Question 1

- List and briefly define categories of passive and active network security attacks.
Question 1

❖ List and briefly define categories of passive and active network security attacks.

➢ Passive attacks have to do with eavesdropping or monitoring transmissions. Electronic mail, file transfers, and client/server exchanges are examples of transmissions that can be monitored.

➢ Active attacks include the modification of transmitted data and attempts to gain unauthorized access to computer systems.
  ■ E.g., masquerade, replay, modification of messages, and denial of service
Question 2

❖ What are the principal ingredients of a public-key cryptosystem?
Question 2

❖ What are the principal ingredients of a public-key cryptosystem?

➢ **Plaintext:** This is the readable message or data that is fed into the algorithm as input.

➢ **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext.

➢ **Public and private keys:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input.

➢ **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.

➢ **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.
Question 3

- What is the difference between a private key and a secret key?
Question 3

What is the difference between a private key and a secret key?

- The key used in conventional encryption is typically referred to as a secret key.
- The two keys used for public-key encryption are referred to as the public key and the private key.
Question 4

- What is a digital signature?
Question 4

❖ What is a digital signature?

➢ A **digital signature** is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature.

➢ The signature is formed by taking the hash of the message and encrypting the message with the creator's **private key**

➢ The signature **guarantees** the **source** and **integrity** of the message
Question 5

❖ What is a public-key certificate?
Question 5

❖ What is a public-key certificate?

➢ A public-key certificate consists of a public key plus a User ID of the key owner, with the whole block signed by a trusted third party.

➢ Typically, the third party is a certificate authority (CA) that is trusted by the user community, such as a government agency or a financial institution.
Question 6

- In general terms, what are four means of authenticating a user’s identity?
In general terms, what are four means of authenticating a user’s identity?

- **Something the individual knows**: Examples includes a password, a personal identification number (PIN), or answers to a prearranged set of questions.
- **Something the individual possesses**: Examples include electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token.
- **Something the individual is** (static biometrics): Examples include recognition by fingerprint, retina, and face.
- **Something the individual does** (dynamic biometrics): Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.
Question 7

- List and briefly describe the principal threats to the secrecy of passwords.
Question 7

❖ List and briefly describe the principal threats to the secrecy of passwords.

➢ **Offline dictionary attack**: Typically, strong access controls are used to protect the system's password file. However, experience shows that determined hackers can frequently bypass such controls and gain access to the file. The attacker obtains the system password file and compares the password hashes against hashes of commonly used passwords. If a match is found, the attacker can gain access by that ID/password combination.

➢ **Specific account attack**: The attacker targets a specific account and submits password guesses until the correct password is discovered.

➢ **Popular password attack**: A variation of the preceding attack is to use a popular password and try it against a wide range of user IDs. A user's tendency is to choose a password that is easily remembered; this unfortunately makes the password easy to guess.

➢ **Password guessing against single user**: The attacker attempts to gain knowledge about the account holder and system password policies and uses that knowledge to guess the password.
Question 7

❖ List and briefly describe the principal threats to the secrecy of passwords.

➢ **Workstation hijacking**: The attacker waits until a logged-in workstation is unattended.
➢ **Exploiting user mistakes**: If the system assigns a password, then the user is more likely to write it down because it is difficult to remember. This situation creates the potential for an adversary to read the written password. A user may intentionally share a password, to enable a colleague to share files, for example. Also, attackers are frequently successful in obtaining passwords by using social engineering tactics that trick the user or an account manager into revealing a password.
➢ **Exploiting multiple password use**: Attacks can also become much more effective or damaging if different network devices share the same or a similar password for a given user.
➢ **Electronic monitoring**: If a password is communicated across a network to log on to a remote system, it is vulnerable to eavesdropping. Simple encryption will not fix this problem, because the encrypted password is, in effect, the password and can be observed and reused by an adversary.
Question 8

- What are two common techniques used to protect a password file?
Question 8

What are two common techniques used to protect a password file?

