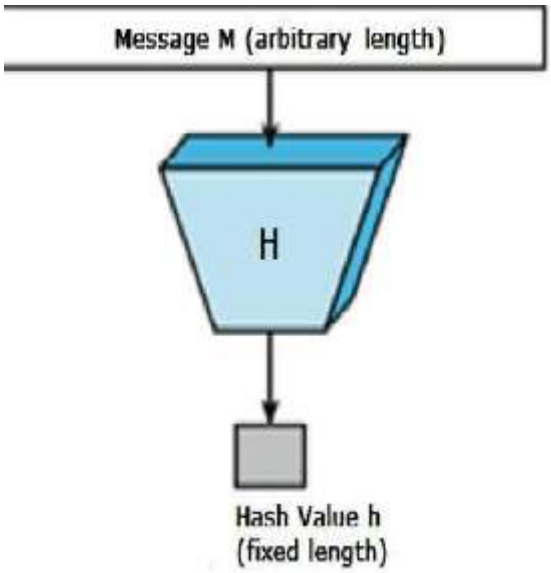
Encryption is the process of converting data to an unrecognizable or "encrypted" form. It is commonly used to protect sensitive information so that only authorized parties can view it. This includes files and storage devices, as well as data transferred over wireless networks and the Internet.

Hash Functions

Hash functions are extremely useful and appear in almost all information security applications.

A hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The input to the hash function is of arbitrary length but output is always of fixed length.

Values returned by a hash function are called message digest or simply hash values. The following picture illustrated hash function



Features of Hash Functions

1. **Fixed Length Output (Hash Value)**
   - Hash function coverts data of arbitrary length to a fixed length. This process is often referred to as hashing the data.
   - In general, the hash is much smaller than the input data, hence hash functions are sometimes called compression functions.
   - Since a hash is a smaller representation of a larger data, it is also referred to as a digest.
   - Hash function with n bit output is referred to as an n-bit hash function. Popular hash functions generate values between 160 and 512 bits.
2. **Efficiency of Operation**
   - Generally for any hash function h with input x, computation of h(x) is a fast operation.
   - Computationally hash functions are much faster than a symmetric encryption.

Properties of Hash Functions

1. **Pre-Image Resistance**
   - This property means that it should be computationally hard to reverse a hash function.
   - In other words, if a hash function h produced a hash value z, then it should be a difficult process to find any input value x that hashes to z.

- This property protects against an attacker who only has a hash value and is trying to find the input.

2. **Second Pre-Image Resistance**

- This property means given an input and its hash, it should be hard to find a different input with the same hash.
- In other words, if a hash function h for an input x produces hash value h(x), then it should be difficult to find any other input value y such that h(y) = h(x).
- This property of hash function protects against an attacker who has an input value and its hash, and wants to substitute different value as legitimate value in place of original input value.

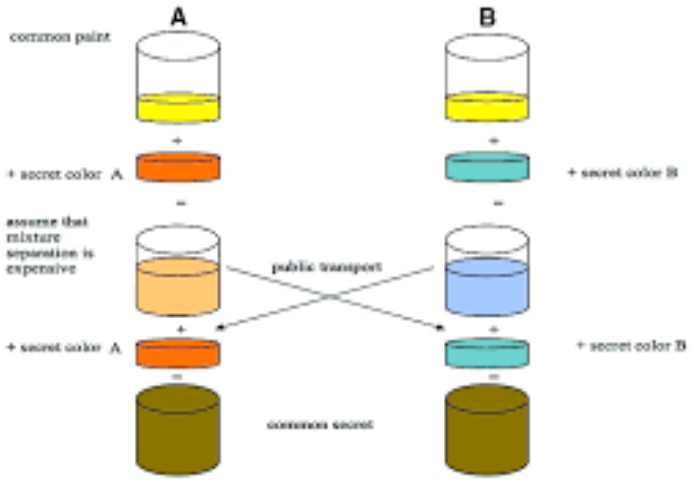
3. **Collision Resistance**

- This property means it should be hard to find two different inputs of any length that result in the same hash. This property is also referred to as collision free hash function.
- In other words, for a hash function h, it is hard to find any two different inputs x and y such that h(x) = h(y).
- Since, hash function is compressing function with fixed hash length, it is impossible for a hash function not to have collisions. This property of collision free only confirms that these collisions should be hard to find.
- This property makes it very difficult for an attacker to find two input values with the same hash.
- Also, if a hash function is collision-resistant then it is second pre-image resistant

Key Exchange

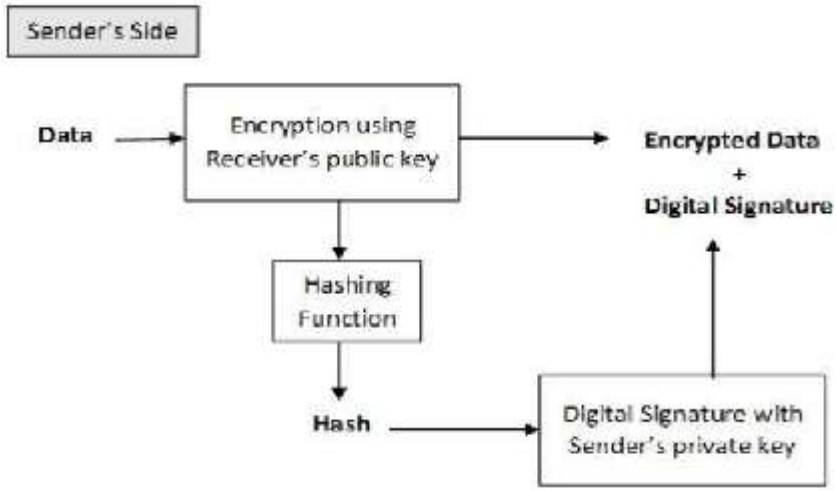
Key exchange (also key establishment) is any method in cryptography by which cryptographic keys are exchanged between two parties, allowing use of a cryptographic algorithm. ... If the cipher is a symmetric key cipher, both will need a copy of the same key.

The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.



Digital Signatures

Digital signatures are the public-key primitives of message authentication. In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind signatory to the message. Similarly, a digital signature is a technique that binds a person/entity to the digital data.



Applications

There are several reasons to implement digital signatures to communications:

1. **Authentication**

Digital signatures help to authenticate the sources of messages. For example, if a bank's branch office sends a message to central office, requesting for change in balance of an account. If the central office could not authenticate that message is sent from an authorized source, acting of such request could be a grave mistake.

2. **Integrity**

Once the message is signed, any change in the message would invalidate the signature.

3. **Non-repudiation**

By this property, any entity that has signed some information cannot at a later time deny having signed it.

Digital Certificates

Digital certificates establish credentials when performing communications and transactions on a network. Certificates contain a name, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting messages and digital signatures), and the digital signature of the certificate-issuing authority so that a recipient can verify that the certificate is real. Most digital certificates conform to a standard referred to as X.509. This

standard sets guidelines for certificate-issuing authorities to follow in order to make use of most modern applications thus making it easier to conduct legitimate transactions over a network.

The most common use of a digital certificate is to verify communication for a user whom he or she claims to be, and to provide the receiver with the means to reply through encryption. Examples of common use might include accounts for a financial institution, or an encrypted email message using PGP.

An individual or company wishing to use encryption with communication over a network would receive a digital certificate from a Certificate Authority (CA). The CA issues an encrypted digital certificate containing the recipient's public key along with extended information such as the serial number. The CA makes its own public key readily available for verification through an Internal Registration Authority or through an Internet Rregistration Authority.

The recipient of encrypted communication uses the CA's public key to decode the digital certificate attached to the message, verifies it as issued by the CA by performing a checksum of the version, the serial number, the sender, and the sender's public key. Then the recipient can verify that certificate against a Registration Authority. With this information, the recipient can respond with an encrypted session

Certificate Authorities

A Certificate Authority (CA) is a computer (or group of computers) on a network that signs and issues public keys for tansaction and message encryption. As part of a public key infrastructure, a CA checks with a Registration Authority to verify information provided by the requestor of a digital certificate. If the RA verifies the requestor's information, the CA can then issue a certificate.

Usually, this means that the CA has an arrangement with a business which provides it with information to confirm an individual's claimed identity. Certificate Authorities are a critical component in data security and electronic commerce because they guarantee that the two parties exchanging information are really who they claim to be.

Certificate Repositories

A Certificate Repository is used to propagate both certificates and the status of the issued certificates. The most commonly used repository service for certificates is a Lightweight Directory Access Protocol (LDAP) server. The CA will push certificates to the repository and the clients pull the certificates from the repository using an LDAP based client application. The LDAP implementation could be either a simple LDAP server or a full X.500 product.

The certificate status repository can take one of two forms. If Certificate Revocation Lists (CRL) or CRL Distribution Points (CDPs) are used, then a LDAP server is used. CRLs work on the concept of an administrator revoking a certificate's unique ID from the distribution list. The CRL is then pushed to the LDAP server with clients pulling the latest CRL from the server on a scheduled basis. The other technique is called On-line Certificate Status Protocol (OCSP). In this case a "cache" (some form of repository) holds the status of certificates. Each time a client application wishes to use a certificate, it will query the OCSP cache for the validity of the certificate

Secured Sockets Layer

Secured Sockets Layer (SSL) is a protocol standard that encrypts your communications over a network. The SSL protocol ensures that the information is sent, unchanged, only to the server you intended to send it to. The SSL standard is not a single protocol, but rather a set of accepted data transfer routines that uses the public and private keys as ciphers to encrypt the transmission between two computers. Online shopping sites and Financial Institutions frequently use SSL technology to safeguard account information. The first time a client accesses a service that uses SSL, the client is presented with a digital certificate. This information allows the client to determine three important things:

- The certicate has not yet expired
- The certificate was created for the computer being accessed.
- The certificate is signed by an authority the client trusts.

If any of these three conditions are not met, most applications will display a warning that explains the violations and requests permission to continue accessing. An example might be an

SSL-capable web browser. The web browser should include a list of respositories from Registration Authorities that are considered trusted. Part of the reason why you don't see any popup certificate window when you enter Amazon's secure area is that their certificate was signed by an authority known to your browser.

Security Concerns

1. **Certificate Authority Trust**: The risk of accepting unauthorized certificates lies in the hands of the user. Knowing what authorities can be trusted by a signature on the issued certificate is essential in maintaining security.

Sites such as Verisign and/or Digicert offer a database of known registrants where users can query valid certificates. This helps users in tracking down a company or organization and verifying its authenticity, but the application must support CRLs or OCSP. If it does not then the user must know to manually check for the validity of the certificate issued.

2. **Access of the private key**: Users' private keys are usually stored on a physically accessible computer. Sharing that computer with other users increases the risk of a private key being stolen. As malware and trojan programs become more sophisticated, the likelihood of a malicious attack and hijack of a user's private key increases.

One such technology that looks to overcome this issue is a token based system called Smart Card Technology. Smart Cards hold a private key along with the extended user information. Devices called Smart Card Readers are used to pass the information from the card to the application sending a token requesting the use of the key for the encrypted transaction.

3. **Registration Authority Security:** The overall security of the Registration Authority that verifies the public key being issued can be a risk. CA issues what is commonly known as a root certificate. If this root certificate is forged, then users can be re issued a public key.

A CA (Along with its RA) can revoke a certificate it has issued. This may be for a number of reasons but a prime reason would be a certificate compromise: someone has accessed the private key associated with a certificate and it needs to be revoked for security reasons. Each CA should publish a list of revoked certificates. This list is known as a Certificate Revocation

List (CRL). Normally these lists are publicly available so that a server can check it before accepting a given certificate

POSSIBLE QUESTIONS

PART A

Q.NO 1 TO 20 (MULTIPLE CHOICE QUESTIONS)

PART B (2 MARKS)

1. Define Symmetric
2. What is the need for DES?
3. Define Hash functions.
4. What are the characteristics of Hash functions?
5. Define Digital Signatures.
6. Differentiate between Symmetric and Asymmetric.
7. List out the uses of encryption
8. What is Stream cipher
9. What are the drawbacks in Digital Signatures

PART C (6 MARKS)

1. Explain the concept of digital signature.
2. Summarize the steps performed in both symmetric and Asymmetric techniques.
3. What is digital signature? Explain its use with the help of example.
4. With suitable sketched, explain the working of DES algorithm
5. Explain limitation of DES in detail
6. List and explain various key management techniques.
7. Diffie-Hellman key exchange – Discuss
8. Discuss Stream cipher and block cipher with example.

KARPAGAM ACADEMY OF HIGHER EDUCTATION

DEPARTMENT OF COMPUTER SCIENCE

III B.Sc CS

INFORMATION SECURITY[16CSU501A]

UNIT 2

S.NO	Question	Option 1	Option 2	Option 3	Option 4	Answer
1	A _____ is one which a block of plain text is treated as a whole and used to produce a ciphertext block of equal length.	stream cipher	symmetric cipher	block cipher	substitution	block cipher
2	The _____ has been the most widely used encryption algorithm until recently	DES	AES	AFS	DEM	DES
3	A _____ is one that encrypts a digital data stream one bit or one byte at a time.	block cipher	symmetric cipher	stream cipher	plain	stream cipher
4	A block cipher operates on a plain text block of n bits to produce a ciphertext block of _____ bits	n*n*	n*n	n power n	n	n
5	most symmetric block encryption algorithms in current use are based on a structure referred to as a _____ block cipher.	Feistel	block	stream	DES	Feistel
6	Feistel refers to this as the _____ cipher because it allows for the maximum number of possible encryption mappings from the plaintext block.	asymmetric	product	symmetric	ideal block	ideal block
7	_____ are said to be secret key.	asymmetric	public key	symmetric	private key	symmetric
8	Symmetric algorithms use _____ key, which works for both encrypion and decryption.	1	2	3	4	1
9	A _____ encrpypts a group of plaintext symbols as one block.	block cipher	stream cipher	symmetric cipher	caesar cipher	block cipher
10	NBS recognized that the general public needed a _____technique for protecting sencitive information.	secure encryptic	stream cipher	symmetric cipher	caesar cipher	secure encryption
11	The DES algorithm is careful and combination of two fundamental building block of encryption.	substitution & transposition	permutation	confusion	Feistel	substitution & transposition

12	A block size of _____ bits has been considered a reasonable tradeoff and was nearly universal in block cipher design	16	8	32	64	64
13	_____ key size means greater security but may decrease encryption/decryption speed	larger	smaller	medium	double	larger
14	The DES adopted in _____ by the national bureau of standards	1977	1978	1979	1980	1977
15	DES used only standard arithmetnic and logical operations on numbers up to _____ long.	64 bits	74 bits	22 bits	65 bits	64 bits
16	For DES data are encrypted in ___ blocks using a _____bit key.	8 bit,25	32bit,45	64 bit,56	16 bit,55	64 bit,56
17	The left and right halves of the output are swapped to produce the _____	preoutput	preinput	postinput	postoutput	preoutput
18	The _____ function is the same for each round but a different subkey is produced because of the repeated shifts of the key bits.	substitution	permutation	confusion	transformation	permutation
19	_____ is a popular commercial-grade encryption technique	public key encryption	key enable	private key encryption	secret key	public key encryption
20	In which year Diffie and Hellman proposed a new kind of encryption system	1976	1777	1977	1978	1976
21	AES stands for	Aavent Encryption Standard	Advanced Enable Standard	Advanced Encryption Standard	Advanced Encryption Stante	Advanced Encryption Standard
22	In a public key encryption system, each user has a key that does not have to be kept _____	secret	soundness	cost	private key	secret
23	_____ refers to the effort required to cryptanalyze an algorithm.	security	cost	randomness	product	security
24	_____ of the mathematical basis for the algorithms security	soundness	randomness	cost	product	soundness

25	A candidate algorithm shall be judged according to relative _____ of design.	permutation	complexity	random	simplicity	simplicity
26	_____ algorithms with greater flexibility will meet the needs of more user than less flexible ones.	candidate	primary	super	foreign	candidate
27	_____ refers to the ability to change keys quickly and with a minimum of resources.	key agility	key enable	key distribution	key change	key agility
28	The forward substitute byte transformation called _____ bytes	max	super	min	sub	sub
29	Public key encryption accomplish this goal by using _____ keys	2	3	4	5	2
30	Public key encryption accomplish this goal by using 2 keys _____	encrypt, decrypt	encrypt	decrypt	public, private	encrypt, decrypt
31	_____ uses an s-box to perform a byte-by-byte substitution of the block	substitution bytes	permutation butes	diffusion butes	confussion butes	substitution bytes
32	ECB stands for	electronic coin book	electronic code book	electronic code bill	email code book	electronic code book
33	CBC stands for	Cipher Block Chating	Cylinder Block Chaining	Cipher Base Chaining	Cipher Block Chaining	Cipher Block Chaining
34	CFB stands for _____	Cipher Formback	Cylinder Feedback	Cipher Front	Cipher Feedback	Cipher Feedback
35	In an asymmetric encryption system, each user has two keys	public, private	encrypt, decrypt	ceaser, rail	keyword, plaintext	public, private
36	To reduce the problem of key proliferation by using _____ apporach.	public key	private key	public & private key	preprocessing	public key
37	ATM stands for	Asynchronous transfer mode	Asynchronous teller mode	All time teller mode	Asynchronous target mode	Asynchronous transfer mode

38	_____ the execution of the underlying encryption algorithm does not depend on input of the plaintext or ciphertext.	preprocessing	software	hardware	none	preprocessing
39	A public key encryption can take _____ long as long to perform as a symmetric encryption.	10000	100	1000	10	10000
40	cryptography can be used to _____ a file.	seal	seek	soul	managem ent	seal
41	How many number of rounds are used in DES	8	16	10	12	16
42	The Secure hash algorithmsare a family of cryptographic hash functions published by the _____	NIST	NNST	NSIT	NNT	NIST
43	The round function F in DES is applied to the _____ part of the data	left	right	alternativel y	both	right
44	A small change in either plain text or the key should produce a significant change in ciphertext is called _____	permutation	confusion	avalanche effect	diffusion	avalanche effect
45	_____ algorithm is based on principle that the greatest common divisor of two numbers	RC4	Euclidean	AES	DES	Euclidean
46	_____ is a stream cipher designed in 1987 by Ron Rivest for RSA security	RC4	RC5	RC6	RC7	RC4
47	_____ encryption has two encryption stages and two keys.	Two DES	Double DES	Triple DES	DES	Double DES
48	In block encryption schemes, _____ means linking each block to the previou's block value	chaining	linking	alternative	shifting	chaining
49	In hash funciton techniques MD4, where MD stands for	Message Digest	Malware double	Medium Digest	managem ent	Message Digest
50	Secure Hash Algorithm or Standard stands for _____	SHA/SHS	SHA/HSS	HSS/ASH	HS/SS	SHA/SHS
51	Which of the following operations is required to convert 32bit right half into 48bit for the DES cipher.	permutation	substitutio n/choice	Expansion /Permutati on	64 bit	Expansion/Permut ation

52	Pre-shared public keys is so called _____	Diffle-Hell-man key exchange protocol	protocol	diffusion	private key	Diffle-Hell-man key exchange protocol
53	A digital signature is a protocol that produces the same effect as a _____	real signature	reel signature	prompt signs	digital signature	real signature
54	_____ , one bit of plain text is encrypted at a time.	stream cipher	block cipher	cipher	decrypt	stream cipher
55	A digital signature must meet _____ primary conditions	2	3	4	1	2
56	In digital signature once a check is cashed , it is cancelled so that it cannot be	used	reused	tangible object	confirm	reused
57	A public key and user's identity are bound together in a _____	certificate	digital	certificate, digital	public, private	certificate
58	What result the MOD function will produce MOD(30,7)?	3	0	2	1	2
59	Two integers a and b are said to be _____ modulo n, if (a mod n) = (b mod n).	modular	congruent	arithmetic	logical	congruent
60	Find the odd man out.	confifentiality	integrity	availability	cipher	cipher

Unit III

Program Security: Secure programs, Non malicious Program errors, Malicious codes virus, Trap doors, Salami attacks, Covert channels, Control against program. Threats: Protection in OS: Memory and Address Protection, Access control, File Protection, User Authentication.

SECURE PROGRAM

The security implies some degree of trust that the program enforces expected confidentiality, integrity, and availability. One way to assess security or quality is to ask people to name the characteristics of software that contribute to its overall security. This difference occurs because the importance of the characteristics depends on who is analyzing the software. For example, one person may decide that code is secure because it takes too long to break through its security controls. And someone else may decide code is secure if it has run for a period of time with no apparent failures. But a third person may decide that any potential fault in meeting security requirements makes code insecure.

Early work in computer security was based on the paradigm of "penetrate and patch," in which analysts searched for and repaired faults. Often, a top-quality "tiger team" would be convened to test a system's security by attempting to cause it to fail. The test was considered to be a "proof" of security; if the system withstood the attacks, it was considered secure. Unfortunately, far too often the proof became a counterexample, in which not just one but several serious security problems were uncovered. The problem discovery in turn led to a rapid effort to "patch" the system to repair or restore the security. However, the patch efforts were largely useless, making the system less secure rather than more secure because they frequently introduced new faults. There are at least four reasons.

1. The pressure to repair a specific problem encouraged a narrow focus on the fault itself and not on its context. In particular, the analysts paid attention to the immediate cause of the failure and not to the underlying design or requirements faults.

2. The fault often had non-obvious side effects in places other than the immediate area of the fault.

3. Fixing one problem often caused a failure somewhere else, or the patch addressed the problem in only one place, not in other related places.

4. The fault could not be fixed properly because system functionality or performance would suffer as a consequence.

The inadequacies of penetrate-and-patch led researchers to seek a better way to be confident that code meets its security requirements. That is, to understand program security, we can examine programs to see whether they behave as their designers intended or users expected. We call such unexpected behavior a program security flaw; it is inappropriate program behaviour caused by program vulnerability.

Program security flaws can derive from any kind of software fault. That is, they cover everything from a misunderstanding of program requirements to a one-character error in coding or even typing. The flaws can result from problems in a single code component or from the failure of several programs or program pieces to interact compatibly through a shared interface. The security flaws can reflect code that was intentionally designed or coded to be malicious or code that was simply developed in a sloppy or misguided way. Thus, it makes sense to divide program flaws into two separate logical categories: inadvertent human errors versus malicious, intentionally induced flaws.

Types of Flaws

To aid our understanding of the problems and their prevention or correction, we can define categories that distinguish one kind of problem from another. For example, a taxonomy of program flaws, dividing them first into intentional and inadvertent flaws. They further divide intentional flaws into malicious and nonmalicious ones.

