

Perfect Secrecy and OTP



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NITK-SURATHKAL

II Sem M.Tech (CSE-IS) Sub: Network Security Surprise Test :01

Date: 30-1-2025 Max Marks: 25

Note : Answer all the questions. Missing data may be suitably assumed.

1. What is Perfect Secrecy? Describe a system that achieves it. -05 Marks
2. Explain briefly the concepts: one-way function, one-way hash function, trapdoor one-way function. -05 Marks
3. Describe the three main concerns with the use of passwords for authentication. Explain what is meant by a social engineering attack on a password. -05 Marks
4. Explain how access control lists are used to represent access control matrices. Describe the environments in which they are widely used and their advantages and disadvantages. -05 Marks
5. An ideal password authentication scheme has to withstand a number of attacks. Describe five of these attacks. -05 Marks

Scanned with OKEN Scanner

Give me answer to question 1

Perfect Secrecy

A cryptosystem is said to have **perfect secrecy** if the ciphertext gives **no information at all** about the plaintext.

Formally, a system has perfect secrecy if the probability of any plaintext message remains the same even after observing the ciphertext:

$$P(M | C) = P(M) \quad \text{for all messages } M \text{ and ciphertexts } C$$

This means that knowing the ciphertext does **not** help an attacker guess the plaintext, even with unlimited computational power.

System Achieving Perfect Secrecy: One-Time Pad (OTP)

The **One-Time Pad** is the only known encryption system that achieves perfect secrecy.

Description:

- The plaintext is combined with a **truly random key** of the **same length** as the plaintext.
- Encryption is usually done using XOR (or modular addition).
- Each key is used **only once** and then discarded.

Why OTP achieves perfect secrecy:

- Every possible plaintext is equally likely for a given ciphertext.
- Without the key, the ciphertext provides no statistical information about the message.
- Even an attacker with infinite computational power cannot break it.

Conditions required for perfect secrecy in OTP:

1. The key must be **truly random**.
2. The key length must be **equal to the message length**.
3. The key must be **used only once**.
4. The key must be kept **completely secret**.

Limitation:

Although OTP is theoretically secure, it is impractical in real systems due to difficulties in secure key generation, storage, and distribution.

✓ Hence, **Perfect Secrecy is achievable in theory (via OTP), but difficult to deploy in practice.**



Answer question 2

One-Way Function, One-Way Hash Function, and Trapdoor One-Way Function

One-Way Function

A **one-way function** is a function that is **easy to compute** in the forward direction but **computationally infeasible to invert**.

- Forward computation:

$$y = f(x) \text{ is easy}$$

- Inversion:

$$x = f^{-1}(y) \text{ is infeasible}$$

Example: Multiplying two large prime numbers is easy, but factoring their product is hard.

One-Way Hash Function

A **one-way hash function** maps data of **arbitrary length** to a **fixed-length hash value** and is infeasible to reverse.

Properties:

1. **Preimage resistance** – given h , it is hard to find x such that $H(x) = h$.
2. **Second preimage resistance** – hard to find another $x' \neq x$ with the same hash.
3. **Collision resistance** – hard to find any two different inputs with the same hash.

Example: SHA-256 used for password storage and data integrity.

Trapdoor One-Way Function

A **trapdoor one-way function** is a special type of one-way function that is **easy to invert only if a secret trapdoor is known**.

- Without trapdoor: inversion is infeasible.
- With trapdoor: inversion becomes easy.

Example:

- **RSA algorithm**
 - Encryption: easy using the public key
 - Decryption: feasible only using the private key (trapdoor)

Summary

- **One-way function:** Hard to reverse for everyone.
- **One-way hash function:** One-way function producing fixed-length output for integrity and passwords.
- **Trapdoor one-way function:** Hard to invert unless secret information is available.

✓ These concepts are fundamental to modern cryptography and secure communication.



explain question 3

Question 3: Passwords for Authentication & Social Engineering Attacks

Three Main Concerns with Using Passwords

1. Weak or Guessable Passwords

- Users often choose **simple, short, or predictable passwords** (e.g., name, date of birth, "123456").
- Such passwords are vulnerable to **dictionary attacks** and **brute-force attacks**.

2. Password Reuse

- The same password is often reused across **multiple services**.
- If one website is compromised, attackers can use the leaked password to access **other accounts** (credential stuffing).

