Questions not covered in ISEs and ESEs

Q1. Substitution Techniques in Cyber Security

Substitution techniques are classical encryption methods in which each element of the plaintext (character or bit) is replaced with another element to produce the ciphertext. The positions of the characters remain unchanged; only the symbols are substituted. These techniques aim to provide confusion, hiding the relationship between plaintext and ciphertext so attackers cannot easily determine the original message.

Types of Substitution Techniques

1 **Caesar Cipher**

- One of the earliest substitution ciphers used by Julius Caesar.
- Each letter is shifted by a fixed number of positions in the alphabet.
- Example: Shift by 3 → A→D, B→E.
- Simple but vulnerable to brute force as only 25 keys exist.

2 **Monoalphabetic Cipher**

- Uses a single substitution alphabet for the entire message.
- Each plaintext letter is replaced with a unique ciphertext letter.
- More secure than Caesar cipher but still vulnerable to frequency analysis attack due to predictable character patterns in languages.

3 **Playfair Cipher**

- Invented by Charles Wheatstone; popularized by Lord Playfair.
- Encrypts digraphs (pairs of letters) instead of single letters.
- Uses a 5×5 key matrix generated from a keyword.
- Reduces frequency analysis effectiveness but still breakable with enough ciphertext.

4 **Hill Cipher**

- Uses matrix multiplication with a key matrix to transform blocks of letters.
- Converts plaintext into vectors and multiplies by an invertible key matrix modulo 26.
- More resistant to classical attacks but vulnerable if chosen plaintext is known, as key matrix can be solved using linear algebra.

5 **Polyalphabetic Cipher**

- Uses multiple substitution alphabets to encrypt the message.

- Vigenère Cipher is a common example using a keyword to shift letters differently at different positions.
- It makes frequency analysis difficult but repeated keyword patterns may weaken security.

6⃣ **One-Time Pad (OTP)**

- A theoretically unbreakable cipher if used correctly.
- Uses a random key equal in length to the plaintext and used only once.
- No meaningful pattern for analysis, but key distribution and management are major challenges.

Conclusion

Substitution techniques are fundamental to modern cryptography by introducing confusion into encryption. While classical substitution ciphers like Caesar and Monoalphabetic are simple and easily broken, advanced methods like Hill Cipher and One-Time Pad provide stronger security principles that are still used in modified forms in modern systems.

Q2. Rotor Machines in Cybersecurity

Definition Rotor machines are electromechanical encryption devices used primarily during the early–mid 20th century (especially World War II) to perform polyalphabetic substitution automatically. They use rotating discs (called rotors) to continuously change the encryption alphabet, producing a complex cipher that is very difficult to break manually.

Structure of Rotor Machines A typical rotor machine contains:

- **Rotors (wheels):** Each rotor has electrical contacts wired internally in a random substitution pattern.
- **Entry and exit plates:** Guide electric signal flow through rotors.
- **Reflector (in some machines like Enigma):** Sends current back through rotors to create symmetric encryption (same process for encryption and decryption).
- **Keyboard and Lampboard:** For input and display of letters.
- **Mechanical stepping system:** Moves rotors after key presses to change substitution mapping each time.

Working Principle

- When a key is pressed, an electrical circuit passes through the rotors causing a letter substitution.
- After each key press, at least one rotor rotates — changing the cipher for the next letter.
- This constant rotation generates a different substitution alphabet for every character, giving a high degree of security.

Key Space and Security The strength of a rotor machine depends on:

- Number of rotors
- Internal wiring complexity
- Rotor order and starting positions
- Plugboard modifications (if present)

The combination of these parameters provides millions to billions of possible keys, making brute-force attack extremely difficult for that era.

Historical Examples

- **ENIGMA** (used by Germany in WWII)
- **SIGABA** (US cipher machine)
- **Typex** (British machine)

These were widely used for military communications before the advent of digital cryptography.

Advantages

- Automatically performs strong polyalphabetic substitution
- Very large key space
- Encryption and decryption using same machine (just reverse rotor setup)

Limitations

- Vulnerable to operational mistakes (repeat settings, predictable messages)
- Once key settings are known, encryption can be broken
- Now obsolete due to modern digital cryptography

In Summary

Rotor machines played a crucial role in the evolution of cryptography by mechanizing complex substitution ciphers. They represented a major leap in secure communication until replaced by modern electronic and algorithm-based encryption systems.