In the taxonomy, the inadvertent flaws fall into six categories:

- validation error (incomplete or inconsistent): permission checks
- domain error: controlled access to data

- serialization and aliasing: program flow order
- inadequate identification and authentication: basis for authorization
- boundary condition violation: failure on first or last case
- other exploitable logic errors

NON MALICIOUS PROGRAM ERRORS

Programmers and other developers make many mistakes, most of which are unintentional and nonmalicious. Many such errors cause program malfunctions but do not lead to more serious security vulnerabilities. However, a few classes of errors have plagued programmers and security professionals for decades, and there is no reason to believe they will disappear.

Buffer Overflows

A buffer overflow is the computing equivalent of trying to pour two liters of water into a one liter pitcher: Some water is going to spill out and make a mess. And in computing, what a mess these errors have made.

Definition

A buffer (or array or string) is a space in which data can be held. A buffer resides in memory. Because memory is finite, a buffer's capacity is finite. For this reason, in many programming languages the programmer must declare the buffer's maximum size so that the compiler can set aside that amount of space.

For example - how buffer overflows can happen in C language

program contains the declaration:

```
char sample[10];
```

The compiler sets aside 10 bytes to store this buffer, one byte for each of the ten elements of the array, sample[0] through sample[9]. Now we execute the statement:

```
sample[10] = 'A';
```

The subscript is out of bounds (that is, it does not fall between 0 and 9), so we have a problem. The nicest outcome (from a security perspective) is for the compiler to detect the problem and mark the error during compilation. However, if the statement were

sample[i] = 'A';

we could not identify the problem until i was set during execution to a too-big subscript. It would be useful if, during execution, the system produced an error message warning of a subscript out of bounds. Unfortunately, in some languages, buffer sizes do not have to be predefined, so there is no way to detect an out-of-bounds error. More importantly, the code needed to check each subscript against its potential maximum value takes time and space during execution, and the resources are applied to catch a problem that occurs relatively infrequently. Even if the compiler were careful in analyzing the buffer declaration and use, this same problem can be caused with pointers, for which there is no reasonable way to define a proper limit. Thus, some compilers do not generate the code to check for exceeding bounds. Let us examine this problem more closely. It is important to recognize that the potential overflow causes a serious problem only in some instances. The problem's occurrence depends on what is adjacent to the array sample. For example, suppose each of the ten elements of the array sample is filled with the letter A and the erroneous reference uses the letter B, as follows:

for (i=0; i<=9; i++) sample[i] = 'A'; sample[10] = 'B'

All program and data elements are in memory during execution, sharing space with the operating system, other code, and resident routines. So there are four cases to consider in deciding where the 'B' goes. If the extra character overflows into the user's data space, it simply overwrites an existing variable value (or it may be written into an as-yet unused location), perhaps affecting the program's result, but affecting no other program or data.

VIRUS AND OTHER MALICIOUS CODE

The programs operate on data, taking action only when data and state changes trigger it. Much of the work done by a program is invisible to users, so they are not likely to be aware of any malicious activity. For instance, when was the last time you saw a bit? Do you know in what form a document file is stored? If you know a document resides somewhere on a disk, can you find it? Can you tell if a game program does anything in addition to its expected interaction with you? Which files are modified by a word processor when you create a document? However,

since computer data are not usually seen directly by users, malicious people can make programs serve as vehicles to access and change data and other programs.

Why Worry About Malicious Code?

Malicious code behaves in unexpected ways, thanks to a malicious programmer's intention. We think of the malicious code as lurking inside our system: all or some of a program that we are running or even a nasty part of a separate program that somehow attaches itself to another (good) program**.**

Malicious Code Can Do Much (Harm)

Malicious code can do anything any other program can, such as writing a message on a computer screen, stopping a running program, generating a sound, or erasing a stored file. Or malicious code can do nothing at all right now; it can be planted to lie dormant, undetected, until some event triggers the code to act. The trigger can be a time or date, an interval (for example, after 30 minutes), an event (for example, when a particular program is executed), a condition (for example, when communication occurs on a modem), a count (for example, the fifth time something happens), some combination of these, or a random situation. In fact, malicious code can do different things each time or nothing most of the time with something dramatic on occasion. In general, malicious code can act with all the predictability of a two year-old child.

Malicious code runs under the user's authority. Thus, malicious code can touch everything the user can touch, and in the same ways. Users typically have complete control over their own program code and data files; they can read, write, modify, append, and even delete them. And well they should. But malicious code can do the same, without the user's permission or even knowledge.

Malicious Code Has Been Around a Long Time

Malicious code is still around, and its effects are more pervasive. It is important for us to learn what it looks like and how it works, so that we can take steps to prevent it from doing damage or at least mediate its effects.

Kinds of Malicious Code

Malicious code or a rogue program is the general name for unanticipated or undesired effects in programs or program parts, caused by an agent intent on damage. This definition eliminates unintentional errors, although they can also have a serious negative effect. This definition also excludes coincidence, in which two benign programs combine for a negative effect. The agent is the writer of the program or the person who causes its distribution. By this definition, most faults found in software inspections, reviews, and testing do not qualify as malicious code, because we think of them as unintentional.

You are likely to have been affected by a virus at one time or another, either because your computer was infected by one or because you could not access an infected system while its administrators were cleaning up the mess one made. In fact, your virus might actually have been a worm: The terminology of malicious code is sometimes used imprecisely. A virus is a program that can pass on malicious code to other nonmalicious programs by modifying them. The term "virus" was coined because the affected program acts like a biological virus: It infects other healthy subjects by attaching itself to the program and either destroying it or coexisting with it. Because viruses are insidious, we cannot assume that a clean program yesterday is still clean today. Moreover, a good program can be modified to include a copy of the virus program, so the infected good program itself begins to act as a virus, infecting other programs. The infection usually spreads at a geometric rate, eventually overtaking an entire computing system and spreading to all other connected systems.

A virus can be either transient or resident. A transient virus has a life that depends on the life of its host; the virus runs when its attached program executes and terminates when its attached program ends. (During its execution, the transient virus may have spread its infection to other programs.) A resident virus locates itself in memory; then it can remain active or be activated as a stand-alone program, even after its attached program ends.

A **Trojan horse** is malicious code that, in addition to its primary effect, has a second, nonobvious malicious effect.

A **logic bomb** is a class of malicious code that "detonates" or goes off when a specified condition occurs. A time bomb is a logic bomb whose trigger is a time or date.

A **trapdoor** or **backdoor** is a feature in a program by which someone can access the program other than by the obvious, direct call, perhaps with special privileges. For instance, an automated bank teller program might allow anyone entering the number 990099 on the keypad to process the log of everyone's transactions at that machine. In this example, the trapdoor could be intentional, for maintenance purposes, or it could be an illicit way for the implementer to wipe out any record of a crime.

A **worm** is a program that spreads copies of itself through a network. The primary difference between a worm and a virus is that a worm operates through networks, and a virus can spread through any medium (but usually uses copied program or data files). Additionally, the worm spreads copies of itself as a stand-alone program, whereas the virus spreads copies of itself as a program that attaches to or embeds in other programs.

These definitions match current careful usage. The distinctions among these terms are small, and often the terms are confused. The term "virus" is often used to refer to any piece of malicious code. Furthermore, two or more forms of malicious code can be combined to produce a third kind of problem. For instance, a virus can be a time bomb if the viral code that is spreading will trigger an event after a period of time has passed.

The **rabbit** as a virus or worm that self-replicates without bound, with the intention of exhausting some computing resource. A rabbit might create copies of it and store them on disk, in an effort to completely fill the disk

The kinds of malicious code are summarized in Table

Code Type	Characteristics
Virus	Attaches itself to program and propagates copies of itself to other programs
Trojan horse	Contains unexpected, additional functionality
Logic bomb	Triggers action when condition occurs

Time bomb	Triggers action when specified time occurs
Trapdoor	Allows unauthorized access to functionality
Worm	Propagates copies of itself through a network
Rabbit	Replicates itself without limit to exhaust resource

Because "virus" is the popular name given to all forms of malicious code and because fuzzy lines exist between different kinds of malicious code.

How Viruses Attach

A printed copy of a virus does nothing and threatens no one. Even executable virus code sitting on a disk does nothing. For a virus to do its malicious work and spread itself, it must be activated by being executed. Fortunately for virus writers, but unfortunately for the rest of us, there are many ways to ensure that programs will be executed on a running computer.

For example, recall the SETUP program that you initiate on your computer. It may call dozens or hundreds of other programs, some on the distribution medium, some already residing on the computer, some in memory. If any one of these programs contains a virus, the virus code could be activated. Suppose the virus code were in a program on the distribution medium, such as a CD; when executed, the virus could install itself on a permanent storage medium (typically, a hard disk), and also in any and all executing programs in memory.

A more common means of virus activation is as an attachment to an e-mail message. In this attack, the virus writer tries to convince the victim (the recipient of an e-mail message) to open the attachment. Once the viral attachment is opened, the activated virus can do its work. Some modern e-mail handlers, in a drive to "help" the receiver (victim), will automatically open attachments as soon as the receiver opens the body of the e-mail message. The virus can be executable code embedded in an executable attachment, but other types of files are equally dangerous. For example, objects such as graphics or photo images can contain code to be executed by an editor, so they can be transmission agents for viruses. In general, it is safer to

force users to open files on their own rather than automatically; it is a bad idea for programs to perform potentially security-relevant actions without a user's consent.

Appended Viruses

A program virus attaches itself to a program; then, whenever the program is run, the virus is activated. This kind of attachment is usually easy to program. In the simplest case, a virus inserts a copy of itself into the executable program file before the first executable instruction. Then, all the virus instructions execute first; after the last virus instruction, control flows naturally to what used to be the first program instruction.

Virus Appended to a Program

This kind of attachment is simple and usually effective. The virus writer does not need to know anything about the program to which the virus will attach, and often the attached program simply serves as a carrier for the virus. The virus performs its task and then transfers to the original program. Typically, the user is unaware of the effect of the virus if the original program still does all that it used to. Most viruses attach in this manner.

Viruses That Surround a Program

An alternative to the attachment is a virus that runs the original program but has control before and after its execution. For example, a virus writer might want to prevent the virus from being detected. If the virus is stored on disk, its presence will be given away by its file name, or its size will affect the amount of space used on the disk. The virus writer might arrange for the virus to attach itself to the program that constructs the listing of files on the disk. If the virus regains control after the listing program has generated the listing but before the listing is displayed or printed, the virus could eliminate its entry from the listing and falsify space counts so that it appears not to exist.

Integrated Viruses and Replacements

A third situation occurs when the virus replaces some of its target, integrating itself into the original code of the target. Clearly, the virus writer has to know the exact structure of the original program to know where to insert which pieces of the virus.

Virus Integrated into a Program.

Finally, the virus can replace the entire target, either mimicking the effect of the target or ignoring the expected effect of the target and performing only the virus effect. In this case, the user is most likely to perceive the loss of the original program.

Document Viruses

Currently, the most popular virus type is what we call the document virus, which is implemented within a formatted document, such as a written document, a database, a slide presentation, or a spreadsheet. These documents are highly structured files that contain both data (words or numbers) and commands (such as formulas, formatting controls, links). The commands are part of a rich programming language, including macros, variables and procedures, file accesses, and even system calls. The writer of a document virus uses any of the features of the programming language to perform malicious actions.

The ordinary user usually sees only the content of the document (its text or data), so the virus writer simply includes the virus in the commands part of the document, as in the integrated program virus.

How Viruses Gain Control

The virus (V) has to be invoked instead of the target (T). Essentially, the virus either has to seem to be T, saying effectively "I am T" (like some rock stars, where the target is the artiste formerly known as T) or the virus has to push T out of the way and become a substitute for T, saying effectively "Call me instead of T." A more blatant virus can simply say "invoke me [you fool]." The virus can assume T's name by replacing (or joining to) T's code in a file structure; this invocation technique is most appropriate for ordinary programs. The virus can overwrite T in storage (simply replacing the copy of T in storage, for example). Alternatively, the virus can

change the pointers in the file table so that the virus is located instead of T whenever T is accessed through the file system.

Virus Completely Replacing a Program.

The virus can supplant T by altering the sequence that would have invoked T to now invoke the virus V; this invocation can be used to replace parts of the resident operating system by modifying pointers to those resident parts, such as the table of handlers for different kinds of interrupts.

Homes for Viruses

- The virus writer may find these qualities appealing in a virus:
- It is hard to detect.
- It is not easily destroyed or deactivated.
- It spreads infection widely.
- It can reinfect its home program or other programs.
- It is easy to create.
- It is machine independent and operating system independent.

Few viruses meet all these criteria. The virus writer chooses from these objectives when deciding what the virus will do and where it will reside.**One-Time Execution**

The majority of viruses today execute only once, spreading their infection and causing their effect in that one execution. A virus often arrives as an e-mail attachment of a document virus. It is executed just by being opened.

Boot Sector Viruses

A special case of virus attachment, but formerly a fairly popular one, is the so-called boot sector virus. When a computer is started, control begins with firmware that determines which hardware components are present, tests them, and transfers control to an operating system. A given hardware platform can run many different operating systems, so the operating system is

not coded in firmware but is instead invoked dynamically, perhaps even by a user's choice, after the hardware test.

The operating system is software stored on disk. Code copies the operating system from disk to memory and transfers control to it; this copying is called the bootstrap (often boot) load because the operating system figuratively pulls itself into memory by its bootstraps. The firmware does its control transfer by reading a fixed number of bytes from a fixed location on the disk (called the boot sector) to a fixed address in memory and then jumping to that address (which will turn out to contain the first instruction of the bootstrap loader). The bootstrap loader then reads into memory the rest of the operating system from disk. To run a different operating system, the user just inserts a disk with the new operating system and a bootstrap loader. When the user reboots from this new disk, the loader there brings in and runs another operating system. This same scheme is used for personal computers, workstations, and large mainframes.

To allow for change, expansion, and uncertainty, hardware designers reserve a large amount of space for the bootstrap load. The boot sector on a PC is slightly less than 512 bytes, but since the loader will be larger than that, the hardware designers support "chaining," in which each block of the bootstrap is chained to (contains the disk location of) the next block. This chaining allows big bootstraps but also simplifies the installation of a virus. The virus writer simply breaks the chain at any point, inserts a pointer to the virus code to be executed, and reconnects the chain after the virus has been installed.

Boot Sector Virus Relocating Code.

The boot sector is an especially appealing place to house a virus. The virus gains control very early in the boot process, before most detection tools are active, so that it can avoid, or at least complicate, detection. The files in the boot area are crucial parts of the operating system. Consequently, to keep users from accidentally modifying or deleting them with disastrous results, the operating system makes them "invisible" by not showing them as part of a normal listing of stored files, preventing their deletion. Thus, the virus code is not readily noticed by users.

Memory-Resident Viruses

Some parts of the operating system and most user programs execute, terminate, and disappear, with their space in memory being available for anything executed later. For very frequently used parts of the operating system and for a few specialized user programs, it would take too long to reload the program each time it was needed. Such code remains in memory and is called "resident" code. Examples of resident code are the routine that interprets keys pressed on the keyboard, the code that handles error conditions that arise during a program's execution, or a program that acts like an alarm clock, sounding a signal at a time the user determines. Resident routines are sometimes called TSRs or "terminate and stay resident" routines.

Virus writers also like to attach viruses to resident code because the resident code is activated many times while the machine is running. Each time the resident code runs, the virus does too. Once activated, the virus can look for and infect uninfected carriers. For example, after activation, a boot sector virus might attach itself to a piece of resident code. Then, each time the virus was activated it might check whether any removable disk in a disk drive was infected and, if not, infect it. In this way the virus could spread its infection to all removable disks used during the computing session.

Other Homes for Viruses

A virus that does not take up residence in one of these cozy establishments has to fend more for itself. But that is not to say that the virus will go homeless.

One popular home for a virus is an application program. Many applications, such as word processors and spreadsheets, have a "macro" feature, by which a user can record a series of commands and repeat them with one invocation. Such programs also provide a "startup macro" that is executed every time the application is executed. A virus writer can create a virus macro that adds itself to the startup directives for the application. It also then embeds a copy of itself in data files so that the infection spreads to anyone receiving one or more of those files.

Libraries are also excellent places for malicious code to reside. Because libraries are used by many programs, the code in them will have a broad effect. Additionally, libraries are often shared among users and transmitted from one user to another, a practice that spreads the infection. Finally, executing code in a library can pass on the viral infection to other transmission media. Compilers, loaders, linkers, runtime monitors, runtime debuggers, and even virus control programs are good candidates for hosting viruses because they are widely shared.

Virus Signatures

A virus cannot be completely invisible. Code must be stored somewhere, and the code must be in memory to execute. Moreover, the virus executes in a particular way, using certain methods to spread. Each of these characteristics yields a telltale pattern, called a signature, that can be found by a program that knows to look for it. The virus's signature is important for creating a program, called a virus scanner, that can automatically detect and, in some cases, remove viruses. The scanner searches memory and long-term storage, monitoring execution and watching for the telltale signatures of viruses. For example, a scanner looking for signs of the Code Red worm can look for a pattern containing the following characters:

/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN When the scanner recognizes a known virus's pattern, it can then block the virus, inform the user, and deactivate or remove the virus. However, a virus scanner is effective only if it has been kept up-to-date with the latest information on current viruses.

Storage Patterns

Most viruses attach to programs that are stored on media such as disks. The attached virus piece is invariant, so that the start of the virus code becomes a detectable signature. The attached piece is always located at the same position relative to its attached file. For example, the virus might always be at the beginning, 400 bytes from the top, or at the bottom of the infected

file. Most likely, the virus will be at the beginning of the file, because the virus writer wants to obtain control of execution before the bona fide code of the infected program is in charge. In the simplest case, the virus code sits at the top of the program, and the entire virus does its malicious duty before the normal code is invoked. In other cases, the virus infection consists of only a handful of instructions that point or jump to other, more detailed instructions elsewhere. For example, the infected code may consist of condition testing and a jump or call to a separate virus module. In either case, the code to which control is transferred will also have a recognizable pattern.

Recognizable Patterns in Viruses.

A virus may attach itself to a file, in which case the file's size grows. Or the virus may obliterate all or part of the underlying program, in which case the program's size does not change but the program's functioning will be impaired. The virus writer has to choose one of these detectable effects.

The virus scanner can use a code or checksum to detect changes to a file. It can also look for suspicious patterns, such as a JUMP instruction as the first instruction of a system program (in case the virus has positioned itself at the bottom of the file but wants to be executed first).

Execution Patterns

A virus writer may want a virus to do several things at the same time, namely, spread infection, avoid detection, and cause harm. These goals are shown in Table, along with ways each goal can be addressed. Unfortunately, many of these behaviors are perfectly normal and might otherwise go undetected. For instance, one goal is modifying the file directory; many normal programs create files, delete files, and write to storage media. Thus, there are no key signals that point to the presence of a virus.

Most virus writers seek to avoid detection for themselves and their creations. Because a disk's boot sector is not visible to normal operations (for example, the contents of the boot sector do not show on a directory listing), many virus writers hide their code there. A resident virus can monitor disk accesses and fake the result of a disk operation that would show the virus hidden in

a boot sector by showing the data that should have been in the boot sector (which the virus has moved elsewhere).

There are no limits to the harm a virus can cause. On the modest end, the virus might do nothing; some writers create viruses just to show they can do it. Or the virus can be relatively benign, displaying a message on the screen, sounding the buzzer, or playing music. From there, the problems can escalate. One virus can erase files, an entire disk; one virus can prevent a computer from booting, and another can prevent writing to disk. The damage is bounded only by the creativity of the virus's author.

Virus Effects and Causes

Virus Effect	**How It Is Caused**
Attach to executable program	• Modify file directory • Write to executable program file
Attach to data or control file	• Modify directory • Rewrite data • Append to data• • Append data to self
Remain in memory handler address table	• Intercept interrupt by modifying interrupt • Load self in nontransient memory area
Infect disks	• Intercept interrupt • Intercept operating system call (to format disk, for example) • Modify system file • Modify ordinary executable program
Conceal self falsify result	• Intercept system calls that would reveal and self Classify self as "hidden" file

Spread infection	• Infect boot sector • Infect systems program • Infect ordinary program • Infect data ordinary program reads to control its execution
Prevent deactivation deactivation	• Activate before deactivating program and block Store copy to reinfect after deactivation

Viruses and Other Malicious Code Transmission Patterns

A virus is effective only if it has some means of transmission from one location to another. As we have already seen, viruses can travel during the boot process, by attaching to an executable file or traveling within data files. The travel itself occurs during execution of an already infected program. Since a virus can execute any instructions a program can, virus travel is not confined to any single medium or execution pattern. For example, a virus can arrive on a diskette or from a network connection, travel during its host's execution to a hard disk boot sector, reemerge next time the host computer is booted, and remain in memory to infect other diskettes as they are accessed.

Polymorphic Viruses

The virus signature may be the most reliable way for a virus scanner to identify a virus. If a particular virus always begins with the string 47F0F00E08 (in hexadecimal) and has string 00113FFF located at word 12, it is unlikely that other programs or data files will have these exact characteristics. For longer signatures, the probability of a correct match increases.

If the virus scanner will always look for those strings, then the clever virus writer can cause something other than those strings to be in those positions. For example, the virus could have two alternative but equivalent beginning words; after being installed, the virus will choose one of the two words for its initial word. Then, a virus scanner would have to look for both patterns. A virus that can change its appearance is called a polymorphic virus. (Poly means "many" and morph means "form".) A two-form polymorphic virus can be handled easily as two independent viruses. Therefore, the virus writer intent on preventing detection of the virus will want either a large or an unlimited number of forms so that the number of possible forms is too large for a virus scanner to search for. Simply embedding a random number or string at a fixed place in the executable version of a virus is not sufficient, because the signature of the virus is just the constant code excluding the random part. A polymorphic virus has to randomly reposition all parts of itself and randomly change all fixed data. Thus, instead of containing the fixed (and therefore searchable) string "HA! INFECTED BY A VIRUS," a polymorphic virus has to change even that pattern sometimes.

Trivially, assume a virus writer has 100 bytes of code and 50 bytes of data. To make two virus instances different, the writer might distribute the first version as 100 bytes of code followed by all 50 bytes of data. A second version could be 99 bytes of code, a jump instruction, 50 bytes of data, and the last byte of code. Other versions are 98 code bytes jumping to the last two, 97 and three, and so forth. Just by moving pieces around the virus writer can create enough different appearances to fool simple virus scanners. Once the scanner writers became aware of these kinds of tricks, however, they refined their signature definitions.

