1. **Consider a system that provides authentication services for critical systems, applications, and devices. Give examples of confidentiality, integrity, and availability requirements associated with the system. In each case, indicate the degree of importance of the requirement. (10 points)**

Ans:

Confidentiality, integrity and availability, also known as the CIA triad, is a model designed to guide policies for information security within an organization. The model is also sometimes referred to as the AIC triad (availability, integrity and confidentiality) to avoid confusion with the Central Intelligence Agency. Although elements of the triad are three of the most foundational and crucial cybersecurity needs, experts believe the CIA triad needs an upgrade to stay effective.

High:

1. **The loss have a severe or catastrophic adverse effect on organizational operations**
2. **Organization cannot do one or more of its primary functions**

Moderate:

1. **The loss could have a serious adverse effect on organizational operations**
2. **Significant degradation on organization Moderate functionality**

Low:

1. **The loss could have a limited effect on organizational operations**
2. **Organization can still perform its normal functions**

Confidentiality:

High:	**Student grade should be available only to students, parents, and relevant employees**
Moderate	**Students enrolment**
Low	**Directory info (list of departments, faculties, and students)**

Integrity:

High:	**Patient information stored in a database – inaccurate information could result in serious harm or death to a patient**
Moderate	**A entertainment Web site that offers a forum. A user/hacker falsify some info.**
Low	**Anonymous online poll with weak authentication**

Availability:

High:	**Authentication provider services.**
Moderate	**University's website**
Low	**An online telephone directory**

2. **Consider a desktop publishing system used to produce documents for various organizations. (15 points)**

(a) Give an example of a type of publication for which confidentiality of the stored data is the most important requirement.

Ans:

Confidentiality: An example of a type of publication for which confidentiality of the stored data is the most important requirement could be a government intelligence report. In this scenario, the content of the document may contain highly classified or sensitive information that, if exposed, could compromise national security or ongoing investigations. Maintaining the utmost confidentiality is crucial.

(b) Give an example of a type of publication in which data integrity is the most important requirement.

Ans:

Data Integrity: For a type of publication where data integrity is the most important requirement, consider medical records or patient healthcare documents. These documents contain critical patient information, and any form of data corruption or unauthorized alterations could result in incorrect diagnoses, treatments, or potentially life-threatening consequences. Ensuring the integrity of healthcare data is paramount.

(c) Give an example in which system availability is the most important requirement.

Ans:

System Availability: In the context of desktop publishing, news websites or online news portals are an example where system availability is the most important requirement. These websites must be available 24/7 to provide real-time news updates to a global audience. Downtime could lead to a loss of readership, credibility, and revenue. Therefore, ensuring the continuous availability of the publishing system is of utmost importance in this case.

3. **Alice was told to design a scheme to prevent messages from being modified by an attacker. Alice decides to append to each message a hash (message digest) of that message. Why doesn't this solve the problem? (10 points)**

Ans:

Appending a hash (message digest) of each message to the message itself is a common approach to provide data integrity and detect message tampering.

However, it doesn't fully solve the problem of message modification by an attacker, and there are several reasons for this:

a. Predictable Hashing Algorithm: If the attacker knows the hashing algorithm being used, they can calculate the hash of the original message and then modify the message and recalculate the hash. If the recalculated hash matches the appended hash in the modified message, the recipient will not detect any tampering. This is known as a "length extension attack."
b. No Secret Key: Appending a simple hash doesn't involve a secret key, so it doesn't provide authentication. An attacker can still modify the message and recompute the hash without the need for any secret information, making it susceptible to forgery.
c. Lack of Encryption: Message integrity mechanisms like hashing protect against unintentional modifications but don't protect against active attackers who can intercept and modify the message. They don't provide confidentiality. Even if the hash detects tampering, the attacker may still know the content of the message.
d. Collision Attacks: In some cases, an attacker may be able to find two different messages with the same hash (collision). If they can do this, they can swap the original message with a different one that has the same hash, and the recipient will not detect the change.

To solve these issues and prevent message modification by an attacker, a more robust approach is needed. This typically involves using digital signatures, which combine a message digest with a secret key to provide both data integrity and authentication. Digital signatures ensure that the message has not been altered and that it was indeed sent by the legitimate sender.

4. What RC4 key value will leave S unchanged during initialization? That is, after the initial permutation of S, the entries of S will be equal to the values from 0 through 255 in ascending order. (10 points)

Ans:

Use a key of length 255 bytes. The first two bytes are zero; that is $K[0] = K[1] = 0$. Thereafter, we have: $K[2] = 255, K[3] = 254, \ldots, K[255] = 2$.

5. With the ECB mode, if there is an error in a block of the transmitted ciphertext, only the corresponding plaintext block is affected. However, in the CBC mode, this error propagates. For example, an error in the transmitted C_1 obviously corrupts P_1 and P_2. Are any blocks beyond P_2 affected? (10 points)

Ans:

In the CBC (Cipher Block Chaining) mode of operation, if there is an error in a

transmitted ciphertext block, the error does not propagate indefinitely to all subsequent plaintext blocks. Rather, the error is localized and only affects the corresponding plaintext block P_i and the subsequent plaintext block P_{i+1}.