Q3. Steganography

Definition Steganography is the technique of hiding secret information within a non-secret, ordinary file or message in such a way that the presence of hidden data is not detected. It focuses on concealment of communication, unlike cryptography which focuses on scrambling the message.

Main Objective To ensure confidential and undetectable communication by embedding secret data inside media (like text, images, audio, videos) without raising suspicion.

How It Works

- A carrier file (cover medium) such as an image or audio file is selected.
- The secret data is embedded using special techniques that minimally change the visible or audible content.
- The resulting file (called stego-object) looks normal but contains hidden content.
- The receiver extracts the hidden message using the correct key or method.

Common Steganographic Techniques

1. **Text Steganography** Hiding data in spaces, punctuation, font changes, or patterns in text.
2. **Image Steganography** Most popular method — hides data in pixel values (e.g., Least Significant Bit (LSB) modification).
3. **Audio Steganography** Embeds data in sound frequencies or LSB of audio samples.
4. **Video Steganography** Hides secret data across frames of a video.
5. **Network Steganography** Conceals information inside network protocols like TCP/IP headers.

Types

- **Pure Steganography:** No cryptographic key used.
- **Secret Key Steganography:** Sender and receiver share a key to embed/extract data.
- **Public Key Steganography:** Uses public-private key pairs for secure hiding.

Applications

- Secure military and government communication
- Watermarking and copyright protection
- Covert communication by journalists or intelligence
- Authentication of digital documents

Advantages

- Hidden data is not noticeable, reducing detection risks
- Can be combined with cryptography for stronger security

Limitations

- Once detected, secrecy is lost
- Susceptible to changes like image resizing or compression
- Limited capacity for hidden data

In Summary

Steganography hides information such that an attacker cannot even detect the existence of the secret message, making it a powerful technique for covert communication. It complements cryptography by adding an extra layer of secrecy.

Q4. Social Engineering in Cyber Offenses

Social engineering is a cyber attack technique where attackers manipulate or deceive individuals into revealing confidential information or performing actions that compromise security. Instead of breaking into systems using technical methods, attackers exploit human psychology, such as trust, fear, urgency, curiosity, or authority.

Key Goals

- To obtain sensitive data (passwords, OTPs, account info)
- To gain unauthorized system access
- To spread malware or complete fraudulent transactions

Example of Social Engineering Attack

A hacker calls a bank customer pretending to be a bank officer and claims there is a security issue in their account. They ask the victim to share a OTP to verify identity. The victim shares the OTP believing the caller is legitimate, and the hacker uses it to transfer money.

(Other examples: phishing emails, fake tech support calls, malicious links, USB drops)

Classification of Social Engineering

Social engineering techniques can mainly be classified into Human-Based and Computer-Based methods:

---

1 **Human-Based Social Engineering**

In this type, attackers interact directly with people to gain trust and information.

Techniques include

- **Impersonation** → Pretending to be a trusted person (e.g., IT staff, bank worker)
- **Shoulder Surfing** → Observing someone's screen/keyboard to capture passwords
- **Tailgating / Piggybacking** → Following authorized persons into restricted areas
- **Dumpster Diving** → Searching discarded documents for confidential information
- **Baiting** → Offering something attractive (like a free gift) to trick victims

---

2️⃣ **Computer-Based Social Engineering**

Attackers misuse electronic communication to deceive users remotely.

Techniques include

- **Phishing** → Fake emails/websites to steal login credentials
- **Spear Phishing** → Targeted phishing toward a specific person/organization
- **Vishing** → Voice phishing using phone calls or IVR systems
- **Smishing** → Fraudulent SMS messages to trick users
- **Malicious Software Pop-ups** → Fake warnings to download malware
- **Fake Social Media Profiles** → Used to collect sensitive information

---

Conclusion

Social engineering attacks are highly successful because humans are the weakest link in security. Organizations must strengthen user awareness, implement verification procedures, and adopt strong security policies to prevent manipulation attacks.

Q5. Buffer Overflow – Explanation

Concept

A buffer overflow occurs when a program writes more data to a memory buffer than it is designed to hold. Buffers are fixed-size memory locations allocated for storing data temporarily. When data exceeds the limit, it overwrites adjacent memory, causing abnormal behavior.

Why it Occurs

- Poor input validation and insecure programming practices in languages like C and C++ where memory boundaries are not automatically checked.
- Attackers exploit these vulnerabilities by sending maliciously crafted input that exceeds buffer limits.