A more sophisticated polymorphic virus randomly intersperses harmless instructions throughout its code. Examples of harmless instructions include addition of zero to a number, movement of a data value to its own location, or a jump to the next instruction. These "extra" instructions make it more difficult to locate an invariant signature.

A simple variety of polymorphic virus uses encryption under various keys to make the stored form of the virus different. These are sometimes called encrypting viruses. This type of

virus must contain three distinct parts: a decryption key, the (encrypted) object code of the virus, and the (unencrypted) object code of the decryption routine. For these viruses, the decryption routine itself or a call to a decryption library routine must be in the clear, and so that becomes the signature. To avoid detection, not every copy of a polymorphic virus has to differ from every other copy. If the virus changes occasionally, not every copy will match a signature of every other copy

The Source of Viruses

Since a virus can be rather small, its code can be "hidden" inside other larger and more complicated programs. Two hundred lines of a virus could be separated into one hundred packets of two lines of code and a jump each; these one hundred packets could be easily hidden inside a compiler, a database manager, a file manager, or some other large utility.

Virus discovery could be aided by a procedure to determine if two programs are equivalent. However, theoretical results in computing are very discouraging when it comes to the complexity of the equivalence problem. The general question, "are these two programs equivalent?" is undecidable (although that question can be answered for many specific pairs of programs). Even ignoring the general undecidability problem, two modules may produce subtly different results that may—or may not—be security relevant. One may run faster, or the first may use a temporary file for work space whereas the second performs all its computations in memory. These differences could be benign, or they could be a marker of an infection. Therefore, we are unlikely to develop a screening program that can separate infected modules from uninfected ones.

Although the general is dismaying, the particular is not. If we know that a particular virus may infect a computing system, we can check for it and detect it if it is there. Having found the virus, however, we are left with the task of cleansing the system of it. Removing the virus in a running system requires being able to detect and eliminate its instances faster than it can spread.

Prevention of Virus Infection

The only way to prevent the infection of a virus is not to share executable code with an infected source. This philosophy used to be easy to follow because it was easy to tell if a file was executable or not. For example, on PCs, a .exe extension was a clear sign that the file was executable. However, as we have noted, today's files are more complex, and a seemingly nonexecutable file may have some executable code buried deep within it. For example, a word processor may have commands within the document file; as we noted earlier, these commands, called macros, make it easy for the user to do complex or repetitive things. But they are really executable code embedded in the context of the document. Similarly, spreadsheets, presentation slides, and other office- or business-related files can contain code or scripts that can be executed in various ways—and thereby harbor viruses. And, as we have seen, the applications that run or use these files may try to be helpful by automatically invoking the executable code, whether you want it run or not! Against the principles of good security, e-mail handlers can be set to automatically open (without performing access control) attachments or embedded code for the recipient, so your e-mail message can have animated bears dancing across the top.

Another approach virus writers have used is a little-known feature in the Microsoft file design. Although a file with a .doc extension is expected to be a Word document, in fact, the true document type is hidden in a field at the start of the file. This convenience ostensibly helps a user who inadvertently names a Word document with a .ppt (Power-Point) or any other extension. In some cases, the operating system will try to open the associated application but, if that fails, the system will switch to the application of the hidden file type. So, the virus writer creates an executable file, names it with an inappropriate extension, and sends it to the victim, describing it is as a picture or a necessary code add-in or something else desirable. The unwitting recipient opens the file and, without intending to, executes the malicious code.

More recently, executable code has been hidden in files containing large data sets, such as pictures or read-only documents. These bits of viral code are not easily detected by virus scanners and certainly not by the human eye. For example, a file containing a photograph may be

highly granular; if every sixteenth bit is part of a command string that can be executed, then the virus is very difficult to detect.

Nevertheless, there are several techniques for building a reasonably safe community for electronic contact, including the following:

- Use only commercial software acquired from reliable, well established vendors
- Test all new software on an isolated computer.
- Open attachments only when you know them to be safe.
- Make a recoverable system image and store it safely
- Make and retain backup copies of executable system files.
- Use virus detectors (often called virus scanners) regularly and update them daily.

TARGETED MALICIOUS PROGRAM

Another class of malicious code is written for a particular system, for a particular application, and for a particular purpose. Many of the virus writers' techniques apply, but there are also some new ones.

Trapdoors

A trapdoor is an undocumented entry point to a module. The trapdoor is inserted during code development, perhaps to test the module, to provide "hooks" by which to connect future modifications or enhancements or to allow access if the module should fail in the future. In addition to these legitimate uses, trapdoors can allow a programmer access to a program once it is placed in production

Salami Attack

An attack known as a salami attack. This approach gets its name from the way odd bits of meat and fat are fused together in a sausage or salami. In the same way, a salami attack merges bits of seemingly inconsequential data to yield powerful results. For example, programs often disregard small amounts of money in their computations, as when there are fractional pennies as interest or tax is calculated.

Such programs may be subject to a salami attack, because the small amounts are shaved from each computation and accumulated elsewhere —such as the programmer's bank account! The shaved amount is so small that an individual case is unlikely to be noticed, and the accumulation can be done so that the books still balance overall. However, accumulated amounts can add up to a tidy sum, supporting a programmer's early retirement or new car. It is often the resulting expenditure, not the shaved amounts, that gets the attention of the authorities.

Covert Channels: Programs That Leak Information

The communication travels unnoticed, accompanying other, perfectly proper, communications. The general name for these extraordinary paths of communication is covert channels.

Suppose a group of students is preparing for an exam for which each question has four choices (a, b, c, d); one student in the group, Sophie, understands the material perfectly and she agrees to help the others. She says she will reveal the answers to the questions, in order, by coughing once for answer "a," sighing for answer "b," and so forth. Sophie uses a communications channel that outsiders may not notice; her communications are hidden in an open channel. This communication is a human example of a covert channel.

Timing Channels

Other covert channels, called timing channels, pass information by using the speed at which things happen. Actually, timing channels are shared resource channels in which the shared resource is time.

A service program uses a timing channel to communicate by using or not using an assigned amount of computing time. In the simple case, a multiprogrammed system with two user processes divides time into blocks and allocates blocks of processing alternately to one process and the other. A process is offered processing time, but if the process is waiting for another event to occur and has no processing to do, it rejects the offer. The service process either uses its block (to signal a 1) or rejects its block (to signal a 0)

CONTROL AGAINST PROGRAM THREAT

There are many ways a program can fail and many ways to turn the underlying faults into security failures. It is of course better to focus on prevention than cure; how do we use controls during software development—the specifying, designing, writing, and testing of the program.

Developmental Controls

Many controls can be applied during software development to ferret out and fix problems. So let us begin by looking at the nature of development itself, to see what tasks are involved in specifying, designing, building, and testing software. The Nature of Software Development Software development is often considered a solitary effort; a programmer sits with a specification or design and grinds out line after line of code. But in fact, software development is a collaborative effort, involving people with different skill sets who combine their expertise to produce a working product. Development requires people who can

- specify the system, by capturing the requirements and building a model of how the system should work from the users' point of view
- design the system, by proposing a solution to the problem described by the requirements and building a model of the solution
- implement the system, by using the design as a blueprint for building a working solution
- test the system, to ensure that it meets the requirements and implements the solution as called for in the design
- review the system at various stages, to make sure that the end products are consistent with the specification and design models
- document the system, so that users can be trained and supported

- manage the system, to estimate what resources will be needed for development and to track when the system will be done
- maintain the system, tracking problems found, changes needed, and changes made, and evaluating their effects on overall quality and functionality

One person could do all these things. But more often than not, a team of developers works together to perform these tasks. Sometimes a team member does more than one activity; a tester can take part in a requirements review, for example, or an implementer can write documentation. Each team is different, and team dynamics play a large role in the team's success. We can examine both product and process to see how each contributes to quality and in particular to security as an aspect of quality. Let us begin with the product, to get a sense of how we recognize high quality secure software.

Modularity, Encapsulation, and Information Hiding

Code usually has a long shelf -life, and it is enhanced over time as needs change and faults are found and fixed. For this reason, a key principle of software engineering is to create a design or code in small, self-contained units, called components or modules; when a system is written this way, we say that it is modular. Modularity offers advantages for program development in general and security in particular. If a component is isolated from the effects of other components, then it is easier to trace a problem to the fault that caused it and to limit the damage the fault causes. It is also easier to maintain the system, since changes to an isolated component do not affect other components. And it is easier to see where vulnerabilities may lie if the component is isolated. We call this isolation encapsulation.

Information hiding is another characteristic of modular software. When information is hidden, each component hides its precise implementation or some other design decision from the others. Thus, when a change is needed, the overall design can remain intact while only the necessary changes are made to particular components.

PROTECTION IN GENERAL PURPOSE OPERATING SYSTEM PROTECTED OBJECT AND METHOD OF PROTECTION MEMORY AND ADDRESS PROTECTION

Protected objects

The rise of multiprogramming meant that several aspects of a computing system required protection:

- memory
- sharable I/O devices, such as disks
- serially reusable I/O devices, such as printers and tape drives
- sharable programs and subprocedures
- networks
- sharable data

As it assumed responsibility for controlled sharing, the operating system had to protect these objects.

Security in operating system

The basis of protection is separation: keeping one user's objects separate from other users.Rushby and Randell noted that separation in an operating system can occur in several ways:

physical separation - in which different processes use different physical objects, such as separate printers for output requiring different levels of security

temporal separation - in which processes having different security requirements are executed at different times

logical separation - in which users operate under the illusion that no other processes exist, as when an operating system constrains a program's accesses so that the program cannot access objects outside its permitted domain

cryptographic separation - in which processes conceal their data and computations in such a way that they are unintelligible to outside processes. Of course, combinations of two or more of

these forms of separation are also possible. The categories of separation are listed roughly in increasing order of complexity to implement, and, for the first three, in decreasing order of the security provided. However, the first two approaches are very stringent and can lead to poor resource utilization. Therefore, we would like to shift the burden of protection to the operating system to allow concurrent execution of processes having different security needs. But separation is only half the answer. We want to separate users and their objects, but we also want to be able to provide sharing for some of those objects. For example, two users with different security levels may want to invoke the same search algorithm or function call.

The users would be able to share the algorithms and functions without compromising their individual security needs. An operating system can support separation and sharing in several ways, offering protection at any of several levels.

Do not protect- Operating systems with no protection are appropriate when sensitive procedures are being run at separate times.

Isolate - . When an operating system provides isolation, different processes running concurrently are unaware of the presence of each other. Each process has its own address space, files, and other objects. The operating system must confine each process somehow so that the objects of the other processes are completely concealed.

Share all or share nothing - With this form of protection, the owner of an object declares it to be public or private. A public object is available to all users, whereas a private object is available only to its owner.

Share via access limitation- With protection by access limitation, the operating system checks the allow ability of each user's potential access to an object. That is, access control is implemented for a specific user and a specific object. Lists of acceptable actions guide the operating system in determining whether a particular user should have access to a particular object. In some sense, the operating guard between users and objects, ensuring that only authorized accesses occurs.

Share by capabilities - An extension of limited access sharing, this form of protection allows dynamic creation of sharing rights for objects. The degree of sharing can depend on the owner or the subject, on the context of the computation, or on the object itself.

Limit use of an object. This form of protection limits not just the access to an object but the use made of that object after it has been accessed. For example, a user may be allowed to view a sensitive document, but not to print a copy of it. More powerfully, a user may be allowed access to data in a database to derive statistical summaries (such as average salary at a particular grade level), but not to determine specific data values (salaries of individuals).

Methods of memory protection

Memory protection is a way to control memory access rights on a computer, and is a part of most modern operating systems. The main purpose of memory protection is to prevent a process from accessing memory that has not been allocated to it. This prevents a bug within a process from affecting other processes, or the operating system itself, and instead results in a segmentation faultorstorage violation exception being sent to the offending process, generally causing abnormal termination (killing the process). Memory protection for computer security includes additional techniques such as address space layout randomization and executable space protection.

Segmentation

Segmentation refers to dividing a computer's memory into segments. A reference to a memory location includes a value that identifies a segment and an offset within that segment. The x86 architecture has multiple segmentation features, which are helpful for using protected memory on this architecture. On the x86 processor architecture, the Global Descriptor Table and Local Descriptor Tables can be used to reference segments in the computer's memory. Pointers to memory segments on x86 processors can also be stored in the processor's segment registers. Initially x86 processors had 4 segment registers, CS (code segment), SS (stack segment), DS (data segment) and ES (extra segment); later another two segment registers were added –FS and GS.

Paged virtual memory

In paging the memory address space is divided into equal -sized blocks called pages. Using virtual memory hardware, each page can reside in any location of the computer's physical memory, or be flagged as being protected. Virtual memory makes it possible to have a linear virtual memory address space and to use it to access blocks fragmented over physical memory address space. Most computer architectures which support paging also use pages as the basis for memory protection.

A page table maps virtual memory to physical memory. The page table is usually invisible to the process. Page tables make it easier to allocate additional memory, as each new page can be allocated from anywhere in physical memory.

It is impossible for an application to access a page that has not been explicitly allocated to it, because every memory address either points to a page allocated to that application, or generates an interrupt called a page fault Unallocated pages, and pages allocated to any other application, do not have any addresses from the application point of view.

A page fault may not necessarily indicate an error. Page faults are not only used for memory protection. The operating system may manage the page table in such a way that a reference to a page that has been previously swapped out to disk causes a page fault. The operating system intercepts the page fault and, loads the required memory page, and the application continues as if no fault had occurred. This scheme, known as virtual memory, allows in -memory data not currently in use to be moved to disk storage and back in a way which is transparent to applications, to increase overall memory capacity.

Software fault handler can, if desired, check the missing key against a larger list of keys maintained by software; thus, the protection key registers inside the processor may be treated as a software-managed cache of a larger list of keys associated with a process.

Simulated segmentation

Simulation is use of a monitoring program to interpret the machine code instruction s of some computer architectures. Such an Instruction Set Simulator can provide memory protection

by using a segmentation - like scheme and validating the target address and length of each instruction in real time before actually executing them. The simulator must calculate the target address and length and compare this against a list of valid address ranges that it holds concerning the thread's environment , such as any dynamic memory blocks acquired since the thread's inception, plus any valid shared static memory slots. The meaning of "valid" may change throughout the thread's life depending upon context. It may sometimes be allowed to alter a static block of storage, and sometimes not, depending upon the current mode of execution, which may or may not depend on a storage key or supervisor state .

It is generally not advisable to use this method of memory protection where adequate facilities exist on a CPU, as this takes valuable processing power from the computer. However, it is generally used for debugging and testing purposes to provide an extra fine level of granularity to otherwise generic storage violations and can indicate precisely which instruction is attempting to overwrite the particular section of storage which may have the same storage key as unprotected storage.

FILE PROTECTION MECHANISM

Basic Forms of Protection

All multi user operating systems must provide some minimal protection to keep one user from maliciously or inadvertently accessing or modifying the files of another. As the number of users has grown, so also has the complexity of these protection schemes.

All "None Protection

OS operating systems, files were by default public. Any user could read, modify, or delete a file belonging to any other user. Instead of soft ware -or hardware- based protection, the principal protection involved trust combined with ignorance. System designers supposed that users could be trusted not to read or modify others' files, because the users would expect the same respect from others. Ignorance helped this situation, because a user could access a file only by name ; presumably users knew the names only of those files to which they had legitimate access.

However, it was acknowledged that certain system files were sensitive and that the system administrator could protect them with a password. A normal user could exercise this feature, but passwords were viewed as most valuable for protecting operating system files. Two philosophies guided password use. Sometimes, passwords were used to control all accesses (read, write, or delete), giving the system administrator complete control over all files. But at, other times passwords would control only write and delete accesses, because only these two actions affected other users. In either case, the password mechanism required a system operator's intervention each time access to the file began .However, this all-or-none protection is unacceptable for several reasons.

- **Lack of trust** - The assumption of trustworthy users is not necessarily justified. For systems with few users who all know each other, mutual respect might suffice; but in large systems where not every user knows every other user, there is no basis for trust.
- **All or nothing** - Even if a user identifies a set of trustworthy users, there is no convenient way to allow access only to them.
- **Rise of timesharing** -This protection scheme is more appropriate for a batch environment, in which users have little chance to interact with other users and in which users do their thinking and exploring when not interacting with the system. However, on timesharing systems, users interact with other users. Because users choose when to execute programs, they are more likely in a timesharing environment to arrange computing tasks to be able to pass results from one program or one user to another.
- **Complexity** - Because (human) operator intervention is required for this file protection, operating system performance is degraded. For this reason, this type of file protection is discouraged by computing centers for all but the most sensitive data sets.
- **File listings** - For accounting purposes and to help users remember for what files they are

- responsible, various system utilities can produce a list of all files. Thus, users are not necessarily ignorant of what files reside on the system. Interactive users may try to browse through any unprotected files

Group Protection

Because the all -or-nothing approach has so many drawbacks they thought that improved way to protect files. They focused on identifying groups of users who had some common relationship. In a typical implementation, the world is divided into three classes: the user, a trusted working group associated with the user, and the rest of the users. For simplicity we can call these classes' user, group, and world. This form of protection is used on some network systems and the Unix system.

All authorized users are separated into groups. A group may consist of several members working on a common project, a department, a class, or a single user. The basis for group membership is needed to share. The group members have some common interest and therefore are assumed to have files to share with the other group members. In this approach, no user belongs to more than one group. (Otherwise, a member belonging to groups A and B could pass along an A file to another B group member.) When creating a file, a user defines access rights to the file for the user, for other members of the same group, and for all other users in general. Typically, the choices for access rights are a limited set, such as {read, write, execute, delete}. For a particular file, a user might declare read - only access to the general world, read and write access to the group, and all rights to the user. This approach would be suitable for a paper being developed by a group, whereby the different members of the group might modify sections being written within the group. The paper itself should be available for people outside the group to review but not change.

A key advantage of the group protection approach is its ease of implementation. A user is recognized by two identifiers (usually numbers): a user ID and a group ID. These identifiers are stored in the file directory entry for each file and are obtained by the operating system when a user logs in. Therefore, the operating system can easily check

whether a proposed access to a file is requested from someone whose group ID matches the group ID for the file to be accessed.

Although this protection scheme overcomes some of the shortcomings of the all – or - nothing scheme, it introduces some new difficulties of its own.

Group affiliation. A single user cannot belong to two groups.

Multiple personalities -To overcome the one person one group restriction, certain people might obtain multiple accounts, permitting them, in effect, to be multiple users. This hole in the protection approach leads to new problems, because a single person can be only one user at a time.

All groups - .To avoid multiple personalities, the system administrator may decide

Limited sharing - . Files can be shared only within groups or with the world. Users want to be able to identify sharing partners for a file on a per file basis, for example, sharing one file with ten people and another file with twenty others.

Password or Other Token

We can apply a simplified form of password protection to file protection by allowing a user to assign a password to a file. User accesses are limited to those who can supply the correct password at the time the file is opened. The password can be required for any access or only for modifications (write access).

Password access creates for a user the effect of having a different "group" for every file. However, file passwords suffer from difficulties similar to those of authentication passwords:

- **Loss** - Depending on how the passwords are implemented, it is possible that no one will be able to replace a lost or forgotten password. The operators or system administrators can certain intervene and unprotect or assign a particular password, but often they cannot determine what password a user has assigned; if the user loses the password, a new one must be assigned.

- **Use** - Supplying a password for each access to a file can be inconvenient and time consuming.
- **Disclosure** -.If a password is disclosed to an unauthorized individual, the file becomes immediately accessible. If the user then changes the password to reprotect the file, all the other legitimate users must be informed of the new password because their old password will fail.
- **Revocation** -.To revoke one user's access right to a file, someone must change the password, thereby causing the same problems as disclosure.

USER AUTHENTICATION

An operating system bases much of its protection on knowing who a user of the system is. In real – life situations, people commonly ask for identification from people they do not know: A bank employee may ask for a driver's license before cashing a check, library employees may require some identification before charging out books, and immigration officials ask for passports as proof of identity. In -person identification is usually easier than remote identification. For instance, some universities do not report grades over the telephone because the office workers do not necessarily know the students calling. However, a professor who recognizes the voice of a certain student can release that student's grades. Over time, organizations and systems have developed means of authentication, using documents, voice recognition, fingerprint and retina matching, and other trusted means of identification. In computing, the choices are more limited and the possibilities less secure. Anyone can attempt to log in to a computing system. Unlike the professor who recognizes a student's voice, the computer cannot recognize electrical signals from on e person as being any different from those of anyone else. Thus, most computing authentication systems must be based on some knowledge shared only by the computing system and the user. Authentication mechanisms use any of three qualities to confirm a user's identity.

- Something the user **knows** a Passwords, PIN numbers, passphrases, a secret handshake, and mother's maiden name are examples of what a user may know.

- Something the user **has** a Identity badges, physical keys, a driver's license, or a uniform are common examples of things people have that make them recognizable.
- Something the user **is** a These authenticators, called biometrics, are based on a physical characteristic of the user, such as a fingerprint, the pattern of a person's voice, or a face (picture). These authentication method s are old (we recognize friends in person by their faces or on a telephone by their voices) but are just starting to be used in computer authentication

Passwords as Authenticators

The most common authentication mechanism for user to operating system is a password, a "word" known to computer and user. Although password protection seems to offer a relatively secure system, human practice sometimes degrades its quality.

Use of Passwords

Passwords are mutually agreed - upon code words, assumed to be known only to the user and the system. In some cases a user chooses passwords; in other cases the system assigns them. The length and format of the password also vary from one system to another. Even though they are widely used, passwords suffer from some difficulties of use:

- **Loss** - Depending on how the passwords are implemented, it is possible that no one will be able to replace a lost or forgotten password. The operators or system administrators can certainly intervene and unprotect or assign a particular password, but often they cannot determine what password a user has chosen; if the user loses the password, a new one must be assigned.
- **Use** - Supplying a password for each access to a file can be inconvenient and time consuming.
- **Disclosure** - If a password is disclosed to an unauthorized individual, the file becomes immediately accessible. If the user then changes the password to reprotect the file,

all other legitimate users must be informed of the new password because their old password will fail.