3. Password Theft

Passwords can be stolen using:

- **Keyloggers**
- **Phishing websites**
- **Shoulder surfing**
- **Malware**

Once stolen, the attacker can directly impersonate the user.

Social Engineering Attack on a Password

A **social engineering attack** is an attack where the adversary **tricks the user into revealing their password**, rather than breaking it technically.

How it Works:

- The attacker pretends to be a **trusted entity** (IT admin, bank, email provider).
- The victim is asked to **enter or share their password** due to:
 - Account verification
 - Security warning
 - Urgent request

Example:

- A fake email saying:
"Your account will be suspended. Click here and log in immediately."
- The link leads to a **fake login page**.
- The user enters credentials, which are captured by the attacker.

Summary

- Password systems suffer due to **human behavior**, not just technical flaws.
- Social engineering bypasses cryptography by **exploiting trust and fear**.
- This makes password-based authentication inherently vulnerable.

✅ This is why modern systems use **multi-factor authentication (MFA)** to strengthen security.



answer question 4

Question 4: Access Control Lists (ACLs) and Access Control Matrices

Access Control Matrix

An **access control matrix** is a conceptual model that represents the permissions of subjects on objects.

- **Rows** → Subjects (users, processes)
- **Columns** → Objects (files, resources)
- **Entries** → Access rights (read, write, execute)

Example:

	File1	File2
User1	R,W	R
User2	R	—

Access Control List (ACL) Representation

An **Access Control List (ACL)** is a **column-wise representation** of the access control matrix.

- Each **object** has a list specifying:
 - Which **subjects** can access it
 - What **permissions** they have

Example (for File1):

```

mathematica

File1 ACL:
User1 → Read, Write
User2 → Read
  
```

Thus, ACLs store only **non-empty entries** of the access control matrix, making them efficient in practice.

Environments Where ACLs Are Widely Used

- **Operating Systems**
 - UNIX/Linux file permissions
 - Windows NTFS ACLs
- **Databases**
 - Table and view access permissions

- **Network Devices**
 - Routers and firewalls for packet filtering
- **Cloud Systems**
 - Access to storage buckets and services

Advantages of ACLs

1. **Easy to manage per object**
2. **Efficient storage** (only permitted entries stored)
3. **Supports fine-grained permissions**
4. **Widely implemented and well understood**

Disadvantages of ACLs

1. **Difficult to determine all accesses of a subject**
2. **Poor scalability** in very large systems
3. **Revocation of user access can be complex**
4. **Not ideal for dynamic environments**

Summary

- ACLs are a **practical implementation** of the access control matrix.
- They are best suited for **object-centric security models**.
- Despite limitations, ACLs remain one of the most commonly used access control mechanisms.

✓ Hence, ACLs effectively translate theoretical access control matrices into real-world systems.



answer question 5

Question 5: Attacks an Ideal Password Authentication Scheme Must Withstand

An ideal password authentication system should be resistant to multiple types of attacks. **Five important attacks are described below:**

1. Brute Force Attack

- The attacker tries **all possible password combinations** until the correct one is found.

- **Defense:** Strong password policies, account lockout, rate limiting.
-

2. Dictionary Attack

- Uses a **predefined list of common passwords** and words instead of random guesses.
 - Very effective against weak or predictable passwords.
 - **Defense:** Enforce complex, uncommon passwords.
-

3. Replay Attack

- Attacker captures authentication data and **reuses it** to gain access.
 - **Defense:** Nonces, timestamps, challenge-response mechanisms.
-

4. Eavesdropping Attack

- Passwords are intercepted while being transmitted over the network.
 - Common in plaintext or weakly encrypted protocols.
 - **Defense:** Secure channels like SSL/TLS, encrypted authentication.
-

5. Phishing / Social Engineering Attack

- User is tricked into **revealing their password** through fake websites or emails.
 - **Defense:** User awareness, MFA, anti-phishing mechanisms.
-

Summary

An ideal password system must protect against:

- Guessing attacks (brute force, dictionary)
- Network-based attacks (eavesdropping, replay)
- Human-targeted attacks (phishing)

✓ This highlights why **strong passwords + secure protocols + multi-factor authentication** are necessary for robust authentication security.