How Attackers Exploit It

- Attackers overwrite control data (e.g., return addresses, function pointers) in memory.
- They insert malicious code (shellcode) into the overflowed buffer.
- When the program executes, control jumps to the attacker's code—giving them unauthorized access or system control.

Impacts of Buffer Overflow Attacks

- Execution of arbitrary malicious code
- System crashes or denial of service (DoS)
- Privilege escalation – attacker gains admin/root access
- Installation of malware or backdoors
- Theft or corruption of sensitive data

Types of Buffer Overflow

- **Stack-based Overflow**
  - Occurs in the call stack where function variables are stored.
  - Most commonly exploited to change return addresses.
- **Heap-based Overflow**
  - Targets dynamically allocated memory (heap).
  - Used to manipulate program data structures and gain control.

Real-World Example (Conceptual)

A login form allows only 20 characters, but an attacker enters 100 characters including malicious instructions → memory around buffer gets overwritten → attacker gains shell access.

Buffer Overflow Prevention Techniques

- Bounds checking and secure coding
- Use languages with automatic memory handling (Python, Java)
- ASLR (Address Space Layout Randomization) – randomizes memory addresses
- DEP/NX-bit (Data Execution Prevention) – prevents execution of injected code in memory
- Stack canaries – special values placed to detect overflow before damage occurs
- Regular patching and code auditing

In Cybercrime Context

Hackers commonly use buffer overflow vulnerabilities to break into systems, bypass authentication, inject malware, and take remote control. It is one of the oldest yet most dangerous exploitation methods used in cyberattacks.

Q6. SQL Injection

SQL Injection is a serious web application vulnerability where an attacker inserts malicious SQL commands into input fields (such as login forms or URL parameters) to manipulate a backend database. It occurs due to improper validation or sanitization of user inputs. Attackers exploit this bug to gain unauthorized access, retrieve confidential data, modify records, or even delete entire databases.

How SQL Injection Works

- Applications take user input and form SQL queries without checking validity.
- Attackers add harmful SQL code like `OR '1'='1'` to bypass authentication.
- The database executes the query believing it is legitimate, exposing sensitive information.

Example

A normal query: `SELECT * FROM users WHERE username='admin' AND password='pass';`

Malicious input: `' OR '1'='1`

Modified query becomes: `SELECT * FROM users WHERE username='' OR '1'='1';`

It always returns true, allowing unauthorized login.

Types of SQL Injection

- **Classic/ In-band SQLi:** Attacker gets results directly from database error messages or webpage response.
- **Blind SQLi:** No direct error or data display; attacker infers information through true/false responses.
- **Error-based SQLi:** Uses database error messages to extract data structure details.

Consequences

- Unauthorized access to sensitive data (passwords, card details, personal data)
- Data modification, deletion, or corruption
- Website defacement or complete database takeover
- Financial loss and reputation damage to organization

Prevention Techniques

- Input validation and sanitization of all user inputs
- Use of prepared statements and parameterized queries
- Limiting database user privileges
- Regular security testing, firewalls (WAF), and patching vulnerabilities

---

Conclusion:

SQL Injection is a major threat to database-driven web applications. Proper security measures and secure coding practices are necessary to protect data and prevent exploitation.

Q7. Credit Card Frauds in Mobile and Wireless Computing Era

Credit card frauds have significantly increased in the mobile and wireless computing era due to the widespread use of smartphones, wireless networks, and online transactions. Criminals exploit vulnerabilities in mobile apps, unsecured Wi-Fi networks, and digital payment gateways to steal cardholder information. The mobility and portability of devices also make users more susceptible to attacks and unauthorized access.

Types and Techniques of Credit Card Frauds

- **Skimming and Cloning:** Attackers use malicious card readers or NFC-based skimmers to capture card data from mobile-enabled cards and clone them for illegal transactions.
- **Phishing and Social Engineering:** Fake mobile apps, SMS links, and emails are used to trick users into revealing card details, CVV, or OTPs.
- **Wireless Sniffing & Man-in-the-Middle Attacks:** Unsecured Wi-Fi networks allow attackers to intercept sensitive data during online transactions and capture credit card information.
- **Malware and Spyware Applications:** Fraudsters distribute malicious apps that secretly record keystrokes or access stored card data and transmit it to hackers.
- **Mobile App Exploits:** Weak authentication, insecure storage, and vulnerabilities in mobile payment apps enable attackers to perform unauthorized transactions.
- **SIM Swap Fraud:** Criminals take control of a user's mobile number by swapping SIM cards, allowing access to OTPs and banking authentication codes.