- **Revocation**. - To revoke one user's access right to a file, someone must change the password, thereby causing the same problems as disclosure. The use of passwords is fairly straightforward. A user enters some piece of identification, such as a name or an assigned user ID; this identification can be available to the public or easy to guess because it does not provide the real security of the system. The system then requests a password from the user. If the password matches that on file for the user, the user is authenticated and allowed access to the system. If the password match fails, the system requests the password again, in case the user mistyped

POSSIBLE QUESTIONS

PART A

Q.NO 1 TO 20 (MULTIPLE CHOICE QUESTIONS)

PART B (2 MARKS)

1. How will you secure programs?
2. What are the categories of flaws?
3. What is buffer overflows
4. What are malicious code
5. Define User Authentication.
6. How will you protect the files
7. How password face the difficulties
8. Define segmentation
9. What are Trap doors
10. How will you control against program threats.

PART C (6 MARKS)

1. Explain the various access control devices
2. List and explicate the major protocols used for secure communications.
3. Discuss the various types of threats to information Consider the following:
4. Enumerate file protection concept
5. Differentiate between non-malicious and malicious code.
6. Write brief note on each of the following

   i) Trapdoors ii) Salami Attack

7. State out the aspect involved in Paging.
8. Discuss User Authentication concept.

KARPAGAM ACADEMY OF HIGHER EDUCTATION
DEPARTMENT OF COMPUTER SCIENCE
III B.Sc CS
INFORMATION SECURITY[16CSU501A]

UNIT 3

S.NO	Question	Option 1	Option 2	Option 3	Option 4	Answer
1	In computer security, …….. means that the information in a computer system only be accessible	Confidentiality	integrity	availabiltiy	Authenticity	Confidentiality
2	The type of threats on the security of a computer system or network are ……..	Interception , modification	Creation	Interruption, Interception,	Interruption, Interception,	Worm
3	Which of the following is independent malicious program that need not any host program?	Trap doors	Trojan horse	Virus	Worm	Worm
4	The ……. is code that recognizes some special sequence of input or is triggered by being run from a	Trap doors	Trojan horse	Logic Bomb	Virus	Trap doors
5	The …….. is code embedded in some legitimate program that is set to “explode” when certain conditions	Trap doors	Trojan horse	Logic Bomb	Virus	Logic Bomb
6	Which of the following malicious program do not replicate automatically?	Trojan horse	Virus	Worm	Zombie	Trojan horse
7	…… programs can be used to accomplish functions indirectly that an unauthorized user could not	Trojan horse	Zombie	Worm	Logic Bomb	Trojan horse
8	A …….. is a program that can infect other programs by modifying them, the modification includes a	Worm	Virus	Trap doors	Zombie	Virus
9	The first computer virus is --------	Sasser	Creeper	Blaster	Virus	Creeper
10	To protect a computer from virus, you should install ------	antivirus	disk defragmenter	disk cleanup	backup wizard	antivirus
11	Which of the following is known as Malicious software?	maliciousware	illegalware	badware	malware	malware
12	MCAfee is an example of	Virus	Quick Heal	Antivirus	Photo Editing Software	Antivirus
13	When a logic bomb is activated by a time related event, it is known as -------	trojan horse	time bomb	virus	time related bomb	time bomb
14	The altering of data so that it is not usable unless the changes are undone is	biometrics	encryption	ergonomics	compression	encryption
15	Types of Flaws is divided into _____ categories	6	4	3	2	6
16	SHA-l has a message digest of	160 bits	512 bits	628 bits	820 bits	160 bits
17	Message authentication is a service beyond	Message Confidentiality	Message Integrity	Message Splashing	Message Sending	Message Integrity

18	In Message Confidentiality, the transmitted message must make sense to only intended	Receiver	Sender	Modulor	Translator	Receiver
19	_____ is the computing equivalent of trying to pour 2 litres of water into a one litre .	overflow	buffer overflow	flaws	malicious	buffer overflow
20	Asymmetric encryption transforms _____ into ciphertext using a one of two keys and an encryption algorithms.	cipher text	plaintext	covert	images	plaintext
21	Alice and Bob use two public key numbers, namely	P,G	P,P	G,G	R,P	P,G
22	_____ is the readable message or data that is fed into the algorithms input	plaintext	cipher text	images	channel	plaintext
23	the _____ algorithm performs various transformation on the plaintext.	images	decryption	encryption	channel	encryption
24	_____ and _____ keys that have been selected so that if one is used for encryption , the other is used for	public&protective	protected&private	encryption and decryption	public&private	public&private
25	_____ is the scrambled message produced as output	images	plaintext	ciphertext	channel	ciphertext
26	_____algorithm accepts the ciphertext and the matching key and produces the original plaintext	decryption	encryption	plaintext	images	decryption
27	the sender_____ a message with the recipients public key	encrypt	decrypt	plaintext	images	encrypt
28	the sender signs a message with its private key is known as	covert writing	analog signature	digital signature	type writer	digital signature
29	Program security flaws can derive from any kind of	hardware	software fault	twoway	three way	software fault
30	In flaws , the failure on first or last case is known as	boundary condition violation	domain error	validation error	serialization and aliasing	boundary condition violation
31	In flaws , permission checks is stated as _____	boundary condition violation	domain error	validation error	serialization and aliasing	validation error
32	In flaws , controlled access to data	boundary condition violation	domain error	validation error	serialization and aliasing	domain error
33	Programmers and other developers make many mistakes, most of which are _____ and _____	unintentional, nonmalicious	unintentional, malicious	intentional, nonmalicious	mistakes,error	unintentional, nonmalicious
34	A buffer resides in _____	memory	cpu	secondary memory	ram	memory
35	In Program Security writing a message on a computer screen, stopping a running program, generating a	malicious code	code	memory	special number fiels store	malicious code
36	Malicious code runs under the	user's authority	authentication	program	validation	user's authority
37	The DSS signature uses which hash algorithm?	MD5	SHA-2	SHA-1	Does not use hash algorithm	SHA-1
38	The client_key_exchange message uses a pre master key of size	48 bytes	56 bytes	64 bytes	32 bytes	48 bytes
39	a simple public key algorithm is _____ key exchange	brute force	hellman	diffie-hellman	none	diffie-hellman

40	Two diiferent users in Diffie Helman Algorithm exchange their keys	public key	private key	values	none	**public key**
41	the _____ algorithm depends for its effectiveness on the difficulty of computing discrete logarithmss	hellman	diffie-hellman	brust force	none	diffie-hellman
42	diffie-hellman key allow _____ users to exchage the key	2	3	4	5	2
43	security of transmission is critical for many network and	internet applications	middle compatibility	backward compatibility	front compatibility	internet applications
44	The _____ attack can endanger the security of the Diffie-Hellman method if two parties are not	man-in-the-middle	ciphertext attack	plaintext attack	none of the above	man-in-the-middle
45	The _____ method provides a one-time session key for two parties.	Diffie-Hellman	RSA	DES	AES	Diffie-Hellman
46	In diffie-hellman users pick up the _____ values for a and b	private	public	secret	none	private
47	A virus can be either transient or _____	resident	public	secret	worm	resident
48	The process of verifying an identity for a system entity	AES	authentication	reliability	security	authentication
49	_____ virus has a life that depends on the life of its ho	transient	resident	worm	trojan horse	transient
50	A _____ virus locates itself in memory	resident	transient	worm	trojan horse	resident
51	Message digest uses_____function	Heap	Hash	Math	logn2	Hash
52	A _____ is malicious code that, in addition to its primary effect, has a second, nonobvious malicious	Trojan horse	transient	worm	resident	Trojan horse
53	A _____is a class of malicious code that "detonates" or goes off when a specified condition occurs.	Trojan horse	logic bomb	worm	resident	logic bomb
54	Trap door otherwise known as _____	back door	logic bomb	worm	resident	back door
55	A ___ is a program that spreads copies of itself through a network	worm	back door	logic bomb	resident	worm
56	The primary difference between a ____ and a _____ is that a worm operates through networks, and a virus can	worm, virus	back door, Trap door	logic bomb , resident	resident, Trojan horse	worm, virus
57	A _____ attack merges bits of seemingly inconsequential data to yield powerful results.	salami	logic bomb	worm	resident	salami
58	The general name for these extraordinary paths of communication is _____	covert channels.	logic bomb	worm	resident	covert channels.
59	_____ is another characteristic of modular software	Information hiding	security	malicious	nonmalicious	Information hiding
60	Files can be shared only within groups is said to be as __	Limited sharing	All groups	Password	disclosure	Limited sharing

UNIT – IV

Database Security: Requirements, Reliability, Integrity, Sensitive data, Inference, Multilevel Security. Security in Networks : Threats in Networks, Security Controls, firewalls, Intrusion detection systems, Secure e-mails

Concept of a Database

A database is a collection of data and a set of rules that organize the data by specifying certain relationships among the data.

- The user describes a logical format for the data.
- The precise physical format of the file is of no concern to the user

A database administrator is a person who defines the rules that organize the data and also controls who should have access to what parts of the data.

The user interacts with the database through a program called a database manager or a database management system (DBMS), informally known as a front end.

Components of Databases

- – contain one related group of data
- Each record contains fields or elements
- The logical structure of a database is called a schema
- A particular user may have access to only part of the database, called a subschema

Adams	212 Market St.	Columbus	OH	43210
Benchly	501 Union St.	Chicago	IL	60603
Carter	411 Elm St.	Columbus	OH	43210

Queries

- Users interact with database managers through commands to the DBMS that retrieve, modify, add, or delete fields and records of the database.
- A command is called a query.

  For example,

  SELECT NAME = 'ADAMS'

- Other, more complex, selection criteria are possible, with logical operators

An example of a select query is

SELECT (ZIP='43210') ^ (NAME='ADAMS')

Advantage of Using Databases

- A database is a single collection of data, stored and maintained at one central location, to which many people may have access as needed
- The users are unaware of the physical arrangements; the unified logical arrangement is all they see
- Shared access – users use one common, centralized set of data
- Minimal redundancy – users do not have to collect and maintain their own sets of data
- Data consistency – change to a data value affects all users of the data value
- Data integrity – data values are protected against accidental or malicious undesirable changes
- Controlled access – only authorized users are allowed to view or to modify data values

Security Requirements

A list of requirements for database security.

Physical database integrity

The data of a database are immune to physical problems, such as power failures and someone can reconstruct the database if it is destroyed through failures, and someone can reconstruct the database if it is destroyed through a catastrophe.

Logical database integrity

The structure of the database is preserved. With logical integrity of a database, a modification to the value of one field does not affect other fields.

Element integrity

The data contained in each element are accurate.

Auditability

Auditability- It is possible to track who or what has accessed (or modified) the elements in the database.

Access control

A user is allowed to access only authorized data, and different users can be restricted to different modes of access (such as read or write).

User Authentication

Every user is positively identified, both for the audit trail and for permission to access certain data.

Availability

Users can access the database in general and all the data for which they are authorized.

Integrity of the Database

Two situations can affect the integrity of a database:

- when the whole database is damaged
- when individual data items are unreadable

Integrity of the database as a whole is the responsibility of

- The DBMS
- The operating system
- The (human) computing system manager.

Sometimes it is important to be able to reconstruct the database at the point of a failure.

- The DBMS must maintain a log of transactions.
- The system can obtain accurate account balances by reverting to a backup Copy of the database and reprocessing all later transactions from the log

Element Integrity

The integrity of database elements is their correctness or accuracy.

This corrective action can be taken in three ways.

- Field checks -activities that test for appropriate values in a position.
- Access control
- A change log - A change log A change log lists every change made to the database; it contains both original and modified values. Using this log, a database administrator can undo any changes that were made in error.

Auditability

- For some applications it may be desirable to generate an audit record of all access (read or write) to a database.
- Such a record can help to maintain the database's integrity, or atleast to discover after the fact who had affected which values and when.

Access Control

Databases are often separated logically by user access privileges.

User Authentication

- The DBMS can require rigorous user authentication.
- A DBMS might insist that a user pass both specific password and time-of-day checks.
- This authentication supplements the authentication performed by the operating system.

Availability

Integrity/Confidentiality/Availability – Computer Security

- Integrity is a major concern in the design of database management systems.
- Confidentiality is a key issue with databases because of the inference problem
- Availability is important because of the shared access motivation underlying database development.

Reliability and Integrity

Databases amalgamate data from many sources, and users expect a DBMS to provide access to the data in a reliable way.

Reliability - mean that the software runs for very long periods of time without failing. Database concerns about reliability and integrity can be viewed from three dimensions:

- **Database integrity**: concern that the database as a whole is protected against damage
- **Element integrity**: concern that the value of a specific data element is written or changed only by authorized users.
- **Element accuracy**: concern that only correct values are written into the elements of a database.

Protection Features from the Operating System

- A responsible system administrator backs up the files of a database periodically along with other user files.
- The files are protected during normal execution against outside access by the operating system's standard access control facilities.
- Finally, the operating system performs certain integrity checks for all data as a part of normal read and writes operations for I/O devices.

Two-Phase Update

A serious problem for a database manager is the failure of the computing system in the middle of modifying data.

If the data item to be modified was a long field, half of the field might show the new value, while the other half would contain the old.

Update Technique

1. **The intent phase** - the DBMS gathers the resources it needs to perform the update
2. **committing,** involves the writing of a commit flag to the database. The commit flag means that the DBMS has passed the point of no return: After committing, the DBMS begins making permanent changes.

Redundancy/Internal Consistency

- Error Detection and Correction Codes
- Shadow Fields - Entire attributes or entire records can be duplicated in a database. If the data are irreproducible, this second copy can provide an immediate replacement if an error is detected.

Recovery

In addition to these error correction processes, a DBMS can maintain a log of user accesses, particularly changes. In the event of a failure, the database is reloaded from a backup copy and all later changes are then applied from the audit log.

Concurrency/Consistency

- Database systems are often multiuser systems.

- If both users try to modify the same data items, we often assume that there is no conflict because each knows what to write; the value to be written does not depend on the previous value of the data item. However, this supposition is not quite accurate.

Monitors

The monitor is the unit of a DBMS responsible for the structural integrity of the database.

Forms of monitors

1. Range Comparisons - A range comparison monitor tests each new value to ensure that the value is within an acceptable range

- Filters or patterns are more general types of data form checks.

2. State constraints describe the condition of the entire database.

3. Transition constraints describe conditions necessary before changes can be applied to a database.

Sensitive Data

- Sensitive data are data that should not be made public.
- There exist cases that some but not all of the elements in the database are sensitive.
- There may be varying degrees of sensitivity.

Several factors can make data sensitive.

Inherently sensitive - The value itself may be so revealing that it is sensitive. Examples are the locations of defensive missiles.

From a sensitive source - The source of the data may indicate a need for confidentiality. An example is information from an informer whose identity would be compromised if the information were disclosed.

Declared sensitive - The database administrator or the owner of the data may have declared the data to be sensitive.

Part of a sensitive attribute or a sensitive record - In a database, an entire attribute or record may be classified as sensitive.

Sensitive in relation to previously disclosed information. - ome data become sensitive in the presence of other data.

Access Decisions

The DBMS may consider several factors when deciding whether to permit an access.

Availability of the data - One or more required elements may be inaccessible.

For example, if a user is updating several fields, other users' accesses to those fields must be blocked temporarily. This blocking ensures that users do not receive inaccurate information

Acceptability of the access - One or more values of the record may be sensitive and not accessible by the general user. A DBMS should not release sensitive data to unauthorized

individuals.

Authenticity of the user - certain characteristics of the user external to the database may also be considered when permitting access.

For example, to enhance security, the database administrator may permit someone to access the database only at certain times, such as during working hours\

Access Decisions - Types of Disclosures

- Exact Data - The most serious disclosure is the exact value of a sensitive data item itself
- Bounds - Another exposure is disclosing bounds on a sensitive value; that is, indicating that a sensitive value, y, is between two values, L and H.
- Negative Result - Sometimes we can word a query to determine a negative result. That is, we can learn that z is not the value of y.
- Existence - The existence of data is itself a sensitive piece of data.
- Probable Value - it may be possible to determine the probability that a certain element has a certain value

Inference

- **Inference is a way to infer or derive sensitive data from non-sensitive data.**

Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45.	1	Holmes
Bailey	M	B	0	0.	0	Grey
Chin	F	A	3000	20.	0	West
Dewitt	M	B	1000	35.	3	Grey
Earhart	F	C	2000	95.	1	Holmes
Fein	F	C	1000	15.	0	West
Groff	M	C	4000	0.	3	West
Hill	F	B	5000	10.	2	Holmes
Koch	F	C	0	0.	1	West
Liu	F	A	0	10.	2	Grey
Majors	M	C	2000	0.	2	Grey

1. Direct Attack

A user tries to determine values of sensitive fields by seeking them directly with queries that yield few records.

2. Indirect Attack

- Sum - An attack by sum tries to infer a value from a reported sum.

- Count - The count can be combined with the sum to produce some even more revealing results.
- Mean - The arithmetic mean (average) allows exact disclosure if the attacker can manipulate the subject population.

3. Tracker Attacks

A tracker attack can fool the database manager into locating the desired data by using additional queries that produce small results.

Controls for Statistical Inference Attacks

- **Suppression** - sensitive data values are not provided; the query is rejected without response.
- **Concealing** - the answer provided is close to but not exactly the actual value.

Random Sample

- With random sample control, a result is not derived from the whole database; instead the result is computed on a random sample of the database.
- The sample chosen is large enough to be valid.

Random Data Perturbation

- It is sometimes useful to perturb the values of the database by a small error.
- Generate a small random error term εi and add it to xi for statistical results.

Query Analysis

- A more complex form of security uses query analysis.

Multilevel security

Multilevel security or multiple levels of security (MLS) is the application of a computer system to process information with incompatible classifications (i.e., at different security levels), permit access by users with different security clearances and needs-to-know, and prevent users from obtaining access to information for which they lack authorization. There are two contexts for the use of multilevel security. One is to refer to a system that is adequate to protect itself from subversion and has robust mechanisms to separate information domains, that is, trustworthy. Another context is to refer to an application of a computer that will require the computer to be strong enough to protect itself from subversion and possess adequate mechanisms to separate information domains, that is, a system we

must trust. This distinction is important because systems that need to be trusted are not necessarily trustworthy.

Lattice Model of Access Security

The military security model is representative of a more general scheme, called a lattice. The dominance relation defined in the military model is the relation for the lattice. The relationis transitive and antisymmetric. The largest element of the lattice is the classification <topsecret; all compartments>, and the smallest element is <unclassified; no compartments>;these two elements respectively dominate and are dominated by all elements. Therefore, themilitary model is a lattice.

BellLaPadula Confidentiality Model

The Bell and La Padula model [BEL73] is a formal description of the allowable paths of information flow in a secure system. The model's goal is to identify allowable communication when maintaining secrecy is important. The model has been used to define security requirements for systems concurrently handling data at different sensitivity levels.

Two properties characterize the secure flow of information.

1. Simple Security Property. A subject s may have read access to an object o only if C(o)=< C (s).

In the military model, this property says that the security class (clearance) of someone receiving a piece of information must be at least as high as the class (classification) of the information.

2. Property (called the "star property"). A subject s who has read access to an object o may have write access to an object p only if C(o) =<C(p).

BellLaPadula model

The BellLaPadula model applies only to secrecy of information: The model identifies paths that could lead to inappropriate disclosure of information. However, the integrity of data is important, too. Biba constructed a model for preventing inappropriate modification of data.

The properties are

1. Simple Integrity Property. Subject s can modify (have write access to) object o only if I(s) >=I(o)

2. Integrity Property. If subject s has read access to object o with integrity level I(o), s can havewrite access to object p only if I(o) >=I(p).

Security in Networks - Threats in Networks

Main aims of threats are to compromise confidentiality, integrity applied against data,

software, hardware by nature accidents, non-malicious humans and malicious attackers.

What Makes A Network Vulnerable?

1. Anonymity
2. Many Points Of Attack
3. Sharing
4. Complexity Of System

Threat Precursors:

1. Port scan
2. Social Engineering
3. Reconnaissance
4. Operating System and Application fingerprinting
5. Bulletin Boards and chats
6. Availability of Documentation

Threats In Transit: Eavesdropping and Wiretapping

The term eavesdrop implies overhearing without expanding any extra effort. For example we can say that an attacker is eavesdropping by monitoring all traffic passing through a node. The more hostile term is wiretap, which means intercepting communication through some effort.

Choices of wiretapping are:

1. Cable
2. Microwave
3. Satellite Communication
4. Optical Fiber
5. Wireless

From, a security stand point we should assume all communication links between network nodes that can broken. For this reason commercial network users employ encryption to protect the confidentiality of their communication.

Protocol Flaws:

Each protocol is identified by its Request For Comment (RFC) number. In TCP, the sequence number of the client increments regularly which can be easily guessed and also which will be the next number.

Impersonation:

In many instances, there is an easier way than wiretapping for obtaining information on a network: impersonate another person or process.

In impersonation, an attacker has several choices:

- Guess the identity and authentication details of the target
- Disable authentication mechanism at the target computer
- Use a target that will not be authenticated
- Use a target whose authentication data are known

Spoofing:

Obtaining the network authentication credentials of an entity(a user, an account, a process, a node, a device) permits an attacker to create a full communication under the entity's identity.

Examples of spoofing are masquerading, session hijacking, and man-in-the-middle attacks.

- In a masquerade one host pretends to be another.
- Session hijacking is intercepting and carrying on a session begun by another entity.
- Man-in-the-middle attack is a similar form of attack, in which one entity intrudes between two others.

Message Confidentiality Threats:

An attacker can easily violate message confidentiality (and perhaps integrity) because of the public nature of networks. Eavesdropping and impersonation attacks can lead to a confidentiality or integrity failure. Here we consider several other vulnerabilities that can affect confidentiality.

1. Misdelivery
2. Exposure
3. Traffic Flow Analysis

Message Integrity Threats:

In many cases, the integrity or correctness of a communication is at least as important as its confidentiality. In fact for some situations, such as passing authentication data, the integrity of the communication is paramount. Threats based upon failures of integrity in communication

- Falsification of messages
- Noise

Web Site Defacement:

One of the most widely known attacks is the web site defacement attack. Because of the large

number of sites that have been defaced and the visibility of the result, the attacks are often reported in the popular press. A defacement is common not only because of its visibility but also because of the ease with which one can be done.

The website vulnerabilities enable attacks known as buffer overflows, dot-dot problems, application code errors, and server side include problems.

Denial of Service:

Availability attacks, sometimes called denial-of-service or DOS attacks, are much more significant in networks than in other contexts. There are many accidental and malicious threats to availability or continued service. There are many accidental and malicious threats to availability or continued service.

