**WQD7010**
**Network & Security (Sem 2, Session 2023/2024)**

# Tutorial 2
## Asymmetric Encryption and Message Authentication

1. Suppose we have a hashing algorithm that generates a 32-bit digest by fusing together two distinct 16-bit sub-functions: XOR and RXOR. These sub-functions are described in lecture slides as "two simple hash functions."

   a) Imagine a scenario where an attacker tampers with the input data, flipping an odd number of bits. Evaluate the effectiveness of this hashing algorithm in detecting such tampering attempts. Provide a rigorous justification for your conclusion, considering the properties of XOR and RXOR. (0.5)

   b) Now, let's explore the algorithm's behavior when an even number of bits are maliciously altered. Determine whether the hash function will consistently identify all such modifications. If there are specific tampering patterns that can evade detection, meticulously describe and characterize those scenarios. (1)

   c) From a security standpoint, assess the robustness of this 32-bit hashing algorithm when employed for authentication purposes. Analyze its resilience against potential attacks, considering factors such as collision resistance and preimage resistance. Provide a thoughtful commentary on whether this algorithm offers sufficient protection for verifying the integrity and authenticity of data. Support your analysis with well-reasoned arguments. (1)

2. Analyze the hash functions below that operate on input data represented as a series of base-10 integers, D = ($m_1$, $m_2$, ..., $m_t$). For each hash function, determine which of the requirements for a cryptographic hash function listed in slides 54 (requirement of hash function) are satisfied. Explain your answers.

   a) h = (Σ(i=1 to t) $m_i$) mod n, for some predefined value n. (0.5)

   b) h = (Σ(i=1 to t) ($m_i$)^2) mod n (0.5)

   c) Calculate the hash value from part (b) for the message D = (237, 632, 913, 423, 349) and n = 757. (0.5)

3. In an RSA cryptosystem, you are given the following parameters:

   p = 17, q = 19, and a specific integer used for encryption

   a) Determine the integer utilized for decryption. (0.5)

   b) Enumerate the components of the public and private key pairs. (0.5)

   You need to encrypt the plaintext message, M = RSA.

   c) Transform this message into its corresponding numerical representation based on the ASCII table. (0.5)

To encrypt this message, you have the option to either amalgamate the numerical values or partition them into blocks of 2 digits (Hint: ensure that the condition M < n is satisfied).

d) Apply the encryption and decryption processes to the message RSA. (1)

e) If the ciphertext's decimal representation (not its character equivalent) is 2190236, perform the decryption process on this value and transform the result into its corresponding message format according to the ASCII table. (0.5)

4. Consider a Diffie–Hellman scheme with a common prime q = 353 and a primitive root a = 3.

a) If user A has public key YA = 40, what is A's private key XA? (0.5)

b) If user B has public key YB = 248, what is the shared secret key K? (0.5)

c) Eve intercepts the communication between User A and User B and chooses her own private key XE = 201. What is Eve's public key YE, and what shared secret key K_BE will User B compute using Eve's public key YE, assuming User B's private key XB = 156? (0.5)

d) What shared secret key K_EB will Eve compute using User B's public key YB, and what shared secret key K_AE will User A compute using Eve's public key YE? (0.5)

e) After the man-in-the-middle attack, what are the shared secret keys between User A and Eve (K_AE) and between User B and Eve (K_BE)? Explain how Eve can intercept, decrypt, and read the messages sent between User A and User B without their knowledge. (0.5)

f) If User A sends a message encrypted with K_AE, describe the steps Eve can take to modify the message and send it to User B, making User B believe it came securely from User A. How does the lack of authentication in the Diffie-Hellman key exchange protocol make it vulnerable to man-in-the-middle attacks, and what additional security measures can be implemented to mitigate this risk? (1)

5. Consider a variant of CMAC that XORs the second key K1 after applying the final block cipher encryption, rather than before. This variant can be expressed as: VMAC(K, M) = CBC(K, M) $\oplus$ $K_1$ where the message M is an integer multiple of the block size.

Suppose an adversary can obtain the MACs for three chosen messages:

Message 0 = $0^n$ (n is the cipher block size)<br>
Message 1 = $1^n$<br>
Message (1 || 0) which is message 1 concatenated with message 0

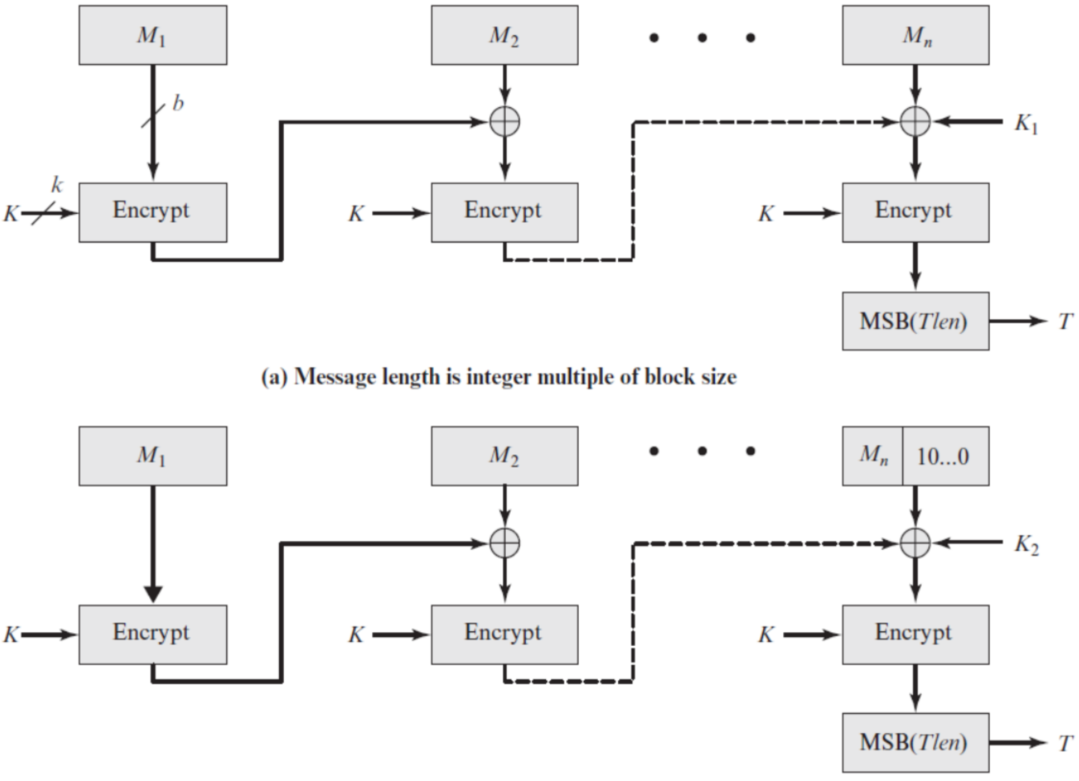
Let the corresponding MACs be:

$T_0$ = CBC(K, 0) $\oplus$ $K_1$

$T_1$ = CBC(K, 1) $\oplus$ $K_1$

$T_2$ = CBC(K, [CBC(K, 1)]) $\oplus$ $K_1$

Show that the adversary can now compute the correct MAC for a new message 0 || ($T_0$ ⊕ $T_1$) without making any additional queries to the MAC oracle.What modification you have to made from the original CMAC (Figure 1) diagram? (1)



(a) Message length is integer multiple of block size

(b) Message length is not integer multiple of block size

Figure 1: CMAC