Impacts of These Frauds

- Financial losses to customers and financial institutions.
- Loss of privacy and identity theft due to compromised personal data.
- Decreased trust in mobile banking and e-commerce platforms.
- Legal and economic burdens for banks to enhance fraud detection systems.

Security Measures to Prevent Fraud

- Strong encryption and secure communication protocols (SSL/TLS).
- Multi-factor authentication like biometrics and OTPs.
- Regular software updates and security patches for mobile apps.
- Awareness to avoid phishing links and use trusted applications only.
- Use of secure payment gateways and monitoring by fraud detection systems.

Conclusion:

In the mobile and wireless computing era, credit card frauds have become more advanced and widespread. Ensuring secure transaction environments, user awareness, and strong security mechanisms are essential to reduce the risks and protect financial data.

Difference Between Worms and Viruses

Let us compare both in the form of the table below :

Basis of Comparison	Worms	Viruses
Definition	A Worm is a form of malware that replicates itself and can spread to different computers via a Network.	A Virus is a malicious executable code attached to another executable file that can be harmless or can modify or delete data.
Objective	The main objective of worms is to eat the system's resources. It consumes system resources such as memory and bandwidth and makes the system slow in speed to such an extent that it stops responding.	The main objective of viruses is to modify the information.
Host	It doesn't need a host to replicate from one computer to another.	It requires a host is needed for spreading.
Harmful	It is less harmful as compared.	It is more harmful.
Detection and Protection	They can be detected and removed by the Antivirus and firewall.	Antivirus software is used for protection against viruses.
Controlled by	They can be controlled by remote.	They can't be controlled by remote.
Execution	They are executed via weaknesses in the system.	They are executed via executable files.
Comes from	Worms generally come from the downloaded files or through a network connection.	They generally come from shared or downloaded files.
Symptoms	1. Hampering computer performance by slowing down it 2. Automatic opening and running of programs 3. Sending of emails without your knowledge	1. Pop-up windows linking to malicious websites 2. Hampering computer performance by slowing down it 3. After booting, starting of unknown programs.
Examples	Examples of worms include Morris worm, storm worm, etc.	Examples of viruses include Creeper, Blaster, Slammer, etc.
Interface	It does not need human action to replicate.	It needs human action to replicate.
Speed	Its spreading speed is faster.	Its spreading speed is slower as compared to worms.

(1996).

Hacking and the Indian Laws:

Section Ref. and Title	Chapter of the Act And Title	Crime	Punishment
Sec.43 (Penalty for damage to computer, computer system etc)	Chapter IX Penalties and Adjudication	Damage to computer system etc.	Compensation for Rs. 1 Crore
Sec.66 (Hacking with computer system)	Chapter XI Offences	Hacking (with intent or knowledge)	Fine of Rs. 2 Lakhs & Imprisonment for 3 years
Sec.67 (Publishing of information which is obscene in electronic form)	Chapter XI Offences	Publication of obscene material in electronic form	Fine of Rs. 1 Lakh & Imprisonment of 5 years and double conviction on second offence
Sec.68 (Power of controller to give directions)	Chapter XI Offences	Not complying with directions of controller	Fine upto Rs. 2 Lakhs & Imprisonment of 3 years
Sec.70 (Protected System)	Chapter XI Offences	Attempting or securing access to computer of another person without his/her knowledge	Imprisonment up to 10 Years
Sec.72 (Penalty for breach of confidentiality and privacy)	Chapter XI Offences	Attempting or securing access to computer for breaking confidentiality of the information of computer	Fine up to Rs. 1 Lakh and Imprisonment up to 2 Years
Sec.73 (Penalty for publishing Digital Signature Certificate false in certain particulars)	Chapter XI Offences	Publishing false Digital Signatures, false in certain particulars	Fine of Rs.1 Lakh or imprisonment of 2 years or both
Sec.74 (Publication for fraudulent purpose)	Chapter XI Offences	Publishing of Digital Signatures for fraudulent purpose	Imprisonment for the term of 2 years and fine of Rs. 1 Lakh

Table: The key provisions under the Indian ITA 2000 (before the amendment)