1) Transmission Failure
2) Connection Flooding
3) Echo-Chargen
4) Ping of Death
5) Smurf
6) Syn Flood
7) Teardrop
8) Traffic Redirection
9) DNS Attacks.

Threats in Active or Mobile Code:

Active code or mobile code is a general name for code that is pushed to the client for execution. Why should the web server waste its precious cycles and bandwidth doing simple work that the client's workstation can do? For example, suppose you want your web site to have bears dancing across the top of the page. To download the dancing bears, you could download a new image for each movement the bears take: one bit forward, two bits forward, and so forth. However, this approach uses far too much server time and bandwidth to compute the positions and download new images. A more efficient use of (server) resources is to download a program that runs on the client's machine and implements the movement of the bears.

Network Security Controls

The list of security attacks is long, and the news media carry frequent accounts of serious security incidents.

Security Threat Analysis:

The three steps of a security threat analysis in other situations are described here. First, we scrutinize all the parts of a system so that we know what each part does and how it interacts with other parts. Next, we consider possible damage to confidentiality, integrity, and availability. Finally, we hypothesize the kinds of attacks that could cause this damage. We can take the same steps with a network. We begin by looking at the individual parts of a network:

All the threats are summarized with a list as

- Intercepting data in traffic
- Accessing programs or data at remote hosts
- Modifying programs or data at remote hosts
- Modifying data in transit
- Inserting communications
- Impersonating a user
- Inserting a repeat of a previous communication
- Blocking selected traffic
- Blocking all traffic
- Running a program at a remote host

Design and Implementation:

Architecture:

As with so many of the areas we have studied, planning can be the strongest control. In particular, when we build or modify computer-based systems, we can give some thought to their overall architecture and plan to "build in" security as one of the key constructs. Similarly, the architecture or design of a network can have a significant effect on its security.

The main areas to cover are

- Segmentation
- Redundancy
- Single point of failure
- Mobile agents

Encryption:

Encryption is powerful for providing privacy, authenticity, integrity, and limited access to data.

Because networks often involve even greater risks, they often secure data with encryption, perhaps in combination with other controls. There are 2 types of encryption scheme exists:

- Link encryption (data are encrypted just before the system places them on the physical communications link)
- End-to-end encryption (provides security from one end of a transmission to the other

Content Integrity:

Content integrity comes as a bonus with cryptography. No one can change encrypted data in a meaningful way without breaking the encryption. This does not say, however, that encrypted data cannot be modified. Changing even one bit of an encrypted data stream affects the result after decryption, often in a way that seriously alters the resulting plaintext. We need to consider three potential threats:

- Malicious modification that changes content in a meaningful way
- Malicious or non-malicious modification that changes content in a way that is not necessarily meaningful
- non-malicious modification that changes content in a way that will not be detected

Encryption addresses the first of these threats very effectively. To address the others, we can use other controls.

Strong Authentication:

In the network case, however, authentication may be more difficult to achieve securely because of the possibility of eavesdropping and wiretapping, which are less common in non networked environments. Also, both ends of a communication may need to be authenticated to each other.

Here the main issues are

- One time password
- Challenge response systems
- Digital distributed authentication

Access Controls:

Authentication deals with the who of security policy enforcement; access controls enforce the what and how.

ACLs on Routers

Routers perform the major task of directing network traffic either to sub-networks they control or to other routers for subsequent delivery to other sub-networks. Routers convert external IP addresses into internal MAC addresses of hosts on a local sub-network. Suppose a host is being spammed (flooded) with packets from a malicious rogue host. Routers can be configured with access control lists to deny access to particular hosts from particular hosts. So, a router could delete all packets with a source address of the rogue host and a destination address of the target host.

Firewalls

A firewall is a device that filters all traffic between a protected or "inside" network and a less trustworthy or "outside" network. Usually a firewall runs on a dedicated device; because it is a single point through which traffic is channeled, performance is important, which means non-firewall functions should not be done on the same machine. Because a firewall is executable code, an attacker could compromise that code and execute from the firewall's device. Thus, the fewer pieces of code on the device, the fewer tools the attacker would have by compromising the firewall. Firewall code usually runs on a proprietary or carefully minimized operating system. The purpose of a firewall is to keep "bad" things outside a protected environment. To accomplish that, firewalls implement a security policy that is specifically designed to address what bad things might happen. For example, the policy might be to prevent any access from outside (while still allowing traffic to pass from the inside to the outside). Alternatively, the policy might permit accesses only from certain places, from certain users, or for certain activities. Part of the challenge of protecting a network with a firewall is determining which security policy meets the needs of the installation.

Design of Firewalls:

A reference monitor must be

- Always invoked
- Tamperproof
- Small and simple enough for rigorous analysis

A firewall is a special form of reference monitor. By carefully positioning a firewall within a network, we can ensure that all network accesses that we want to control must pass through it. This restriction meets the "always invoked" condition. A firewall is typically well isolated, making it highly immune to modification. Usually a firewall is implemented on a separate computer, with direct connections only to the outside and inside networks. This isolation is expected to meet the "tamperproof" requirement. And firewall designers strongly recommend keeping the functionality of the

firewall simple.

Types of Firewalls:

Firewalls have a wide range of capabilities. Types of firewalls include

- Packet filtering gateways or screening routers
- Stateful inspection firewalls
- Application proxie
- Guards
- Personal firewalls

Packet Filtering Gateway:

A packet filtering gateway or screening router is the simplest, and in some situations, the most effective type of firewall. A packet filtering gateway controls access to packets on the basis of packet address (source or destination) or specific transport protocol type (such as HTTP web traffic).

Stateful Inspection Firewall:

Filtering firewalls work on packets one at a time, accepting or rejecting each packet and moving on to the next. They have no concept of "state" or "context" from one packet to the next. A stateful inspection firewall maintains state information from one packet to another in the input stream.

One classic approach used by attackers is to break an attack into multiple packets by forcing some packets to have very short lengths so that a firewall cannot detect the signature of an attack split across two or more packets. (Remember that with the TCP protocols, packets can arrive in any order, and the protocol suite is responsible for reassembling the packet stream in proper order before passing it along to the application.) A stateful inspection firewall would track the sequence of packets and conditions from one packet to another to thwart such an attack.

Application Proxy

Packet filters look only at the headers of packets, not at the data inside the packets. Therefore, a packet filter would pass anything to port 25, assuming its screening rules allow inbound connections to that port. But applications are complex and sometimes contain errors. Worse, applications (such as the e-mail delivery agent) often act on behalf of all users, so they require privileges of all users (for example, to store incoming mail messages so that inside users can read them). A flawed application, running with all users' privileges, can cause much damage. An application proxy gateway, also called a bastion host, is a firewall that simulates the (proper) effects of an application so that the application

receives only requests to act properly. A proxy gateway is a two-headed device: It looks to the inside as if it is the outside (destination) connection, while to the outside it responds just as the insider would. An application proxy runs pseudo-applications.

For instance, when electronic mail is transferred to a location, a sending process at one site and a receiving process at the destination communicate by a protocol that establishes the legitimacy of a mail transfer and then actually transfers the mail message. The protocol between sender and destination is carefully defined. A proxy gateway essentially intrudes in the middle of this protocol exchange, seeming like a destination in communication with the sender that is outside the firewall, and seeming like the sender in communication with the real destination on the inside. The proxy in the middle has the opportunity to screen the mail transfer, ensuring that only acceptable e-mail protocol commands are sent to the destination.

Guard:

A guard is a sophisticated firewall. Like a proxy firewall, it receives protocol data units, interprets them, and passes through the same or different protocol data units that achieve either the same result or a modified result. The guard decides what services to perform on the user's behalf in accordance with its available knowledge, such as whatever it can reliably know of the (outside) user's identity, previous interactions, and so forth. The degree of control a guard can provide is limited only by what is computable. But guards and proxy firewalls are similar enough that the distinction between them is sometimes fuzzy. That is, we can add functionality to a proxy firewall until it starts to look a lot like a guard.

Personal Firewalls:

A personal firewall is an application program that runs on a workstation to block unwanted traffic, usually from the network. A personal firewall can complement the work of a conventional firewall by screening the kind of data a single host will accept, or it can compensate for the lack of a regular firewall, as in a private DSL or cable modem connection.

The personal firewall is configured to enforce some policy. For example, the user may decide that certain sites, such as computers on the company network, are highly trustworthy, but most other sites are not. The user defines a policy permitting download of code, unrestricted data sharing, and management access from the corporate segment, but not from other sites. Personal firewalls can also generate logs of accesses, which can be useful to examine in case something harmful does slip through the firewall.

A personal firewall runs on the very computer it is trying to protect. Thus, a clever attacker is likely to attempt an undetected attack that would disable or reconfigure the firewall for the future. Still, especially for cable modem, DSL, and other "always on" connections, the static workstation is a visible and vulnerable target for an ever-present attack community. A personal firewall can provide reasonable protection to clients that are not behind a network firewall.

Comparison of Firewall types:

Packet Filtering	**Sateful Inspection**	**Application Proxy**	**Guard**	**Personal firewall**
Simple	More complex	Even complex	Most complex	Similar to packet filtering
Sees only addresses and service protocol type	Can see either addresses or data	Sees full data portion of packet	Sees full text of communication	Can see full data portion of packet
Auditing difficult	Auditing possible	Can audit activity	Can audit activity	Can and usually does audit activity
Screens based on connection rules	Screens based on information across packetsin either header or data field	Screens based on behavior of proxies	Screens based on interpretation of message contents	Typically, screens based on information in a single packet, using header or data
Complex addressing rules can make configuration tricky	Usually preconfigured to detect certain attack signatures	Simple proxies can substitute for complex addressing rules	Complex guard functionality can limit assurance	Usually starts in "deny all inbound" mode, to which user adds trusted addresses as they appear

Intrusion Detection System:

An intrusion detection system (IDS) is a device, typically another separate computer, that monitors activity to identify malicious or suspicious events. An IDS is a sensor, like a smoke detector, that raises an alarm if specific things occur. A model of an IDS is shown in below figure. The

components in the figure are the four basic elements of an intrusion detection system, based on the Common Intrusion Detection Framework of [STA96]. An IDS receives raw inputs from sensors. It saves those inputs, analyzes them, and takes some controlling action.

Types of IDSs

The two general types of intrusion detection systems are signature based and heuristic. Signature-based intrusion detection systems perform simple pattern-matching and report situations that match a pattern corresponding to a known attack type. Heuristic intrusion detection systems, also known as anomaly based, build a model of acceptable behavior and flag exceptions to that model; for the future, the administrator can mark a flagged behavior as acceptable so that the heuristic IDS will now treat that previously unclassified behavior as acceptable.

Intrusion detection devices can be network based or host based. A network-based IDS is a stand-alone device attached to the network to monitor traffic throughout that network; a host-based IDS runs on a single workstation or client or host, to protect that one host.

Signature-Based Intrusion Detection:

A simple signature for a known attack type might describe a series of TCP SYN packets sent to many different ports in succession and at times close to one another, as would be the case for a port scan. An intrusion detection system would probably find nothing unusual in the first SYN, say, to port 80, and then another (from the same source address) to port 25. But as more and more ports receive SYN packets, especially ports that are not open, this pattern reflects a possible port scan. Similarly, some implementations of the protocol stack fail if they receive an ICMP packet with a data length of 65535 bytes, so such a packet would be a pattern for which to watch.

Heuristic Intrusion Detection:

Because signatures are limited to specific, known attack patterns, another form of intrusion detection becomes useful. Instead of looking for matches, heuristic intrusion detection looks for behavior that is out of the ordinary. The original work in this area focused on the individual, trying to find characteristics of that person that might be helpful in understanding normal and abnormal behavior. For example, one user might always start the day by reading e-mail, write many documents using a word processor, and occasionally back up files. These actions would be normal. This user does not seem to use many administrator utilities. If that person tried to access sensitive system management utilities, this new behavior might be a clue that someone else was acting under the user's identity.

Inference engines work in two ways. Some, called state-based intrusion detection systems,

see the system going through changes of overall state or configuration. They try to detect when the system has veered into unsafe modes. Others try to map current activity onto a model of unacceptable activity and raise an alarm when the activity resembles the model. These are called model-based intrusion detection systems. This approach has been extended to networks .Later work sought to build a dynamic model of behavior, to accommodate variation and evolution in a person's actions over time. The technique compares real activity with a known representation of normality.

Alternatively, intrusion detection can work from a model of known bad activity. For example, except for a few utilities (login, change password, create user), any other attempt to access a password file is suspect. This form of intrusion detection is known as misuse intrusion detection. In this work, the real activity is compared against a known suspicious area.

Stealth Mode:

An IDS is a network device (or, in the case of a host-based IDS, a program running on a network device). Any network device is potentially vulnerable to network attacks. How useful would an IDS be if it itself were deluged with a denial-of-service attack? If an attacker succeeded in logging in to a system within the protected network, wouldn't trying to disable the IDS be the next step?

To counter those problems, most IDSs run in stealth mode, whereby an IDS has two network interfaces: one for the network (or network segment) being monitored and the other to generate alerts and perhaps other administrative needs. The IDS uses the monitored interface as input only; it never sends packets out through that interface. Often, the interface is configured so that the device has no published address through the monitored interface; that is, a router cannot route anything to that address directly, because the router does not know such a device exists. It is the perfect passive wiretap. If the IDS needs to generate an alert, it uses only the alarm interface on a completely separate control network.

Goals for Intrusion Detection Systems:

1. **Responding to alarms:**

Whatever the type, an intrusion detection system raises an alarm when it finds a match. The alarm can range from something modest, such as writing a note in an audit log, to something significant, such as paging the system security administrator. Particular implementations allow the user to determine what action the system should take on what events.

In general, responses fall into three major categories (any or all of which can be used in a single response):

1. Monitor, collect data, perhaps increase amount of data collected
2. Protect, act to reduce exposure
3. Call a human

2. **False Results:**

Intrusion detection systems are not perfect, and mistakes are their biggest problem. Although an IDS might detect an intruder correctly most of the time, it may stumble in two different ways: by raising an alarm for something that is not really an attack (called a false positive, or type I error in the statistical community) or not raising an alarm for a real attack (a false negative, or type II error). Too many false positives means the administrator will be less confident of the IDS's warnings, perhaps leading to a real alarm's being ignored. But false negatives mean that real attacks are passing the IDS without action. We say that the degree of false positives and false negatives represents the sensitivity of the system. Most IDS implementations allow the administrator to tune the system's sensitivity, to strike an acceptable balance between false positives and negatives.

IDS strength and limitations:

On the upside, IDSs detect an ever-growing number of serious problems. And as we learn more about problems, we can add their signatures to the IDS model. Thus, over time, IDSs continue to improve. At the same time, they are becoming cheaper and easier to administer. On the downside, avoiding an IDS is a first priority for successful attackers. An IDS that is not well defended is useless. Fortunately, stealth mode IDSs are difficult even to find on an internal network, let alone to compromise. IDSs look for known weaknesses, whether through patterns of known attacks or models of normal behavior. Similar IDSs may have identical vulnerabilities, and their selection criteria may miss similar attacks. Knowing how to evade a particular model of IDS is an important piece of intelligence passed within the attacker community. Of course, once manufacturers become aware of a shortcoming in their products, they try to fix it. Fortunately, commercial IDSs are pretty good at identifying attacks. Another IDS limitation is its sensitivity, which is difficult to measure and adjust. IDSs will never be perfect, so finding the proper balance is critical.

In general, IDSs are excellent additions to a network's security. Firewalls block traffic to particular ports or addresses; they also constrain certain protocols to limit their impact. But by definition, firewalls have to allow some traffic to enter a protected area. Watching what that traffic actually does inside the protected area is an IDS's job, which it does quite well.

Secure Email:

We rely on e-mail's confidentiality and integrity for sensitive and important communications, even though ordinary e-mail has almost no confidentiality or integrity. we investigate how to add confidentiality and integrity protection to ordinary e-mail.

Security of email:

Sometimes we would like e-mail to be more secure. To define and implement a more secure form, we begin by examining the exposures of ordinary e-mail.

Threats to E-mail

1. Message interception (confidentiality)
2. Message interception (blocked delivery)
3. Message interception and subsequent replay
4. Message content modification
5. Message origin modification
6. Message content forgery by outsider
7. Message origin forgery by outsider
8. Message content forgery by recipient
9. Message origin forgery by recipient
10. Denial of message transmission

Requirements and solutions:

Following protections must be taken for protection in emails

1. Message confidentiality (the message is not exposed en route to the receiver)
2. Message integrity (what the receiver sees is what was sent)
3. Sender authenticity (the receiver is confident who the sender was)
4. Non repudiation (the sender cannot deny having sent the message)

Designs:

One of the design goals for encrypted e-mail was allowing security-enhanced messages to travel as ordinary messages through the existing Internet e-mail system. This requirement ensures that the large existing e-mail network would not require change to accommodate security. Thus, all protection occurs within the body of a message.

Confidentiality:

The encrypted e-mail standard works most easily as just described, using both symmetric and asymmetric encryption. The standard is also defined for symmetric encryption only: To use

symmetric encryption, the sender and receiver must have previously established a sharedsecret encryption key. The processing type ("Proc-Type") field tells what privacy enhancement services have been applied. In the data exchange key field ("DEK-Info"), the kind of key exchange (symmetric or asymmetric) is shown. The key exchange ("Key-Info") field contains the message encryption key, encrypted under this shared encryption key. The field also identifies the originator (sender) so that the receiver can determine which shared symmetric key was used. If the key exchange technique were to use asymmetric encryption, the key exchange field would contain the message encryption field, encrypted under the recipient's public key. Also included could be the sender's certificate (used for determining

authenticity and for generating replies). The encrypted e-mail standard supports multiple encryption algorithms, using popular algorithms such as DES, triple DES, and AES for message confidentiality, and RSA and Diffie-Hellman for key exchange.

Encryption of secure e-mail:

Encrypted e-mail provides strong end-to-end security for electronic mail. Triple DES, AES, and RSA cryptography are quite strong, especially if RSA is used with a long bit key (1024 bits or more). The vulnerabilities remaining with encrypted e-mail come from the points not covered: the endpoints. An attacker with access could subvert a sender's or receiver's machine, modifying the code that does the privacy enhancements or arranging to leak a cryptographic key.

Examples of Secure E-mail:

1. PGP (Pretty Good Privacy)
2. S/MIME (Secure Multipurpose Internet Mail Extensions)

POSSIBLE QUESTIONS

PART A

Q.NO 1 TO 20 (MULTIPLE CHOICE QUESTIONS)

PART B (2 MARKS)

1. Define firewall.
2. What are the measures that may use for intrusion detection system?
3. Mention the limitations of intrusion detection systems (IDS).
4. List out types of Firewalls?
5. Define Phishing.
6. List out the file protection in E-mail.
7. List out the threats in E-mail
8. List any 3 comparison of firewalls types.
9. What are the main issues in strong authentication?
10. What Makes A Network Vulnerable?

PART C (6 MARKS)

1. Elucidate the different types of Intrusion Detection System (IDS) with their advantages and disadvantages.
2. Define firewall. What are its different types? Explain the working of each in detail.
3. Explain the different types of requirements in database security.
4. List and explicate the major role on secure e-mails.
5. Enlighten Inference and Sensitive data concept in database security.
6. List and explicate the major role on threats in network.
7. Explain the issues in Multilevel Security.
8. List and explicate any few techniques in network security controls.
9. Illustrate Inference with neat sketch.
10. List and explain the goals of Intrusion Detection System.

