Fuzail and Peyrin
Summer 2022

CS 161
Computer Security

Discussion

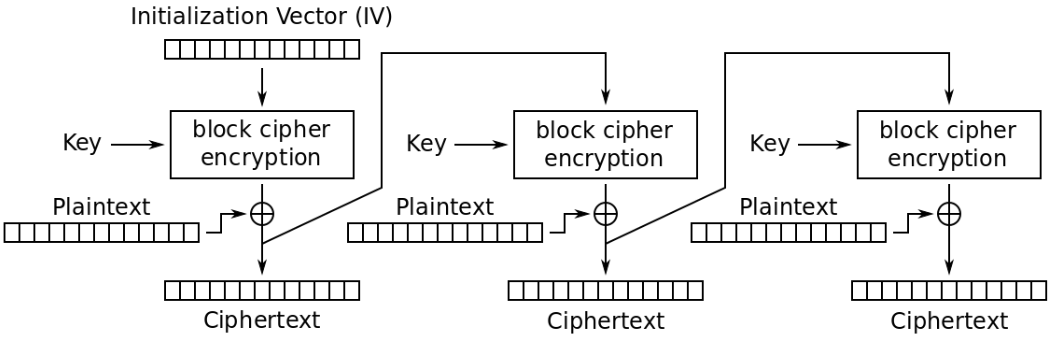
**Question 1** ***Block ciphers*** ()

Consider the Cipher feedback (CFB) mode, whose encryption is given as follows:

$$C_i = \begin{cases} \text{IV}, i = 0 \\ E_K(C_{i-1}) \oplus P_i, \text{otherwise} \end{cases}$$

Q1.1 Draw the encryption diagram for CFB mode.

**Solution:**



Cipher Feedback (CFB) mode encryption

Q1.2 What is the decryption formula for CFB mode?

**Solution:**

$$P_i = E_k(C_{i-1}) \oplus C_i$$

Q1.3 Select the true statements about CFB mode:

□ Encryption can be paralellized

■ Decryption can be paralellized

■ The scheme is IND-CPA secure

**Solution:** Encryption is not parallelizable because the encryption of the $n'th$ block of plaintext is dependent on the $n - 1'th$ ciphertext. Decryption is parallelizable because the decryption of the $n'th$ block of ciphertext is dependent on the $n - 1'th$ ciphertext. The scheme is IND-CPA secure because an adversar cannot provide two messages of equal length such that they gain a non-negligible advantage in the IND-CPA game, as long as the IV is not reused. Note that if the IV is reused, the scheme would be deterministic.

Q1.4 What happens if two messages are encrypted with the same key and nonce? What can the attacker learn about the two messages just by looking at their ciphertexts?

**Solution:** If the IV is reused in AES-CFB, the attacker can determine if two messages have identical prefix, up to but not including the first block containing the difference. This is because the $n$th plaintext block affects the input to $n$th input to the block cipher, and any difference in the plaintext block results in a completely different block cipher output.

When we use non-repeating IVs for CFB-mode, even if we encrypt the same message multiple times, CFB-mode will generate distinct and random-looking ciphertexts each time.

Q1.5 If an attacker recovers the IV used for a given encryption, but not the key, will they be able to decrypt a ciphertext encrypted with the recovered IV and a secret key?

**Solution:**

No, the secrecy of the IV does not affect the security of the encryption scheme, as the IV is passed as part of the output of an encryption. The only condition is that the IV must not be reused in order for the given scheme to be secure.

**Question 2** ***Crytographic Hashes*** ()

For each of the given functions $H$ below, determine if it is one-way or not, and if it is collision-resistant or not.

Q2.1 $\mathsf{H}(x) = x^2$

○ (A) One way

○ (B) Collision resistant

○ (C) Both

● (D) Neither

**Solution:** This function is not collision-resistant. Consider $H(1) = H(-1) = 1$.

This function is not one-way because given $H(x)$, we can calculate $\sqrt(H(x)) = \sqrt(x^2) = x$.

Q2.2 For this part you have access to a SHA-256 hash function. The notation $[x : y]$ refers to a slice of bytes $x$ to $y - 1$.

$\mathsf{H}(x) = \text{SHA-256}(x[0 : \mathsf{len}(x) - 1])$

● (G) One way

○ (H) Collision resistant

○ (I) Both

○ (J) Neither

**Solution:** This function is not collision-resistant. Consider the values of $x =$ "*abc*" and $x =$ "*abd*". As defined by the hash function, we take the first len(x) - 1 bytes and pass that into the SHA-256 hash function. Therefore both vales of x would become SHA-256("ab") and have the same hash value.