- One technique is to restrict access to the password file using standard access control measures. Another technique is to force users to select passwords that are difficult to guess.
Problem 1

Consider the following general code for allowing access to a resource:

```c
DWORD dwRet = IsAccessAllowed(...);
if (dwRet == ERROR_ACCESS_DENIED) {
    // Security check failed.
    // Inform user that access is denied.
} else {
    // Security check OK.
}
```

a) Explain the security flaw in this program.
b) Rewrite the code to avoid the flaw
DWORD dwRet = IsAccessAllowed(...);

if (dwRet == ERROR_ACCESS_DENIED) {
    // Security check failed.
    // Inform user that access is denied.
} else {
    // Security check OK.

→ What happens if IsAccessAllowed fails? For example, what happens if the system runs out of memory, or object handles, when this function is called? The user can execute the privileged task because the function might return an error such as ERROR NOT ENOUGH MEMORY.
Problem 1

    DWORD dwRet = IsAccessAllowed(...);

    if (dwRet == NO_ERROR) {
        // Secure check OK.
        // Perform task.
    } else {
        // Security check failed.
        // Inform user that access is denied.
    }

→ In this case, if the call to IsAccessAllowed fails for any reason, the user is denied access to the privileged operation
Problem 2

Suppose that someone suggests the following way to confirm that the two of you are both in possession of the same secret key. You create a random bit string the length of the key, XOR it with the key, and send the result over the channel. Your partner XORs the incoming block with the key (which should be the same as your key) and sends it back. You check, and if what you receive is your original random string, you have verified that your partner has the same secret key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme?
Problem 2

Suppose that someone suggests the following way to confirm that the two of you are both in possession of the same secret key. You create a random bit string the length of the key, XOR it with the key, and send the result over the channel. Your partner XORs the incoming block with the key (which should be the same as your key) and sends it back. You check, and if what you receive is your original random string, you have verified that your partner has the same secret key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme?

→ **Yes.** The eavesdropper is left with two strings, one sent in each direction, and their XOR is the secret key.

Alice--->Bob : RandomBitString(XOR)Key
Bob -->Alice : RandomBitString
Attacker : RandomBitString(XOR)Key (XOR) RandomBitString = key
Problem 3

In this problem we will compare the security services that are provided by digital signatures (DS) and message authentication codes (MAC). We assume that Oscar is able to observe all messages sent from Alice to Bob and vice versa. Oscar has no knowledge of any keys but the public one in case of DS. State whether and how (i) DS and (ii) MAC protect against each attack. The value $\text{auth}(x)$ is computed with a DS or a MAC algorithm, respectively.
Problem 3

a) (Message integrity) Alice sends a message $x = \text{“Transfer$1000 to Mark”}$ in the clear and also sends $\text{auth}(x)$ to Bob. Oscar intercepts the message and replaces “Mark” with “Oscar.” Will Bob detect this?
Problem 3

a) (Message integrity) Alice sends a message $x = \text{"Transfer $1000 to Mark"}$ in the clear and also sends $\text{auth}(x)$ to Bob. Oscar intercepts the message and replaces "Mark" with "Oscar." Will Bob detect this?

→ Will be detected with both (i) DS and (ii) MAC
Problem 3

b) (Replay) Alice sends a message $x = \text{“Transfer $1000 to Oscar”}$ in the clear and also sends $\text{auth}(x)$ to Bob. Oscar observes the message and signature and sends them 100 times to Bob. Will Bob detect this?
Problem 3

b) (Replay) Alice sends a message $x = \text{"Transfer $1000 to Oscar"}$ in the clear and also sends $\text{auth}(x)$ to Bob. Oscar observes the message and signature and sends them 100 times to Bob. Will Bob detect this?

→ Won’t be detected by either

Use timestamps!!!
Problem 3

c) (Sender authentication with cheating third party) Oscar claims that he sent some message \( x \) with a valid \( \text{auth}(x) \) to Bob but Alice claims the same. Can Bob clear the question in either case?
Problem 3

c) (Sender authentication with cheating third party) Oscar claims that he sent some message $x$ with a valid $\text{auth}(x)$ to Bob but Alice claims the same. Can Bob clear the question in either case?