KARPAGAM ACADEMY OF HIGHER EDUCTATION

DEPARTMENT OF COMPUTER SCIENCE

III B.Sc CS

INFORMATION SECURITY[16CSU501A]

UNIT 4

S.NO	Question	Option 1	Option 2	Option 3	Option 4	Answer
1	A _____ is a collection of data and a set of rules that organize the data by specifying certain relationships among the data.	data	database	information	dbms	database
2	In database security who is responsible to describes a logical format for the data.	user	administrator	server	database manager	user
3	The precise_____ of the file is of no concern to the user	physical format	logical format	database format	format	physical format
4	A _____is a person who defines the rules that organize the data	database administrator	administrator	dbms	user	database administrator
5	In database, _____ controls who should have access to what parts of the data.	physical format	database administrator	logical format	logical format & physical format	database administrator
6	The user interacts with the database through a program called a _____	database manager	database administrator	database system	manager	database manager
7	_____ contain one related group of data	Record	file	fields	schema	Record
8	Each_____ contains fields or elements	Record	file	fields	schema	Record
9	The logical structure of a database is called a _____	Record	schema	fields	file	schema
10	The _____ of a database is called a schema	Record	schema	fields	logical structure	logical structure
11	A particular user may have access to only part of the database, called a _____	subschema	schema	Record	database	subschema

12	Users interact with _____through commands to the DBMS that retrieve, modify, add, or delete fields and records of the database.	server	database managers	user	administrator	database managers
13	A command is called a _____	query	record	user	syntax	query
14	more complex, selection criteria are possible, with _____	logical operators	bit operators	binary operators	arithmetic operators	logical operators
15	A _____is a single collection of data, stored and maintained at one central location, to which many people may have access as needed	database	data	information	dbms	database
16	users use one common, centralized set of data is said to be _____	Shared access	Minimal redundancy	Data consistency	Data integrity	Shared access
17	users do not have to collect and maintain their own sets of data is known as ____	Shared access	Minimal redundancy	Data consistency	Data integrity	Minimal redundancy
18	change to a data value affects all users of the data value is _____	Shared access	Minimal redundancy	Data consistency	Data integrity	Data consistency
19	data values are protected against accidental or malicious undesirable changes IS _____	Shared access	Minimal redundancy	Data consistency	Data integrity	Data integrity
20	only authorized users are allowed to view or to modify data values is _____	Controlled access	Shared access	Minimal redundancy	Data consistency	Controlled access
21	Find the odd man out	element integrity	integrity of the database	user authentiction	query	query
22	The structure of the database is preserved in _____	physical database integrity	logical database integrity	element integrity	integrity of the database	logical database integrity
23	With _____ of a database, a modification to the value of one field does not affect other fields.	logical integrity	element integrity	integrity of the database	physical database integrity	logical integrity
24	The data contained in each element are accurate. In _____	element integrity	integrity of the database	user authentiction	query	element integrity
25	It is possible to track who or what has accessed (or modified) the elements in the database is said to be _____	auditability	user authentiction	access control	controlled access	auditability
26	A user is allowed to access only authorized data, and different users can be restricted to different modes of access (such as read or write).	auditability	user authentiction	access control	controlled access	access control
27	Every user is positively identified, both for the audit trail and for permission to access certain data is _____	element integrity	integrity of the database	user authentication	controlled access	user authentication

28	Users can access the database in general and all the data for whic	availability	user authentiction	access control	controlled access	availability
29	_____ situations can affect the integrity of a database	two	three	four	one	two
30	individual data items are unreadable in _____ of the database	integrity	element	user authentication	auditability	integrity
31	whole database is damaged in _____	integrity	element	user authentication	auditability	integrity
32	Integrity of the database as a whole is the responsibility in	DBMS	data	information	user authenticatio	DBMS
33	In database, it is important to be able to_____ the database at the point of a failure.	reconstruct	crashed	schema	subschema	reconstruct
34	The DBMS must maintain a _____ in integrity of the database	log of transacti	database	schema	subschema	log of transactions
35	The system can obtain ___ account balances by reverting to a backup Copy of the database and reprocessing all later transactions from the log	accurate	inaccurae	integrity of the database	schema	accurate
36	The_____ is their correctness or accuracy.	integrity of database	user authentication	auditability	database	integrity of database
37	activities that test for appropriate values in a position is done by _____	field checks	data	information	database manager	field checks
38	A _____ can undo any changes that were made in error.	database administrator	database manager	user	user authentiction	database administrator
39	For some applications it may be desirable to generate an audit record of all access (read or write) to a database is termed as _____	auditability	user authentiction	access control	controlled access	auditability
40	Databases are often separated logically by user access privileges is done by ____	auditability	user authentiction	access control	controlled access	access control
41	This authentication supplements the authentication performed by the operating system is said to be ____	auditability	user authentiction	access control	controlled access	user authentiction
42	Integrity/Confidentiality/Availability – is denotedby	Computer Security	database	data	information	Computer Security
43	_____ is a major concern in the design of database manageme	integrity	confidentially	availability	auditability	integrity
44	_____is a key issue with databases because of the inference	integrity	confidentially	availability	auditability	confidentially
45	_____ is important because of the shared access motivation underlying	integrity	confidentially	availability	auditability	availability
46	_____mean that the software runs for very long periods of time	reliability	integrity	confidentially	availability	reliability

47	Database concerns about reliability and integrity can be viewed fro	3	2	1	0	3
48	A responsible _____ backs up the files of a database periodically along with other user files.	system administrator	database administrator	user	server	system administrator
49	the data item to be modified was a long field, half of the field might show the new value, while the other half would contain the old is termed as	two-phase update	update technique	intent phase	commiting	two-phase update
50	the DBMS gathers the resources it needs to perform the update	two-phase update	update technique	intent phase	commiting	intent phase
51	_____ involves the writing of a commit flag to the database	two-phase update	update technique	intent phase	commiting	commiting
52	Database systems are often _____	multiuser systems	multiple systems	system recources	maintainability	multiuser systems
53	_____ are data that should not be made public.	Sensitive data	information	element	integrity	Sensitive data
54	The _____ or the owner of the data may have declared the data to be sensitive.	database administrator	system administrator	user	server	database administrator
55	The abbreviation for IDS is	intrusion dectection	intention dectection	integrity dection system	intrusion develop system	intrusion dectection
56	Electronic Mail is often abbreviated as _____	E-Mail	E- Message	Error Message	Exchanging digital messages	E-Mail
57	Network- based e-mail was initially exchanged on the APRANET in extension to the (FTP) but today is carried by _____	SMTP	SPTR	MAAs	MTA	SMTP
58	Multipurpose Internet Mail Extension is abbraviated as _____	MIM	IME	MIME	MPIME	MIME
59	To send mail, a system must have _____ MTA	client	server	client-server	sensitive data	client
60	The protocols that define the MTA client & server in the internet is called _____	single mail transfer protocols	simple mail transfer protocols	smallest mail tranfer protocols	simple mail target protocols	simple mail transfer protocols

UNIT V

Security Planning, Risk Analysis, Organisational Security Policy, Physical Security. Ethical issues in Security: Protecting Programs and data. Information and law.

Security planning:

Contents of security planning:

A security plan identifies and organizes the security activities for a computing system. The plan is both a description of the current situation and a plan for improvement. Every security plan must address seven issues.

1. Policy, indicating the goals of a computer security effort and the willingness of the people involved to work to achieve those goals
2. Current state, describing the status of security at the time of the plan
3. Requirements, recommending ways to meet the security goals
4. Recommended controls, mapping controls to the vulnerabilities identified in the policy and requirements
5. Accountability, describing who is responsible for each security activity
6. Timetable, identifying when different security functions are to be done
7. Continuing attention, specifying a structure for periodically updating the security plan

1. **Policy**:

The policy statement should specify the following:

- The organization's goals on security. What is the higher priority: serving customers or securing data?
- Where the responsibility for security lies.
- The organization's commitment to security.

2. **Current Security Status:**

To be able to plan for security, an organization must understand the vulnerabilities to which it may be exposed. The organization can determine the vulnerabilities by performing a risk analysis: a careful investigation of the system, its environment, and the things that might go

wrong. The risk analysis forms the basis for describing the current status of security. The status can be expressed as a listing of organizational assets, the security threats to the assets, and the controls in place to protect the assets. The status portion of the plan also defines the limits of responsibility for security. It describes not only which assets are to be protected but also who is responsible for protecting them. The plan may note that some groups may be excluded from responsibility; for example, joint ventures with other organizations may designate one organization to provide security for all member organizations. The plan also defines the boundaries of responsibility, especially when networks are involved. For instance, the plan should clarify who provides the security for a network router or for a leased line to a remote site. Even though the security plan should be thorough, there will necessarily be vulnerabilities that are not considered. These vulnerabilities are not always the result of ignorance rather, they can arise from the addition of new equipment or data as the system evolves. They can also result from new situations, such as when a system is used in ways not anticipated by its designers. The security plan should detail the process to be followed when someone identifies a new vulnerability.

3. **Requirements**:

The heart of the security plan is its set of security requirements: functional or performance demands placed on a system to ensure a desired level of security. The requirements are usually derived from organizational needs. Sometimes these needs include the need to conform to specific security requirements imposed from outside, such as by a government agency or a commercial standard.

4. **Recommended Controls**:

The security requirements lay out the system's needs in terms of what should be protected. The security plan must also recommend what controls should be incorporated into the system to meet those requirements. Throughout this book you have seen many examples of controls, so we need not review them here. As we see later in this chapter, we can use risk analysis to create a map from vulnerabilities to controls. The mapping tells us how the system will meet the security

requirements. That is, the recommended controls address implementation issues: how the system will be designed and developed to meet stated security requirements.

5. **Responsibility for Implementation**:

A section of the security plan should identify which people are responsible for implementing the security requirements. This documentation assists those who must coordinate their individual responsibilities with those of other developers. At the same time, the plan makes explicit who is accountable should some requirement not be met or some vulnerability not be addressed. That is, the plan notes that is responsible for implementing controls when a new vulnerability is discovered or a new kind of asset is introduced. People building, using, and maintaining the system play many roles. Each role can take some responsibility for one or more aspects of security. Consider, for example, the groups listed here.

- Personal computer users may be responsible for the security of their own machines. Alternatively, the security plan may designate one person or group to be coordinator of personal computer security.
- Project leaders may be responsible for the security of data and computations.

6. **Timetable**:

A comprehensive security plan cannot be executed instantly. The security plan includes a timetable that shows how and when the elements of the plan will be performed. These dates also give milestones so that management can track the progress of implementation.

7. **Continuing Attention**:

Good intentions are not enough when it comes to security. We must not only take care in defining requirements and controls, but we must also find ways for evaluating a system's security to be sure that the system is as secure as we intend it to be. Thus, the security plan must call for reviewing the security situation periodically. As users, data, and equipment change, new exposures may develop. In addition, the current means of control may become obsolete or ineffective (such as when faster processor times enable attackers to break an encryption algorithm). The inventory of objects and the list of controls should periodically be scrutinized and updated, and risk analysis performed anew.

Security Planning Team Members:

Security in operating systems and networks requires the cooperation of the systems administration staff. Program security measures can be understood and recommended by applications programmers. Physical security controls are implemented by those responsible for general physical security, both against human attacks and natural disasters. Finally, because controls affect system users, the plan should incorporate users' views, especially with regard to usability and the general desirability of controls.

Thus, no matter how it is organized, a security planning team should represent each of the following groups.

- Computer hardware group
- System administrators
- Systems programmers
- Applications programmers
- Data entry personnel
- Physical security personnel
- Representative users

In some cases, a group can be adequately represented by someone who is consulted at appropriate times, rather than a committee member from each possible constituency being enlisted.

Assuring Commitment To a security plan:

After the plan is written, it must be accepted and its recommendations carried out. Acceptance by the organization is key; a plan that has no organizational commitment is simply a plan that collects dust on the shelf. Commitment to the plan means that security functions will be implemented and security activities carried out. Three groups of people must contribute to making the plan a success.

- The planning team must be sensitive to the needs of each group affected by the plan.

- Those affected by the security recommendations must understand what the plan means for the way they will use the system and perform their business activities. In particular, they must see how what they do can affect other users and other systems.
- Management must be committed to using and enforcing the security aspects of the system.

Management commitment is obtained through understanding. But this understanding is not just a function of what makes sense technologically; it also involves knowing the cause and the potential effects of lack of security. Managers must also weigh tradeoffs in terms of convenience and cost. The plan must present a picture of how cost effective the controls are, especially when compared to potential losses if security is breached without the controls. Thus, proper presentation of the plan is essential, in terms that relate to management as well as technical concerns.

Management is often reticent to allocate funds for controls until the value of those controls is explained. By describing vulnerabilities in financial terms and in the context of ordinary business activities (such as leaking data to a competitor or an outsider), security planners can help managers understand the need for controls.

The plans we have just discussed are part of normal business. They address how a business handles computer security needs. Similar plans might address how to increase sales or improve product quality, so these planning activities should be a natural part of management. Next we turn to two particular kinds of business plans that address specific security problems: coping with and controlling activity during security incidents.

Business Continuity Plan:

A business continuity plan documents how a business will continue to function during a computer security incident. An ordinary security plan covers computer security during normal times and deals with protecting against a wide range of vulnerabilities from the usual sources.

A business continuity plan deals with situations having two characteristics:

- Catastrophic situations, in which all or a major part of a computing capability is suddenly unavailable

- Long duration, in which the outage is expected to last for so long that business will suffer

There are many situations in which a business continuity plan would be helpful.

- A fire destroys a company's entire network.
- A seemingly permanent failure of a critical software component renders the computing system unusable.
- A business must deal with the abrupt failure of its supplier of electricity, telecommunications, network access, or other critical service.
- A flood prevents the essential network support staff from getting to the operations center.

The key to coping with such disasters is advance planning and preparation, identifying activities that will keep a business viable when the computing technology is disabled. The steps in business continuity planning are these:

- Assess the business impact of a crisis.
- Develop a strategy to control impact.
- Develop and implement a plan for the strategy

Incident response plan:

Incident response Plan should be

- define what constitutes an incident
- identify who is responsible for taking charge of the situation
- describe the plan of action

Risk Analysis:

We distinguish a risk from other project events by looking for three things,

1. A loss associated with an event. The event must generate a negative effect: compromised security, lost time, diminished quality, lost money, lost control, lost understanding, and so on. This loss is called the risk impact.

2. The likelihood that the event will occur. The probability of occurrence associated with each risk is measured from 0 (impossible) to 1 (certain). When the risk probability is 1, we say we have a problem.

3. The degree to which we can change the outcome. We must determine what, if anything, we can do to avoid the impact or at least reduce its effects. Risk control involves a set of actions to reduce or eliminate the risk.

We usually want to weigh the pros and cons of different actions we can take to address each risk. To that end, we can quantify the effects of a risk by multiplying the risk impact by the risk probability, yielding the risk exposure. For example, if the likelihood of virus attack is 0.3 and the cost to clean up the affected files is $10,000, then the risk exposure is $3,000. So we can use a calculation like this one to decide that a virus checker is worth an investment of $100, since it will prevent a much larger potential loss. Clearly, risk probabilities can change over time, so it is important to track them and plan for events accordingly.

Risk is inevitable in life: Crossing the street is risky but that does not keep us from doing it. We can identify, limit, avoid, or transfer risk but we can seldom eliminate it. In general, we have three strategies for dealing with risk:

1. **Avoiding the risk**, by changing requirements for security or other system characteristics
2. **Transferring the risk**, by allocating the risk to other systems, people, organizations, or assets; or by buying insurance to cover any financial loss should the risk become a reality
3. **Assuming the risk**, by accepting it, controlling it with available resources, and preparing to deal with the loss if it occurs

Thus, costs are associated not only with the risk's potential impact but also with reducing it. Risk leverage is the difference in risk exposure divided by the cost of reducing the risk.

The Nature of Risk:

In our everyday lives, we take risks. In crossing the road, eating oysters, or playing the lottery, we take the chance that our actions may result in some negative result such as being injured, getting sick, or losing money. Consciously or unconsciously, we weigh the benefits of

taking the action with the possible losses that might result. Just because there is a risk to a certain act we do not necessarily avoid it; we may look both ways before crossing the street, but we do cross it. In building and using computing systems, we must take a more organized and careful approach to assessing our risks. Many of the systems we build and use can have a dramatic impact on life and health if they fail. For this reason, risk analysis is an essential part of security planning.

We cannot guarantee that our systems will be risk free; that is why our security plans must address actions needed should an unexpected risk become a problem. And some risks are simply part of doing business; for example, as we have seen, we must plan for disaster recovery, even though we take many steps to avoid disasters in the first place. When we acknowledge that a significant problem cannot be prevented, we can use controls to reduce the seriousness of a threat. For example, you can back up files on your computer as a defense against the possible failure of a file storage device. But as our computing systems become more complex and more distributed, complete risk analysis becomes more difficult and time consuming and more essential.

Steps of a Risk Analysis:

Risk analysis is performed in many different contexts; for example, environmental and health risks are analyzed for activities such as building dams, disposing of nuclear waste, or changing a manufacturing process. Risk analysis for security is adapted from more general management practices, placing special emphasis on the kinds of problems likely to arise from security issues. By following well-defined steps, we can analyze the security risks in a computing system.

The basic steps of risk analysis are listed below.

1. Identify assets.
2. Determine vulnerabilities.
3. Estimate likelihood of exploitation.
4. Compute expected annual loss.
5. Survey applicable controls and their costs.

6. Project annual savings of control.

Arguments For and against risk analysis:

Risk analysis is a well-known planning tool, used often by auditors, accountants, and managers. In many situations, such as obtaining approval for new drugs, new power plants, and new medical devices, a risk analysis is required by law in many countries. There are many good reasons to perform a risk analysis in preparation for creating a security plan.

- Improve awareness
- Relate security mission to management objectives.
- Identify assets, vulnerabilities, and controls
- Improve basis for decisions

Organizational Security Policies:

A security policy is a high-level management document to inform all users of the goals of and constraints on using a system. A policy document is written in broad enough terms that it does not change frequently. The information security policy is the foundation upon which all protection efforts are built. It should be a visible representation of priorities of the entire organization, definitively stating underlying assumptions that drive security activities. The policy should articulate senior management's decisions regarding security as well as asserting management's commitment to security. To be effective, the policy must be understood by everyone as the product of a directive from an authoritative and influential person at the top of the organization.

Purpose:

Security policies are used for several purposes, including the following:

- recognizing sensitive information assets
- clarifying security responsibilities
- promoting awareness for existing employees
- guiding new employees

Audience:

A security policy addresses several different audiences with different expectations. That is, each group users, owners, and beneficiaries uses the security policy in important but different ways.

Users

Users legitimately expect a certain degree of confidentiality, integrity, and continuous availability in the computing resources provided to them. Although the degree varies with the situation, a security policy should reaffirm a commitment to this requirement for service. Users also need to know and appreciate what is considered acceptable use of their computers, data, and programs. For users, a security policy should define acceptable use.

Owners

Each piece of computing equipment is owned by someone, and the owner may not be a system user. An owner provides the equipment to users for a purpose, such as to further education, support commerce, or enhance productivity. A security policy should also reflect the expectations and needs of owners.

Beneficiaries

A business has paying customers or clients; they are beneficiaries of the products and services offered by that business. At the same time, the general public may benefit in several ways: as a source of employment or by provision of infrastructure.

Contents:

A security policy must identify its audiences: the beneficiaries, users, and owners. The policy should describe the nature of each audience and their security goals. Several other sections are required, including the purpose of the computing system, the resources needing protection, and the nature of the protection to be supplied.

- Purpose
- Protected resources
- Nature of protection

Characteristics of a Good Security Policy:

If a security policy is written poorly, it cannot guide the developers and users in providing appropriate security mechanisms to protect important assets. Certain characteristics make a security policy a good one.

- Durability
- Realism
- Usefulnes

Physical security

Physical security is the term used to describe protection needed outside the computer system. Typical physical security controls include guards, locks, and fences to deter direct attacks. In addition, there are other kinds of protection against less direct disasters, such as floods and power outages; these, too, are part of physical security.

Natural Disasters:

It is impossible to prevent natural disasters, but through careful planning it is possible to reduce the damage they inflict. Some measures can be taken to reduce their impact. Because many of these perils cannot be prevented or predicted, controls focus on limiting possible damage and recovering quickly from a disaster. Issues to be considered include the need for offsite backups, the cost of replacing equipment, the speed with which equipment can be replaced, the need for available computing power, and the cost or difficulty of replacing data and programs. Some of them are

- Flood
- Fire
- Other natural disasters

Power loss:

Computers need their food electricity and they require a constant, pure supply of it. With a direct power loss, all computation ceases immediately. Because of possible damage to media by sudden loss of power, many disk drives monitor the power level and quickly retract the recording head if power fails. For certain time-critical applications, loss of service from the

system is intolerable; in these cases, alternative complete power supplies must be instantly available.

Human vandals:

Because computers and their media are sensitive to a variety of disruptions, a vandal can destroy hardware, software, and data. Human attackers may be disgruntled employees, bored operators, saboteurs, people seeking excitement, or unwitting bumblers. If physical access is easy to obtain, crude attacks using axes or bricks can be very effective. One man recently shot a computer that he claimed had been in the shop for repairs many times without success. Physical attacks by unskilled vandals are often easy to prevent; a guard can stop someone approaching a computer installation with a threatening or dangerous object. When physical access is difficult, more subtle attacks can be tried, resulting in quite serious damage. People with only some sophisticated knowledge of a system can short-circuit a computer with a car key or disable a disk drive with a paper clip. These items are not likely to attract attention until the attack is completed.

- Unauthorized access and use
- Theft
- Preventing access
- Preventing portability
- Detecting theft

Interception of Sensitive Information:

When disposing of a draft copy of a confidential report containing its sales strategies for the next five years, a company wants to be especially sure that the report is not reconstructable by one of its competitors. When the report exists only as hard copy, destroying the report is straightforward, usually accomplished by shredding or burning. But when the report exists digitally, destruction is more problematic. There may be many copies of the report in digital and paper form and in many locations (including on the computer and on storage media). There may

also be copies in backups and archived in e-mail files. Here, we look at several ways to dispose of sensitive information. They are

- Shredding
- Overwriting magnetic data
- Degaussing
- Protecting against Emanation

Contingency Planning:

The key to successful recovery is adequate preparation. Seldom does a crisis destroy irreplaceable equipment; most computing systems personal computers to mainframes are standard, off-the-shelf systems that can be easily replaced. Data and locally developed programs are more vulnerable because they cannot be quickly substituted from another source. Let us look what to do after a crisis occurs.

- Back-up
- Off-site backup
- Network storage
- Cold site
- Hot site

Physical security backup:

We have to protect the facility against many sorts of disasters, from weather to chemical spills and vehicle crashes to explosions. It is impossible to predict what will occur or when. The physical security manager has to consider all assets and a wide range of harm. Malicious humans seeking physical access are a different category of threat agent. The primary physical controls are strength and duplication. Strength means overlapping controls implementing a defense-in-depth approach so that if one control fails, the next one will protect. People who built ancient castles practiced this philosophy with moats, walls, drawbridges, and arrow slits. Duplication means eliminating single points of failure. Redundant copies of data protect against harm to one copy from any cause. Spare hardware components protect against failures.

Ethical issues in Security:

Protecting Programs and data

Copyrights, patents, and trade secrets are legal devices that can protect computers, programs and data. Here how each of these forms are originally designed to be used and how each is currently used in computing are described.

Copyrights:

Copyrights are designed to protect the expression of ideas. Thus it is applicable to a creative work, such as story, photographs, song or pencil sketch. The right to copy an expression of an idea is protected by copyright. The idea of copyright is to allow regular and free exchange of ideas. Copyright gives the author the exclusive right to make copies of the expression and sell them in public. That is, only the author can sell the copies of the author's book.

Patents:

Patents are unlike copyrights in that they protect inventions, tangible objects, or ways to make them, not works of the mind. The distinction between patents and copyrights is that patents were intended to apply to the results of science, technology, and engineering, where as copyrights are meant to cover works in the arts, literature, and written in the scholarship. A Patent is designed to protect the device or process for carrying out an idea itself.

Trade Secrets:

A trade secret is unlike a patent and copyright in that it must kept secret. The information has value only as secret, and an infringer is one who divulges the secret. Once divulged, the information usually cannot be made secret. A trade secret is information that gives one company a competitive edge over others. For example the formula of a soft drink is a trade secret, as is a mailing list of customer or information about a product due to be announced in a few months.

Computer Crime:

Crimes involving computers are an area of the law that is even less clear than the other areas.

Issues in computer crime are

- Rules of property
- Rules of evidence

- Threats to integrity and confidentiality
- Value of data
- Acceptance of computer terminology

Why Computer crime is hard to define?

Some people in the legal process do not understand computers and computing, so crimes involving computers are not always treated properly. Main reasons are

1. Lack of understanding
2. Lack of physical evidence
3. Lack of recognition assets
4. Lack of political impacts
5. Complexity of case
6. Juveniles

Privacy:

In particular, we want to investigate the privacy of sensitive data about the user. The user should be protected against the system's misuse of the private data and the system's failure to protect its user's private data against outside attack and disclosure. This is termed as privacy in computer ethics.

Ethical Issues in Computer Security:

The primary purpose of this section is to explore some of ethical issues associated with computer security and to show how ethics functions as a control.

