HW4- SECURE CODING

1)

The one-time pad encryption technique is considered theoretically unbreakable if the key used is truly random and never reused. However, if the same key is employed to encrypt multiple plaintext messages, it becomes vulnerable to a known plaintext attack. In a known plaintext attack, the attacker has access to both the encrypted message (ciphertext) and the original unencrypted message (plaintext) corresponding to that ciphertext. By having this pair of ciphertext and plaintext, the attacker can exploit the properties of the one-time pad encryption to recover the key. Suppose there are two plaintext messages, P1 and P2, encrypted using the same key (K) to produce ciphertexts C1 and C2, respectively. If the attacker knows both C1 and C2, as well as the plaintext P1 corresponding to C1, they can perform the following steps:

Use the known plaintext P1 and its corresponding ciphertext C1 to derive the key K by applying the one-time pad decryption process (XORing P1 and C1).

Similarly, compute P2 = C2 $\oplus$ K and compute P1 = C1 $\oplus$ K.
Through this way, the attacker can recover the key K and he can decrypt any message that was encrypted with the same key which includes decryption of P1.

2)

a) Alice and Bob have a method for secure communication using Message Authentication Codes (MACs). They establish a shared secret key to compute the MAC for messages. Upon receiving a message, Bob recalculates the MAC using the agreed secret key. If the message has been tampered with, the recalculated MAC will not match the original one obtained by Alice, signaling potential alterations.

b) Alice and Bob can attain a comparable level of security using a Hash-based message authentication code. They select the code employing a hash function known only to them. When Bob receives the message, he recalculates the Hash-based message authentication code using the agreed hash function. If the recalculated authentication code differs from the received one, Bob can confirm that the message has been tampered with.

c) In private key cryptography, Alice can ensure message integrity by signing it with a digital signature created using her private key. Bob, who possesses Alice's public key from her certificate, can then verify the signature. If the message is altered, the signature that Bob generates using Alice's public key will not match the one originally attached to the message, providing a clear indication of tampering.

3) The vulnerability lies in the simplicity of the hashing method, allowing attackers to exploit patterns between messages and their hashes. By modifying both the message and its associated hash accordingly, attackers can conceal tampering, making it challenging for receivers to detect alterations. To avoid this risk, using advanced cryptographic techniques like salted hashes or cryptographic hash functions resistant to pattern recognition attacks is important. Additionally, utilizing methods such as digital signatures or asymmetric encryption enhances security by ensuring message integrity and thwarting tampering attempts effectively.

4)

- The hash function should be quick to compute, ensuring minimal computational overhead.
- Given the same input, the hash function should consistently produce the same output, ensuring predictability and reliability. Moreover, different inputs should give different hash values to avoid collisions.
- Even a minor alteration in the input should result in a completely different hash value, thereby dispersing the impact of changes throughout the output.
- The hash function should not possess any inherent vulnerabilities or weaknesses, ensuring that the computed hash value remains secure and resistant to attacks or exploits.

5) The suggested Hash Function has a deterministic vulnerability due to the possibility of generating the same output for different inputs using a XOR operation. This introduces collision vulnerabilities, compromising the hash function's ability to generate a unique and distinct representation of messages. So, the function is not a good hash function as it fails to provide the desired level of uniqueness and reliability in hash value generation.