(i) DS: Bob simply has to verify the message with the public key from both. Obviously, only Alice’s public key results in a successful verification.

(ii) MAC: Bob has to challenge both to reveal their secret key to him (which he knows anyway). Only Bob can do that.
Problem 3

d) (Authentication with Bob cheating) Bob claims that he received a message $x$ with a valid signature $\text{auth}(x)$ from Alice (e.g., “Transfer $1000$ from Alice to Bob”) but Alice claims she has never sent it. Can Alice clear this question in either case?
Problem 3

d) (Authentication with Bob cheating) Bob claims that he received a message $x$ with a valid signature $\text{auth}(x)$ from Alice (e.g., “Transfer $1000 from Alice to Bob”) but Alice claims she has never sent it. Can Alice clear this question in either case?

(i) DS: Alice has to force Bob to prove his claim by sending her a copy of the message in question with the signature. Then Alice can show that message and signature can be verified with Bob’s public key. Bob must have generated the message.

(ii) MAC: No, Bob can claim that Alice generated this message.
Problem 4

Explain the suitability or unsuitability of the following passwords:

A. YK 334
B. mfmitm
C. Natalie1
D. Washington
E. Aristotle
F. tv9stove
G. 12345678
H. dribgib
Problem 4

Explain the suitability or unsuitability of the following passwords:

A. YK 334 , (seems like a license plate) → unsuitable/easily guessable
B. mfmitm → (for “my favorite movie is tender mercies) → Suitable
C. Natalie1 → Unsuitable/easily guessable
D. Washington → Unsuitable/easily guessable
E. Aristotle → Unsuitable/easily guessable
F. tv9stove → Suitable (* include special chars and numbers)
G. 12345678 → Most common password. Exists in the first lines of any password dictionary
H. dribgib → Unsuitable Why?
Problem 4

Explain the suitability or unsuitability of the following passwords:

A. YK 334 , (seems like a license plate) → unsuitable/easily guessable
B. mfmitm → (for “my favorite movie is tender mercies) → Suitable
C. Natalie1 → Unsuitable/easily guessable
D. Washington → Unsuitable/easily guessable
E. Aristotle → Unsuitable/easily guessable
F. tv9stove → Suitable (* include special chars and numbers)
G. 12345678 → Most common password. Exists in the first lines of any password dictionary
H. dribgib → Unsuitable Why?

Passwords **should** include **numbers** and **special chars**
Problem 5

It was stated that the inclusion of the salt in the UNIX password scheme increases the difficulty of guessing by a factor of 4096*. But the salt is stored in plaintext in the same entry as the corresponding ciphertext password. Therefore, those two characters are known to the attacker and need not be guessed. Why is it asserted that the salt increases security?

*Early Unix implementations limited passwords to eight characters and used a 12-bit salt, which allowed for 4,096 possible salt values
Problem 5

It was stated that the inclusion of the salt in the UNIX password scheme increases the difficulty of guessing by a factor of 4096. But the salt is stored in plaintext in the same entry as the corresponding ciphertext password. Therefore, those two characters are known to the attacker and need not be guessed.

Why is it asserted that the salt increases security?

→ Without the salt, the attacker can guess a password and encrypt it. If ANY of the users on a system use that password, then there will be a match. With the salt, the attacker must guess a password and then encrypt it once for each user, using the particular salt for each user.
Problem 6

Wouldn’t it be possible to thwart completely all password crackers by dramatically increasing the salt size to, say, 24 or 48 bits?
Problem 6

Wouldn’t it be possible to thwart completely all password crackers by dramatically increasing the salt size to, say, 24 or 48 bits?

→ It depends on the size of the user population, not the size of the salt, since the attacker presumably has access to the salt for each user. The benefit of larger salts is that the larger the salt, the less likely it is that two users will have the same salt. If multiple users have the same salt, then the attacker can do one encryption per password guess to test all of those users.