Difference between Law and Ethics:

Law	Ethics
Described by formal, written documents	Described by unwritten principles
Interpreted by courts	Interpreted by each individual
Established by legislature representing all people	Presented by philosophers, religions, professional groups
Applicable to everyone	Personal choice

Priority determined by courts if two laws conflict	Priority determined by an individual if two principles conflict
Court final arbiter of "right"	No external arbiter
Enforceable by police and court	Limited enforcement

Studying Ethics:

The study of ethics is not so easy because the issues are complex. Sometimes people confuse between ethics and religion because many religions provide a framework in which to make ethical choices. Here some of the problems and how understanding of ethics can deal with issues of computer security is explained.

- Ethics and religion
- Ethical principles are not universal\Ethics does not provide answers

Solutions to the issues:

1. Ethical reasoning
2. Examining the case for ethical issues

Here some steps are used to make ethical choices justifiable. Those are

I. Understanding the situation

II. Know several theories of ethical reasoning

III. List the ethical principles involved

IV. Determine which principles outweigh others

Examples of ethical principle:

1. Consequence based principles
2. Rule based principles

Taxonomy o Ethical theories:

	Consequence based	Rule based
Individual	Based on consequences to individual	Based on rules acquired by the individual from religion experience, analysis

Universal	Based on consequences to all of society	Based on universal rules, evident to everyone

Law and Ethics in Information Security

Laws are rules adopted and enforced by governments to codify expected behavior in modern society.

The key difference between law and ethics is that law carries the sanction of a governing authority and ethics do not.

Ethics are based on cultural mores: relatively fixed moral attitudes or customs of a societal group.

The Legal Environment

The information security professional and managers involved in information security must possess a rudimentary grasp of the legal framework within which their organizations operate.

This legal environment can influence the organization to a greater or lesser extent depending on the nature of the organization and the scale on which it operates.

Types of Law

- Civil law embodies a wide variety of laws pertaining to relationships between and among individuals and organizations.
- Criminal law addresses violations harmful to society and is actively enforced and prosecuted by the state.
- Tort law is a subset of civil law which allows individuals to seek recourse against others in the event of personal, physical, or financial injury.
- Private law regulates the relationships among individuals and among individuals and organizations, and encompasses family law, commercial law, and labor law.
- Public law regulates the structure and administration of government agencies and their relationships with citizens, employees, and other governments.
- Public law includes criminal, administrative, and constitutional law

Key U.S. Laws of Interest to Information Security Professionals

Area	Act	Date	Web resource location	Description
Criminal intent	National Information Infrastructure Protection Act of 1996	1996	http://policyworks.gov/policydocs/14.pdf	Categorizes crimes based on defendant's authority to access a protected computer system *and* criminal intent
Terrorism	USA Patriot Act of 2001 (H.R. 3162)	2001	thomas.loc.gov/cgi-bin/bdquery/z?d107:H.R.3162:	Defines stiffer penalties for prosecution of terrorist crimes
Threats to computers	Computer Fraud and Abuse Act (also known as Fraud and Related Activity in Connection with Computers) (18 U.S.C. 1030)	1986 (amended 1994, 1996, and 2001)	www.usdoj.gov/criminal/cybercrime/1030_new.html	Defines and formalizes laws to counter threats from computer-related acts and offenses
Telecommunications	Communications Act of 1934 Updated By The Telecommunications Deregulation And Competition Act of 1996	1934 (amended 1996 and 2001)	www.fcc.gov/Reports/1934new.pdf	Regulates interstate and foreign telecommunications
Federal agency information security	Computer Security Act of 1987	1987	www.cio.gov/Documents/computer_security_act_Jan_1998.html	Requires all federal computer systems that contain classified information to have surety plans in place, and requires periodic security training for all individuals who operate, design, or manage such systems
Privacy	Federal Privacy Act of 1974	1974	www.usdoj.gov/foia/privstat.htm	Governs federal agency use of personal information

Table 1

Key U.S. Laws of Interest to Information Security Professionals (continued)

Area	Act	Date	Web resource location	Description
Cryptography	The Electronic Communications Privacy Act of 1986	1986	www.itpolicy.gsa.gov/itpolicy/5.pdf	Regulates interception and disclosure of electronic information; also referred to as the Federal Wiretapping Act
Banking	Gramm-Leach-Bliley Act of 1999 (GLB) or The Financial Services Modernization Act	1999	www.senate.gov/~banking/conf/	Focuses on facilitating affiliation among banks, insurance, and securities firms. It has significant impact on the privacy of personal information used by these industries
Trade secrets	The Economic Espionage Act (EEA) of 1996	1996	www.ncix.gov/pubs/online/eea_96.htm	Designed to prevent abuse of information gained while employed elsewhere
Accountability	The Sarbanes-Oxley Act of 2002 or Public Company Accounting Reform and Investor Protection Act	2002	http://news.findlaw.com/hdocs/docs/gwbush/sarbanesoxley072302.pdf	Intended as enforced accountability for executives at publicly traded companies, this law is having ripple effects throughout the accounting, IT, and related units of many organizations

Table 2

The Computer Fraud and Abuse Act of 1986 (CFA Act) is the cornerstone of many computer-related federal laws and enforcement efforts.

It was amended in October 1996 by the National Information Infrastructure Protection act of 1996, which modified several sections of the previous act, and increased the penalties for elected crimes.

The CFA Act was further modified by the USA Patriot Act of 2001—the abbreviated name for "Uniting and Strengthening America Act by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism Act of 2001," which provides law enforcement agencies with broader latitude to combat terrorism-related activities. Some of the laws modified by the Patriot Act date from the earliest laws created to deal with electronic technology.

The Communication Act of 1934 was revised by the Telecommunications Deregulation and Competition Act of 1996, which attempts to modernize the archaic terminology of the older act.

The Computer Security Act of 1987 was one of the first attempts to protect federal computer systems by establishing minimum acceptable security practices.

Privacy Laws

Many organizations collect, trade, and sell personal information as a commodity, and any individuals are becoming aware of these practices and looking to the governments to protect their privacy.

In the past it was not possible to create databases that contained personal information collected from multiple sources.

Today, the aggregation of data from multiple sources permits unethical organizations to build databases with alarming quantities of personal information.

The Privacy of Customer Information Section of the section of regulations covering common carriers specifies that any proprietary information shall be used explicitly for providing services, and not for any marketing purposes.

The Federal Privacy Act of 1974 regulates the government's use of private information. The Federal Privacy Act was created to ensure that government agencies protect the privacy of individuals' and businesses' information, and holds those agencies responsible if any portion of this information is released without permission.

The Electronic Communications Privacy Act of 1986 is a collection of statutes that regulates the interception of wire, electronic, and oral communications.

These statutes work in cooperation with the Fourth Amendment of the U.S. Constitution, which prohibits search and seizure without a warrant.

The Health Insurance Portability & Accountability Act Of 1996 (HIPPA), also known as the Kennedy-Kassebaum Act, is an attempt to protect the confidentiality and security of health care data by establishing and enforcing standards and by standardizing electronic data interchange.

HIPPA requires organizations that retain health care information to use information security mechanisms to protect this information, as well as policies and procedures to maintain them, and also requires a comprehensive assessment of the organization's information security systems, policies, and procedures.

HIPPA has five fundamental privacy principles:

1. Consumer control of medical information
2. Boundaries on the use of medical information
3. Accountability for the privacy of private information
4. Balance of public responsibility for the use of medical information for the greater good measured against impact to the individual
5. Security of health information

State and Local Regulations

It is the responsibility of information security professionals to understand state laws and regulations and ensure that their organization’s security policies and procedures comply with the laws and regulations.

POSSIBLE QUESTIONS

PART A

Q.NO 1 TO 20 (MULTIPLE CHOICE QUESTIONS)

PART B (2 MARKS)

1. Differentiate between laws and ethics
2. What are the three types of security policies?
3. Define policy
4. How will you protect programs and data
5. Define Patents
6. What are trade secrets
7. List out legal issues relating to information
8. List out the characteristics for policy
9. Define Risk Analysis
10. Specify the different strategies in risk

PART C (6 MARKS)

1. Discuss the legal and ethical issues associated with the information security.
2. Discuss the various types of security policies.
3. Illustrate in detail risk analysis.
4. List and explicate the major principles in organizational Security policy.
5. Discuss the risk analysis factor in administrating security.
6. List and explicate the major role on security plan.
7. Discuss the factors in Physical Security

KARPAGAM ACADEMY OF HIGHER EDUCTATION
DEPARTMENT OF COMPUTER SCIENCE
III B.Sc CS
INFORMATION SECURITY[16CSU501A]

UNIT 5

S.NO	Question	Option 1	Option 2	Option 3	Option 4	Answer
1	A _____ identifies and organizes the security activities for a computing system.	security plan	secure	scalable	reliable	security plan
2	describing the status of security at the time of the plan is known as____	Current state	policy	timetable	requirements	Current state
3	recommending ways to meet the security goals is	requirements	Current state	policy	timetable	requirements
4	mapping controls to the vulnerabilities identified in the policy and requirements is ___	requirements	Recommended controls	Current state	timetable	Recommended controls
5	describing who is responsible for each security activity is	Accountability	requirements	Current state	policy	Accountability
6	identifying when different security functions are to be done is ____	requirements	Current state	policy	timetable	timetable
7	specifying a structure for periodically updating the security plan is ___	Continuing attention	Current state	policy	timetable	Continuing attention
8	indicating the goals of a computer security effort and the willingness of the people involved to work to achieve those goals	Accountability	requirements	Current state	policy	policy
9	_____able to plan for security, an organization must understand the vulnerabilities to which it may be exposed.	Current Security Status	Accountability	requirements	Current state	Current Security Status
10	The organization can determine the vulnerabilities by p	risk analysis	risk factor	security	vulnerabilities	risk analysis
11	The _____ forms the basis for describing the current status of security	risk analysis	risk factor	security	vulnerabilities	risk analysis
12	The _____ of the plan also defines the limits of responsibility for security	status portion	risk analysis	requirements	timetable	status portion
13	The heart of the security plan is its set of security ___	requirements	risk analysis	risk factor	Current state	requirements

14	A section of the security plan should identify which people are responsible for _____ the security requirements	implementing	risk analysis	requirements	administrating security	implementing
15	The security plan includes a ____ that shows how and when the elements of the plan will be performed	timetable	risk analysis	requirements	administrating security	timetable
16	Security in operating systems and networks requires the ___ of the systems administration staff	cooperation	team members	status portion	none	cooperation
17	Program security measures can be understood and recommended by	applications programmers.	team members	software	network security	applications programmers.
18	_____controls are implemented by those responsible for general physical security, both against human attacks and natural disasters	physical security	software	network security	Protocols	physical security
19	The planning team must be _____ to the needs of each group affected by the plan.	sensitive	not sensitive	security	secure	sensitive
20	_____ must be committed to using and enforcing the security aspects of the system	management	physical security	sensitive	networks	management
21	how many characteristics deal with busines continuity plan	2	3	4	5	2
22	by changing requirements for security or other system characteristics	transferring the risk	avoiding the risk	assuming the risk	risk analysis	avoiding the risk
23	____ by allocating the risk to other systems, people, organizations, or assets; or by buying insurance to cover any financial loss should the risk become a reality	transferring the risk	avoiding the risk	assuming the risk	risk analysis	transferring the risk
24	_____ by accepting it, controlling it with available resources, and preparing to deal with the loss if it occurs	transferring the risk	avoiding the risk	assuming the risk	risk analysis	assuming the risk
25	____ is the difference in risk exposure divided by the cost of reducing the risk.	Risk leverage	transferring the risk	avoiding the risk	assuming the risk	Risk leverage
26	A_____ is a high-level management document to inform all users of the goals of and constraints on using a system	Risk leverage	security policy	policy	organizational policy	security policy
27	A_____ document is written in broad enough terms th	policy	security policy	vulnerabilities	organizational policy	policy
28	To describe the plan of action is said to be _____	incident response p	secure	security policy	policy	incident response plan

29	____ is a well-known planning tool, used often by auditors, accountants, and managers.	risk factor	Risk analysis	vulnerabilities	organizational policy	Risk analysis
30	If a security policy is written_____, it cannot guide the developers and users in providing appropriate security mechanisms to protect important assets	clearly	efficient	poorly	worst	poorly
31	find the odd man out	durability	usefulness	realism	policy	policy
32	____ is the term used to describe protection needed outside the computer system.	Physical security	logical security	policy	networks	Physical security
33	Copyrights, patents, and trade secrets are legal devices that can protect computers ____ and _____	programs, data	power, data	programs, denial	power, denial	programs, data
34	The idea of _____ is to allow regular and free exchange of ideas	copyright	programs, data	patents	trade	copyright
35	___ gives the author the exclusive right to make copies of the expression and sell them in public	copyright	programs, data	patents	trade	copyright
36	____ are unlike copyrights in that they protect inventions, tangible objects, or ways to make them, not works of the mind	copyright	programs, data	patents	trade	patents
37	_____ is designed to protect the device or process for carrying out an idea itself.	copyright	programs, data	Patent	trade	Patent
38	A _____ secret is unlike a patent and copyright in that it must kept secret	copyright	programs, data	Patent	trade	trade
39	computer crime is hard to define in _____	lack of understanding	programs, data	network security	networks	lack of understanding
40	Described by formal, written documents is said to be _____	law	ethics	information	law, ethics	law
41	_____described by unwritten principles	law	ethics	information	law, ethics	ethics
42	Interpreted by courts is handled by _____	law	ethics	professional etheics	law, ethics	law
43	Applicable to everyone is denoted by _____	ethics	law	law,ethics	computer security	law
44	Personal choice is denoted by	ethics	law	law,ethics	computer security	ethics
45	In law, Priority determined by____ if two laws conflict	courts	individual	security	policy	courts

46	Priority determined by an _____l if two principles conflict	courts	individual	security	policy	individual
47	Ethical principles are not _____	universal	individual	security	personal	universal
48	____ embodies a wide variety of laws pertaining to relationships between and among individuals and organizations	Civil law	Criminal law	Tort law	Private law	Civil law
49	___ is a subset of civil law which allows individuals to seek recourse against others in the event of personal, physical, or financial injury.	Civil law	Criminal law	Tort law	Private law	Tort law
50	____ regulates the relationships among individuals and among individuals and organizations, and encompasses family law, commercial law, and labor	Civil law	Criminal law	Tort law	Private law	Private law
51	_____ regulates the structure and administration of government agencies and their relationships with citizens, employees, and other governments	private law	Criminal law	Civil law	public law	Public law
52	_____ includes criminal, administrative, and constitutional law	Civil law	Public law	Criminal law	Private law	Public law
53	The CFA Act was further modified by the USA Patriot Act of _____	1986	1996	2001	2017	2001
54	The Computer Security Act of 1987 was one of the_____ attempts to protect federal computer systems by establishing minimum acceptable security practices.	first	second	third	forth	first
55	The _____ Act of 1986 is a collection of statutes that regulates the interception of wire, electronic, and oral communications	E-service communication privacy	Electronic Communications Privacy	federal privacy	computer security	Electronic Communications Privacy
56	The Health Insurance Portability & Accountability Act Of 1996 (HIPPA), also known as	kennedy Act	Kassebanum	Kennedy-Kassebaum Act	federal privacy act	Kennedy-Kassebaum Act
57	_____requires organizations that retain health care information to use information security mechanisms to protect this information, as well as policies and procedures to maintain	HIPPA	HIPPO	HIPA	HIAPP	HIPPA
58	Pretty good privacy (PGP) is used in _____	browser security	email security	FTP security	computer security	email security
59	A _____ destroys a company's entire network.	virus	fire	uncontrol access	abrupt failure	fire
60	The basic steps of risk analysis is _____	uncontrol access	Determine vulnerabilities	abrupt failure	protection of data	Determine vulnerabilities

Register Number_____

[16CSU501A]

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Coimbatore- 641021.
B.Sc COMPUTER SCIENCE
FIRST INTERNAL EXAMINATION - JULY 2018
Fifth Semester
INFORMATION SECURITY

Date & Session: 13.7.2018 & FN **Duration: 2 Hours**
Maximum : 50 Marks **Class: III – B. Sc (CS) A & B**

PART – A (20*1 = 20)
Answer All the Questions

1. A _____ is anything that can cause harm.
   a) Vulnerability b) phish c) threat d) spoof
2. The phrase _____ describes viruses, worms, Trojan horse attack applets, and attack scripts.
   a) malware b) spam c) phish d) virus
3. The abbreviation of DOS is _____
   a) Denial of Service b) data of Service c) Denial of Security d) Data on Security
4. An original text is known as _____
   a) Plain Text b) Plate Text c) Cipher text d) message
5. The process of converting from plaintext to ciphertext is known as _____.
   a) encryption b) decryption c) cryptography d) cipher text
6. The process of converting from ciphertext to plaintext is known as _____
   a) encryption b) decryption c) cryptography d) cipher text
7. _____ key is also input to the encryption.
   a) secret b) decryption c) cryptography d) integrity
8. _____ is designed to detect, prevent or recover from a security attack.
   a) security mechanism b) security attack c) security services d) cryptography
9. _____cipher reverses the order of the letters in a plaintext.
   a) transposition b) substitution c) Caesar d) Rail fence
10. _____ is a weakness in the security system.

a) vulnerability b) attacks c) threat d) active attack

11. Substitution cipher otherwise can be called as _____.

a) simple substitution b) cipher substitution c) cipher d) Caesar cipher

12. In symmetric encryption _____ key is used.

a) same b) different c) null d) same & different

13. In Asymmetric encryption _____ key is used.

a) same b) different c) null d) same & different

14. The abbreviation of DES is _____

a) Data Encryption standard b) Data Export standard

c) Data Export Security d) Data Encryption Security

15. The DES Algorithm is fixed for a _____.

a) 56- bit key b) 46- bit key c) 16- bit key d) 24- bit key

16. Using _____ keys are known as double DES.

a) 2 b) 3 c) 0 d) 6

17. Using _____ keys are known as triple DES.

a) 2 b) 3 c) 0 d) 6

18. _____ refers to assuring that a communication is authentic.

a) authentication b) access control c) data integrity d) nonrepudiation

19. Replacing each letter of the alphabet is said to be _____.

a) Caesar cipher b) play fair cipher d) hill cipher d) simple substitutions

20. The Playfair algorithm is based on the _____ rules.

a) 4 b) 3 c) 2 d) 5

PART – B (3*2 = 6)
Answer All the Questions

21. Define Security.

22. Define Computer Criminals.

23. Define Encryption.

PART – C (3*8 = 24)
Answer All the Questions

24. (a) Give in detail Security Attacks with example.

(or)

(b) Distinguish between Security Services and Security Mechanisms

25. (a) Illustrate Substitution Ciphers with examples.

(or)

(b) Demonstrate Transpositions Ciphers with examples.

26. (a) Differentiate symmetric and asymmetric encryption scheme.

(0r)

(b) Explain the operation of DES algorithm using diagram. What is the strength of a DES algorithm?

[16CSU501A]

Reg.No_____

Karpagam Academy of Higher Education
(Deemed to be university)
Department of Computer Science
First Internal Examinations – July 2018
III B.SC CS
Information Security – Answer Key

Date: 13 .7.18(FN) **Max.Marks:50**

PART – A (20*1 = 20)
Multiple Choice Questions

1. A _____ is anything that can cause harm.
a) vulnerability b) phish c) **threat** d) spoof

2. The phrase _____ describes viruses, worms, Trojan horse attack applets, and attack scripts.
a) **malware** b) spam c) phish d) virus

3. The abbreviation of DOS is _____
a) **Denial of Service** b) data of Service c) Denial of Security d) Data on Security

4. An original text is known as _____
a) **Plain Text** b) Plate Text c) Cipher text d) message

5. The process of converting from plaintext to ciphertext is known as _____.
a) **encryption** b) decryption c) cryptography d) cryptoanalysis

6. The process of converting from ciphertext to plaintext is known as _____
a) encryption b) **decryption** c) cryptography d) cryptoanalysis

7. _____ key is also input to the encryption.
a)**secret** b) decryption c) cryptography d) integrity

8. _____ is designed to detect, prevent or recover from a security attack.
a) **security mechanism** b) security attack c) security services d) cryptography

9. _____cipher reverse the order of the letters in a plaintext.
a) **transposition** b) substitution c) Caesar d) Rail fence

10. _____ is a weakness in the security system.
a) **vulnerability** b) attacks c) threat d) active attack

11.Substitution cipher otherwise can be called as _____.
a) **simple substitution** b) cipher substitution c) cipher d) Caesar cipher

12. In symmetric encryption _____ key is used.
a) **same** b) different c) null d) same & different

13. In Asymmetric encryption _____ key is used.
a) same b) **different** c) null d) same & different

14. The abbreviation of DES is _____
a) **Data Encryption standard** b) Data Export standard
c) Data Export Security d) Data Encryption Security

15. The DES Algorithm is fixed for a _____.
a) **56- bit key** b) 46- bit key c) 16- bit key d) 24- bit key

16. Using _____ keys are known as double DES.

a) **2** b) 3 c) 0 d) 6

17. Using _____ keys are known as triple DES.

a) 2 b) **3** c) 0 d) 6

18. _____ refers to assuring that a communication is authentic.

a) **authentication** b) access control c) data integrity d) nonrepudiation

19. Replacing each letter of the alphabet is said to be _____.

a) **Caesar cipher** b) play fair cipher d) hill cipher d) simple substitutions

20. The Playfair algorithm is based on the _____ rules.

a) **4** b) 3 c) 2 d) 5

PART – B (3*2 = 6)

Answer the Following Questions

21. Define Security.

Security is a continuous process of protecting an object from attack. That object may be a person, an organization such as a business, or property such as a computer system or a file.

22. Define Computer Criminals.

Computer crime, or cybercrime, is defined as any criminal activity in which computers, or a computer network, are the method or source of a crime. This encompasses a wide variety of crimes, from hacking into databases and stealing sensitive information to using computers to set up illegal activities.

23. Define Encryption.

The process of converting from plaintext to ciphertext is known as Encryption

PART – C (3*8 = 24)

Answer the Following Questions

24. (a) Give in detail Security Attacks with example.

Security Attacks:

Security attacks are those attacks on information and data to steal, delete or misuse them. These attacks are taking advantage of the weaknesses of either information technology.