Here's how error propagation works in CBC mode:

An error in the transmitted ciphertext block C_i affects the decryption of that block, resulting in an incorrect plaintext block P_i .

However, because the decryption of each block relies on the ciphertext of the previous block C_{i-1}, this error also affects the decryption of C_{i-1}, which, in turn, affects the decryption of P_{i-1}. This propagates the error to the preceding plaintext block P_{i-1}.

The error is localized to these two adjacent blocks, and it does not propagate further. Any subsequent blocks (P_{i+2}, P_{i+3}, and so on) are not directly affected by the error in C_i or P_i.

So, in summary, an error in the CBC mode propagates to the current plaintext block P_i and the next plaintext block P_{i+1}, but it does not affect blocks beyond P_{i+1}. This error localization is one of the advantages of CBC mode as it contains errors and prevents them from affecting the entire decrypted message.

6. **Is it possible to perform encryption operations in parallel on multiple blocks of plaintext in CBC mode? How about decryption? (15 points)**

Ans:

In CBC (Cipher Block Chaining) mode, encryption operations for multiple blocks of plaintext cannot be performed in parallel due to the dependency on the previous ciphertext block.

Encryption in CBC Mode:

In CBC mode, each plaintext block P_i is first XORed with the previous ciphertext block C_{i-1} before encryption. This means that you need the ciphertext of the previous block to encrypt the current block. Consequently, encryption of each block must be sequential, as the output of one block depends on the encryption of the previous block.

Decryption in CBC Mode:

Decryption in CBC mode is more forgiving and can be parallelized to some extent. This is because, during decryption, you have the ciphertext blocks and the decryption of one block does not depend on the previous ciphertext block, only on the previous ciphertext block and the current ciphertext block. So, if you have all the necessary ciphertext blocks, you can perform decryption operations on multiple blocks in parallel.

However, note that even in decryption, there may still be some sequential dependency because you need the result of decrypting the previous block to XOR

with the current ciphertext block to recover the plaintext block. But within a group of blocks that are not dependent on each other, parallel decryption is possible.

In summary, parallel encryption in CBC mode is not possible due to the sequential dependency on the previous ciphertext block. Decryption can be parallelized to some extent as long as you have all the necessary ciphertext blocks for the blocks you want to decrypt in parallel.

7. Suppose $H(m)$ is a collision-resistant hash function that maps a message of arbitrary bit length into an n-bit hash value. Is it true that, for all messages x, x' with $x \neq x'$, we have $H(x) \neq H(x')$? Explain your answer. (10 points)

Ans:

Given that $H(m)$ is a collision-resistant hash function, it does not necessarily mean that for all messages x and x' where $x \neq x'$, the hash values $H(x)$ and $H(x')$ will be different. This is a key point that relates to the nature of hash functions and the possibility of collision resistance.

The property of collision resistance means that it should be computationally infeasible to find two different messages that produce the same hash value. However, it does not imply that different messages will always produce different hash values.

In other words, collision resistance guarantees that it is hard to find two messages x and x' such that $H(x) = H(x')$ when $x \neq x'$. However, it does not guarantee that for all x and x' where $x \neq x'$, $H(x) \neq H(x')$.

Therefore, it is possible that for some specific messages x and x' with $x \neq x'$, the hash values $H(x)$ and $H(x')$ could be the same, even if the hash function $H(m)$ is collision-resistant.

The fundamental idea behind collision resistance is that it should be computationally infeasible to find these collisions, even if they might exist. The security of hash functions is based on the assumption that finding such collisions requires an impractical amount of computational resources.

8. It is possible to use a hash function to construct a block cipher with a structure similar to DES. Because a hash function is one way and a block cipher must be reversible (to decrypt), how is it possible? (10 points)

Ans:

Constructing a block cipher from a one-way hash function is not straightforward because a block cipher needs to be reversible, and a one-way hash function, by definition, is not designed to be reversible. However, it is possible to create a construction that behaves like a block cipher using a one-way hash function within a structure known as a cryptographic primitive.

One common approach to achieving this is to use a method called a "Merkle-Damgard construction." This construction essentially involves repeatedly applying the one-way hash function to a series of blocks, often combining them in some way to make the transformation reversible.

Here is a simplified example of how this might work:

Key Expansion: You can use a portion of the encryption key to create a series of derived keys. These derived keys are used to modify the behavior of the hash function.

Initialization Vector (IV): A random IV is generated for each encryption operation. The IV is treated as the initial "plaintext" block in the hash function.

Hash Chain: Starting with the IV, you apply the one-way hash function to the current block and XOR the result with the next block of data (plaintext or ciphertext). The output becomes the new "plaintext" block, and you repeat this process for the entire message.

Finalization: You might have some final processing steps to ensure the output is indistinguishable from random data and suitable for use as ciphertext.

9. **In an RSA system, the public key of a given user is $e = 3, n = 667$. What is the private key for this user? (10 points)**

Ans:

Let $p = 23$ and $q = 29$, then $n = 23 * 29 = 667$

$\phi(n) = (p - 1) * (q - 1) = 22 * 28 = 616$

given user is $e = 3$ $s.t. gcd(e, \phi(n)) = 1$

Calculate d from $ed = 1 \, mod \, \phi(n)$

$d = 411$ because $411 * 3 = 1233 \equiv 1 \, mod \, 616$