The function is one way because SHA-3 is one way and knowing the output of $H(x)$ does not tell us about the input x.

Q2.3 $\mathsf{H}(x) = x^3$

○ (A) One way

● (B) Collision resistant

○ (C) Both

○ (D) Neither

**Solution:** This function is collision-resistant because the function $x^3$ is monotonically increasing and no two values of $x$ will have the same output.

This function is not one-way similar to the reasoning in part 1. Given $H(x)$, we can calculate $\sqrt[3]{(}H(x)) = \sqrt[3]{(}x^3) = x$.

**Question 3** ***Confidentiality and integrity*** ()

Alice and Bob want to communicate with confidentiality and integrity. They have:

- Symmetric encryption.
  - Encryption: $\mathsf{Enc}(\mathsf{K}, \mathsf{m})$.
  - Decryption: $\mathsf{Dec}(\mathsf{K}, \mathsf{c})$.
- Cryptographic hash function: $\mathsf{Hash}(\mathsf{m})$.
- MAC: $\mathsf{MAC}(\mathsf{K}, \mathsf{m})$.

They share a symmetric key $\mathsf{K}$ and know each other's public key.

---

We assume these cryptographic tools do not interfere with each other when used in combination; *i.e.*, we can safely use the same key for encryption and MAC.

| Alice sends to Bob |
|---|
| 1. $\mathsf{c} = \mathsf{Hash}(\mathsf{Enc}(\mathsf{K}, \mathsf{m}))$ |
| 2. $\mathsf{c} = \mathsf{c}_1, \mathsf{c}_2$ : where $\mathsf{c}_1 = \mathsf{Enc}(\mathsf{K}, \mathsf{m})$ and $\mathsf{c}_2 = \mathsf{Hash}(\mathsf{Enc}(\mathsf{K}, \mathsf{m}))$ |
| 3. $\mathsf{c} = \mathsf{c}_1, \mathsf{c}_2$ : where $\mathsf{c}_1 = \mathsf{Enc}(\mathsf{K}, \mathsf{m})$ and $\mathsf{c}_2 = \mathsf{MAC}(\mathsf{K}, \mathsf{m})$ |
| 4. $\mathsf{c} = \mathsf{c}_1, \mathsf{c}_2$ : where $\mathsf{c}_1 = \mathsf{Enc}(\mathsf{K}, \mathsf{m})$ and $\mathsf{c}_2 = \mathsf{MAC}(\mathsf{K}, \mathsf{Enc}(\mathsf{K}, \mathsf{m}))$ |

Q3.1 Which ones of them can Bob decrypt?

□ 1 □ 2 □ 3 □ 4

**Solution:** Bob cannot decrypt Scheme 1 because he cannot invert $\mathsf{Hash}$.

***In sum:*** 2-4

Q3.2 Consider an eavesdropper Eve, who can see the communication between Alice and Bob.

Which schemes, of those decryptable in (a), also provide *confidentiality* against Eve?

□ 1 □ 2 □ 3 □ 4

**Solution:** Scheme 3 does not provide confidentiality because the MAC is sent in plaintext. For the same message, the MAC is the same, thus leaky.

***In sum:*** 2, 4

Q3.3 Consider a man-in-the-middle Mallory, who can eavesdrop and modify the communication between Alice and Bob.

Which schemes, of those decryptable in (a), provide *integrity* against Mallory? *i.e.*, Bob can detect any tampering with the message?

□ 1 □ 2 □ 3 □ 4

**Solution:** Scheme 2 does not provide integrity as Mallory can forge a message by sending Bob $(\mathsf{c}', \mathsf{Hash}(\mathsf{c}'))$.

***In sum:*** 3, 4

Q3.4 Many of the schemes above are insecure against a *replay attack.*

If Alice and Bob use these schemes to send many messages, and Mallory remembers an encrypted message that Alice sent to Bob, some time later, Mallory can send the exact same encrypted message to Bob, and Bob will believe that Alice sent the message *again.*

How to modify those schemes with confidentiality & integrity to prevent replay attack?

◇ The scheme providing confidentiality & integrity is Scheme □.

The modification is:

**Solution:** Add a non-repeating nonce or timestamp in the MAC.

***In sum:*** 4, we replace message $\mathsf{m}$ with $\mathsf{timestamp} \parallel \mathsf{m}$.