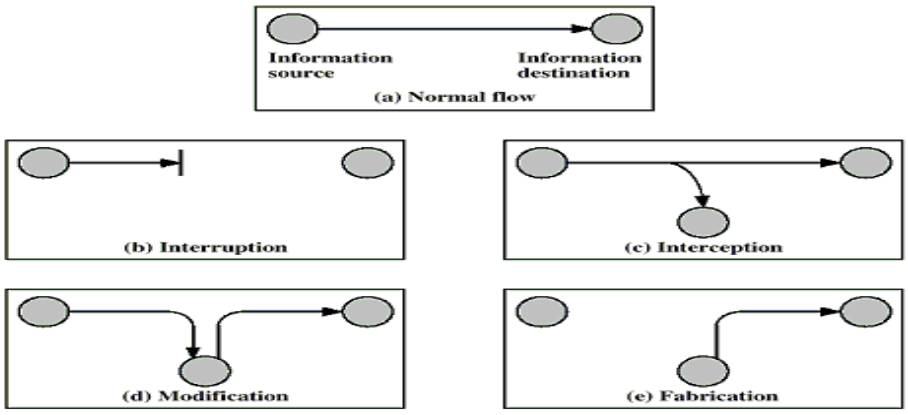
Vulnerabilities, Threats, Attacks, and Controls

Vulnerability is a weakness in the security system,

A threat to a computing system is a set of circumstances that has the potential to cause loss or harm.

An attack can also be launched by another system, as when one system sends an overwhelming set of messages to another, virtually shutting down the second system's ability to function.

To address these problems, we use a control as a protective measure. Thatis, a control is an action, device, procedure, or technique that removes or reduces vulnerability



Types of Attacks:

- Active Attacks
- Passive Attacks

Active attacks are highly malicious in nature, often locking out users, destroying memory or files, or forcefully gaining access to a targeted system or network.

Examples: Viruses, worms

A **passive attack** is an information security event or incident based on monitoring or scanning communications, information flows or systems

Examples: Traffic Analysis, Tapping

24. (b) Distinguish between Security Services and Security Mechanisms

Security Services - A service that enhances the security of data processing systems and information transfers. A security service makes use of one or more security mechanisms.

Classification of Security Services

1. Confidentiality (privacy)
2. Authentication (who created or sent the data)
3. Integrity (has not been altered)
4. Non-repudiation (the order is final)
5. Access control (prevent misuse of resources)
6. Availability (permanence, non-erasure)
    - Denial of Service Attacks

- Virus that deletes files

Security Mechanism - A mechanism that is designed todetect, prevent, or recover from a security attack.

- Secure Socket Layer (SSL) Encryption
- Authentication
- Firewalls
- Computer Anti-Virus Protection
- Data Integrity
- Ensuring Your Online Safety

25. (a) Illustrate Substitution Ciphers with examples.

Substitution ciphers

Substitution cipher, data encryption scheme in which units of the plaintext (generally single letters or pairs of letters of ordinary text) are replaced with other symbols or groups of symbols.

Types of substitution cryptosystems

- Monoalphabetic substitution
- Polyalphabetic substitution
- Homophonic substitution
- Polygraphic substitution

1. **Caesar Cipher** - This cryptosystem is generally referred to as the Shift Cipher. The concept is to replace each alphabet by another alphabet which is 'shifted' by some fixed number between 0 and 25.

2. **Play fair ciphers -** In playfair cipher, initially a key table is created. The key table is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext.

3. **Vigenere Cipher -** This scheme of cipher uses a text string (say, a word) as a key, which is then used for doing a number of shifts on the plaintext

4. **One-Time Pad**

- The length of the keyword is same as the length of the plaintext.

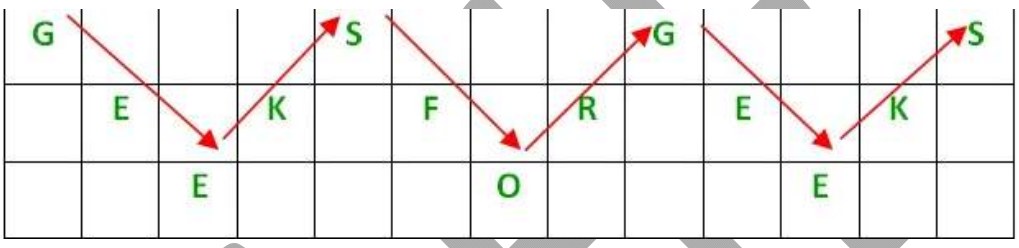
- The keyword is a randomly generated string of alphabets.
- The keyword is used only once.

25. (b) Demonstrate Transpositions Ciphers with examples.

Transpositions Ciphers - The order of the alphabets in the plaintext is rearranged to create the ciphertext. The actual plaintext alphabets are not replaced

Rail fence techniques

In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence



Row Transposition ciphers

- In general write message in a number of columns and then use some rule to read off from these columns
- key could be a series of number being the order to: read off the cipher; or write in the plain-text

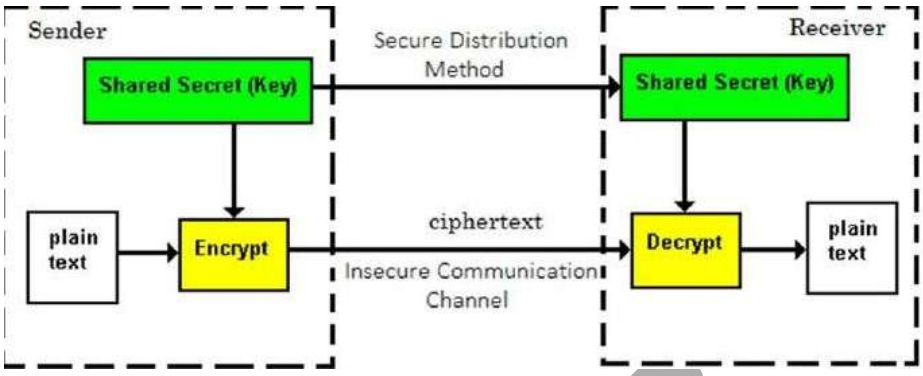
26. (a)Differentiate symmetric and asymmetric encryption scheme.

Symmetric Key Encryption

The encryption process where same keys are used for encrypting and decrypting the information is known as Symmetric Key Encryption

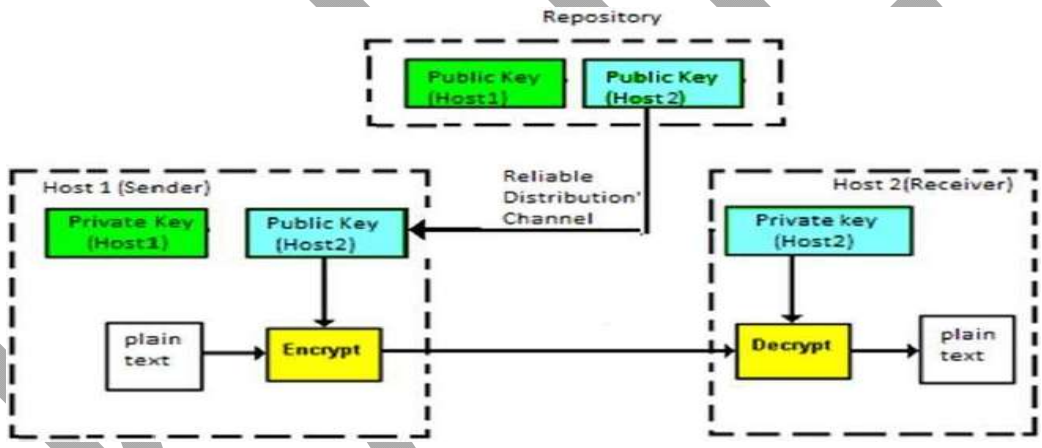
The study of symmetric cryptosystems is referred to as symmetric cryptography.

A few well-known examples of symmetric key encryption methods are − Digital Encryption Standard (DES), Triple-DES (3DES), IDEA, and BLOWFISH



Asymmetric Key Encryption

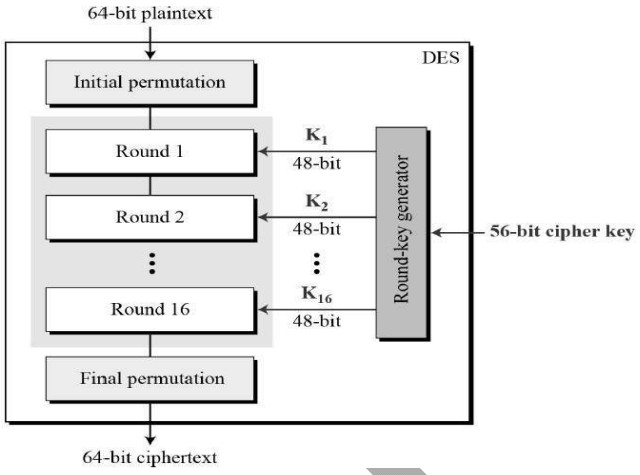
The encryption process where different keys are used for encrypting and decrypting the information is known as Asymmetric Key Encryption. Though the keys are different, they are mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible.



26. (b) Explain the operation of DES algorithm using diagram. What is the strengthof a DES algorithm?

The Data Encryption Standard (DES) is a symmetric-key block cipher.

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



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

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

Register Number_____

[16CSU501A]

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Coimbatore- 641021.
B.Sc COMPUTER SCIENCE
SECOND INTERNAL EXAMINATION –AUG' 2018
Fifth Semester
INFORMATION SECURITY

Date & Session: 14.8.2018 &FN **Duration: 2 Hours**
Maximum : 50 Marks **Class: III – B. Sc (CS) A & B**

PART – A (20*1 = 20 Marks)
Answer All the Questions

1. To reduce the problem of key proliferation by using _____ approach.
a) public key b) private key c) public & private key d) preprocessing

2. ATM stands for _____
a) Asynchronous transfer mode b) Asynchronous teller mode
c) All time teller mode d) Asynchronous target mode

3. Pre-shared public keys is so called _____
a) Diffle-Hell-man key exchange protocol b) protocol
c) diffusion d) private key

4. A digital signature is a protocol that produces the same effect as a _____
a) real signature b) reel signature c) prompt signs d) digital signature

5. In digital signature once a check is cashed, it is cancelled so that it cannot be _____
a) used b) reused c) tangible object d) confirm

6. A public key and user's identity are bound together in a _____
a) certificate b) digital c) certificate, digital d) public, private

7. In computer security, means that the information in a computer system only be accessible for reading by authorized parities.
a) Confidentiality b) integrity c) availability d) Authenticity

8. Which of the following is independent malicious program that need not any host program?
a)Trap doors b) Trojan horse c) Virus d) Worm

9. The …….. is code that recognizes some special sequence of input or is triggered by being run from a certain user ID of by unlikely sequence of events.
a) Trap doors b) Trojan horse c) Logic Bomb d) Virus

10. Which of the following malicious program do not replicate automatically?

a) Trojan horse b) Virus c) Worm d) Zombie

11. A …….. is a program that can infect other programs by modifying them, the modification includes a copy of the virus program, which can go on to infect other programs.
a) Worm b) Virus c) Trap doors d) Zombie

12. The first computer virus is --------
a) Sasser b) Creeper c) Blaster d) Virus

13. To protect a computer from virus, you should install -------- in your computer.
a) antivirus b) disk defragmenter c) disk cleanup d) backup wizard

14. Which of the following is known as Malicious software?
a) maliciousware b) illegalware c) badware d) malware

15. When a logic bomb is activated by a time related event, it is known as -------
a) trojan horse b) time bomb c) virus d) time related bomb sequence

16. Types of Flaws is divided into _____ categories.
a) 6 b) 4 c) 3 d) 2

17._____ is the computing equivalent of trying to pour 2 litres of water into a one litre.
a) overflow b) buffer overflow c) flaws d) malicious

18. Alice and Bob use two public key numbers, namely
a) P,G b) P,P c) G,G d) R,P

19.A simple public key algorithm is _____ key exchange
a) brute force b) hellman c) diffie-hellman d) none

20. Diffie-hellman key allow _____ users to exchange the key.
a) 2 b) 3 c) 4 d) 5

PART – B (3*2 = 6 Marks)
Answer All the Questions

21. What is Hash Function?
22. Define Virus. Specify the types of virus.
23. What is Salami Attack?

PART – C (3*8 = 24 Marks)
Answer All the Questions

24.(a) Give in detail on Hash Functions. [or]

(b) Explicate the concept of digital signatures

25. (a) Illustrate Diffie – Hellman Key exchange concept with neat diagram. [or]

(b) Write brief note on each of the following

i) Trapdoors ii) Salami Attack

26. (a) Enumerate malicious code [or]

(b) Explain file protection concept.

Register Number_____

[16CSU501A]

KARPAGAM ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act 1956)
Coimbatore- 641021.
B.Sc COMPUTER SCIENCE
SECOND INTERNAL EXAMINATION –AUG' 2018
Fifth Semester
INFORMATION SECURITY – ANSWER KEY

Date & Session:14.8.2018 &FN **Duration: 2 Hours**
Maximum : 50 Marks **Class: III – B. Sc (CS) A & B**

PART – A (20*1 = 20 Marks)
Answer All the Questions

1. To reduce the problem of key proliferation by using _____ approach.
a) **public key** b) private key c) public & private key d) preprocessing

2. ATM stands for _____
a) **Asynchronoustransfermode** b) Asynchronous teller mode
c) All time teller mode d) Asynchronous target mode

3. Pre-shared public keys is so called _____
a) **Diffle-Hell-man key exchange protocol** b) protocol
c) diffusion d) private key

4. A digital signature is a protocol that produces the same effect as a _____
a) **realsignature** b) reel signature c) prompt signs d) digital signature

5. In digital signature once a check is cashed, it is cancelled so that it cannot be _____
a) used b) **reused** c) tangible object d) confirm

6. A public key and user's identity are bound together in a _____
a) **certificate** b) digital c) certificate, digital d)public, private

7. In computer security, means that the information in a computer system only be accessible for reading by authorized parities.
a)**Confidentiality** b) integrity c) availability d) Authenticity

8. Which of the following is independent malicious program that need not any host program?
a)Trap doors b) Trojan horse c) Virus d) **Worm**

9. The …….. is code that recognizes some special sequence of input or is triggered by being run from a certain user ID of by unlikely sequence of events.
a) **Trap doors** b) Trojan horse c) Logic Bomb d) Virus

10. Which of the following malicious program do not replicate automatically?
a) **Trojanhorse** b) Virus c) Worm d) Zombie

11. A …….. is a program that can infect other programs by modifying them, the modification includes a copy of the virus program, which can go on to infect other programs.
a) Worm b) **Virus** c) Trap doors d) Zombie

12. The first computer virus is --------
a) Sasser b) **Creeper** c) Blaster d) Virus

13. To protect a computer from virus, you should install -------- in your computer.
a) **antivirus** b) disk defragmenter c) disk cleanup d)backup wizard

14. Which of the following is known as Malicious software?
a) maliciousware b) illegalware c) badware d) malware
15. When a logic bomb is activated by a time related event, it is known as -------
a) trojan horse b) **timebomb** c) virus d) time related bomb sequence
16. Types of Flaws is divided into _____ categories.
a) **6** b) 4 c) 3 d) 2
17._____ is the computing equivalent of trying to pour 2 litres of water into a one litre.
a) overflow b) **bufferoverflow** c) flaws d) malicious
18. Alice and Bob use two public key numbers, namely
a) **P,G** b) P,P c) G,G d) R,P
19.A simple public key algorithm is _____ key exchange
a) brute force b) hellman c) **diffie-hellman** d)none
20. Diffie-hellman key allow _____ users to exchage the key.
a) **2** b) 3 c) 4 d) 5

PART – B (3*2 = 6 Marks)
Answer All the Questions

21. What is Hash Function?

A hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The input to the hash function is of arbitrary length but output is always of fixed length.

22. Define Virus. Specify the types of virus.

A virus is aprogram that can pass on malicious code to other nonmalicious programs by modifying them. The types of virus are Trojan horse, Worm**,** Spyware & Adware, etc.

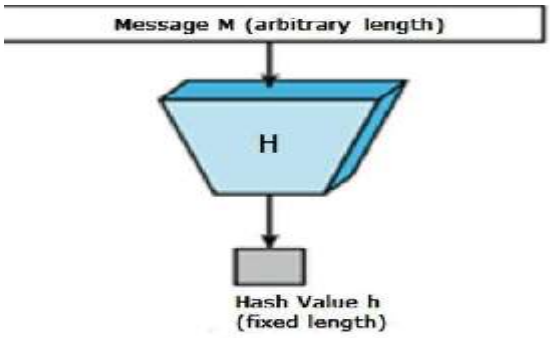
23. What is Salami Attack?

In information security, a salami attack is a series of minor attacks that together results in a larger attack.

PART – C (3*8 = 24 Marks)
Answer All the Questions

24.(a) Give in detail on Hash Functions.

A hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The input to the hash function is of arbitrary length but output is always of fixed length.



Features of Hash Functions

1. **Fixed Length Output (Hash Value)**
   - Hash function with n bit output is referred to as an n-bit hash function. Popular hash functions generate values between 160 and 512 bits

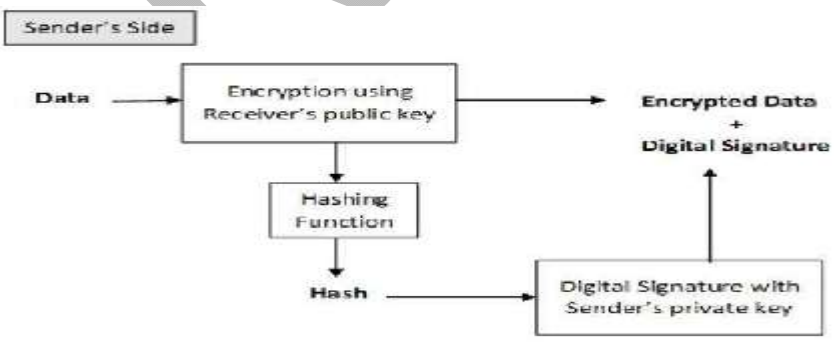
2. **Efficiency of Operation**
   - Generally for any hash function h with input x, computation of h(x) is a fast operation.

Properties of Hash Functions

- **Pre-Image Resistance -** This property means that it should be computationally hard to reverse a hash function.
- **Second Pre-Image Resistance -** This property means given an input and its hash, it should be hard to find a different input with the same hash.
- **Collision Resistance -** This property means it should be hard to find two different inputs of any length that result in the same hash. This property is also referred to as collision free hash function.

24. (b) Explicate the concept of digital signatures

Digital signatures are the public-key primitives of message authentication.



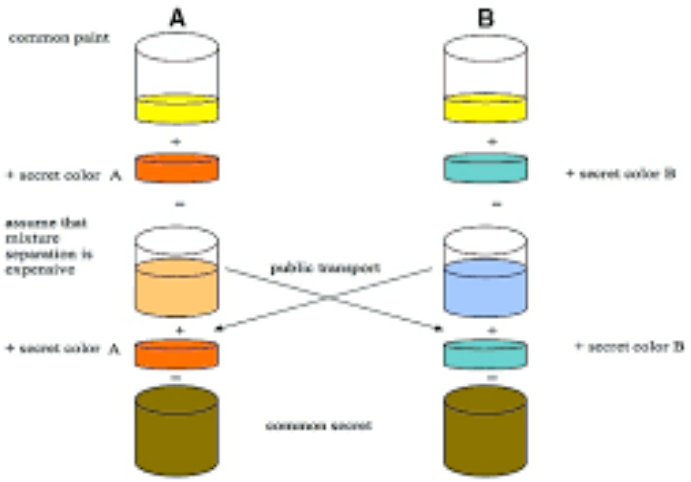
Applications

There are several reasons to implement digital signatures to communications:

- **Authentication**
- **Integrity**
- **Non-repudiation**

25. (a) Illustrate Diffie – Hellman Key exchange concept with neat diagram.

Key exchange (also key establishment) is any method in cryptography by which cryptographic keys are exchanged between two parties, allowing use of a cryptographic algorithm. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.



25. (b) Write brief note on each of the following

i) Trapdoors ii) Salami Attack

Trapdoors

A trapdoor is an undocumented entry point to a module. The trapdoor is inserted during code development, perhaps to test the module, to provide "hooks" by which to connect future modifications or enhancements or to allow access if the module should fail in the future. In addition to these legitimate uses, trapdoors can allow a programmer access to a program once it is placed in production.

Salami Attack

An attack known as a salami attack. This approach gets its name from the way odd bits of meat and fat are fused together in a sausage or salami. In the same way, a salami attack merges bits of seemingly inconsequential data to yield powerful results. For

example, programs often disregard small amounts of money in their computations, as when there are fractional pennies as interest or tax is calculated.

26. (a) Enumerate malicious code

Malicious code behaves in unexpected ways,Malicious code can do anything any other program can, such as writing a message on a computer screen, stopping a running program, generating a sound, or erasing a stored file. Or malicious code can do nothing at all right now; it can be planted to lie dormant, undetected, until some event triggers the code to act. Malicious code runs under the user's authority. Thus, malicious code can touch everything the user can touch, and in the same ways.

Kinds of Malicious Code

Code Type	Characteristics
Virus	Attaches itself to program and propagates copies of itself to other programs
Trojan horse	Contains unexpected, additional functionality
Logic bomb	Triggers action when condition occurs
Time bomb	Triggers action when specified time occurs
Trapdoor	Allows unauthorized access to functionality
Worm	Propagates copies of itself through a network
Rabbit	Replicates itself without limit to exhaust resource

26. (b) Explain file protection concept.

Basic Forms of Protection

All multi user operating systems must provide some minimal protection to keep one user from maliciously or inadvertently accessing or modifying the files of another. As the number of users has grown, so also has the complexity of these protection schemes.

All "None Protection– The user could read, modify, or delete a file belonging to any other user. Instead of soft ware -or hardware- based protection, the principal protection involved trust combined with ignorance.This all-or-none protection is unacceptable for several reasons.

- **Lack of trust**
- **All or nothing**
- **Rise of timesharing**

- **Complexity**
- **Filelistings**

Group Protection

Because the all -or-nothing approach has so many drawbacks they thought that improved way to protect files. They focused on identifying groups of users who had some common relationship.Although this protection scheme overcomes some of the shortcomings of the all – or - nothing scheme, it introduces some new difficulties of its own.

- **Group affiliation**
- **Multiple personalities**
- **All groups**
- **Limitedsharing**

Password or Other Token

We can apply a simplified form of password protection to file protection by allowing a user to assign a password to a file. User accesses are limited to those who can supply the correct password at the time the file is opened. The password can be required for any access or only for modifications (write access). file passwords suffer from difficulties similar to those of authentication passwords:

- **Loss**
- **Use**
- **Disclosure**
- **Revocation**