Cryptography

Question 1 ***Pairing an IOT Device*** **(28 min)**

Alice wishes to pair her new IoT device and her laptop by having them exchange a symmetric key k. The devices will later use k to encrypt plaintext messages and send the ciphertexts to each other. Assume that there is a MITM on the network between the IoT device and the laptop. To defend against the MITM, Alice is considering the security of different pairing protocols. For each scenario below, select all true statements.

The "old key" refers to a symmetric key from some previous pairing. $\mathsf{Enc}(\mathsf{PK}; m)$ refers to public-key encryption of m with PK. Each subpart is independent.

Q1.1 The IoT device chooses k randomly and sends it to the laptop unencrypted over the network.

■ (A) MITM can decrypt the messages from the IoT device to the laptop

■ (B) MITM can decrypt the messages from the laptop to the IoT device

■ (C) At least one of the devices could accept an attacker's key that was not an old key

■ (D) MITM can make at least one of the devices to accept an old key

☐ (E) None of the above

☐ (F) ——

Solution: All because there is no security here.

Q1.2 The IoT device sends a message to the laptop asking for its public key PK. The laptop sends PK to the IoT device. The IoT device chooses k randomly and sends $\mathsf{Enc}(\mathsf{PK}; k)$ to the laptop.

■ (G) MITM can decrypt the messages from the IoT device to the laptop

■ (H) MITM can decrypt the messages from the laptop to the IoT device

■ (I) At least one of the devices could accept an attacker's key that was not an old key

■ (J) MITM can make at least one of the devices to accept an old key

☐ (K) None of the above

☐ (L) ——

Solution: MITM can supply its own PK to the IoT device so there is no security here.

Q1.3 Alice manually enters the publicly-known PK of the laptop into the IoT device. The IoT device chooses k randomly and sends $\mathsf{Enc}(\mathsf{PK}; k)$, to the laptop.

☐ (A) MITM can decrypt the messages from the IoT device to the laptop

■ (B) MITM can decrypt the messages from the laptop to the IoT device

■ (C) At least one of the devices could accept an attacker's key that was not an old key

■ (D) MITM can make at least one of the devices to accept an old key

☐ (E) None of the above

☐ (F) ——

Solution: MITM cannot read messages from the IoT device but can provide a corrupted k' to the laptop by encrypting it under the public key of the laptop.

Q1.4 Alice manually enters the publicly-known PK of the laptop into the IoT device, and the publicly-known verification key of the IoT device into the laptop. The IoT device chooses k randomly, computes $\mathsf{Enc}(\mathsf{PK}; k)$, and sends this ciphertext to the laptop along with a signature of the ciphertext from the IoT device. The laptop verifies the signature and rejects the key if the signature fails.

☐ (G) MITM can decrypt the messages from the IoT device to the laptop

☐ (H) MITM can decrypt the messages from the laptop to the IoT device

☐ (I) At least one of the devices could accept an attacker's key that was not an old key

■ (J) MITM can make at least one of the devices to accept an old key

☐ (K) None of the above

☐ (L) ——

Solution: The MITM can replay an old key.

Q1.5 The IoT device and the laptop run Diffie-Hellman key exchange to agree on the symmetric key.

■ (A) MITM can decrypt the messages from the IoT device to the laptop

■ (B) MITM can decrypt the messages from the laptop to the IoT device

■ (C) At least one of the devices could accept an attacker's key that was not an old key

☐ (D) MITM can make at least one of the devices to accept an old key

☐ (E) None of the above

☐ (F) ——

Solution: DH is vulnerable to MITM.

Option (D) is incorrect because a MITM cannot force the new key to match an old key (without solving the discrete log problem).

Q1.6 Alice manually enters the verification key of the IoT device into the laptop. The IoT device and the laptop run Diffie-Hellman key exchange to agree on k. The IoT device signs its DH public key and sends it with a signature to the laptop as part of this exchange. The laptop verifies the signature and rejects the key if the signature fails.

■ (G) MITM can decrypt the messages from the IoT device to the laptop

☐ (H) MITM can decrypt the messages from the laptop to the IoT device

■ (I) At least one of the devices could accept an attacker's key that was not an old key

■ (J) MITM can make at least one of the devices to accept an old key

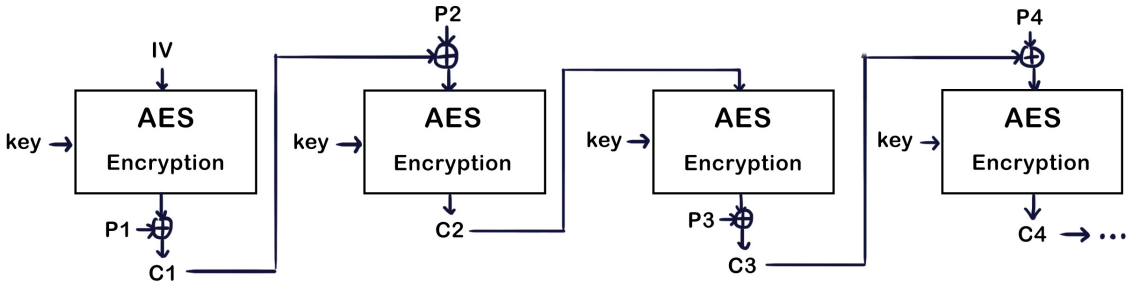
☐ (K) None of the above

☐ (L) ——

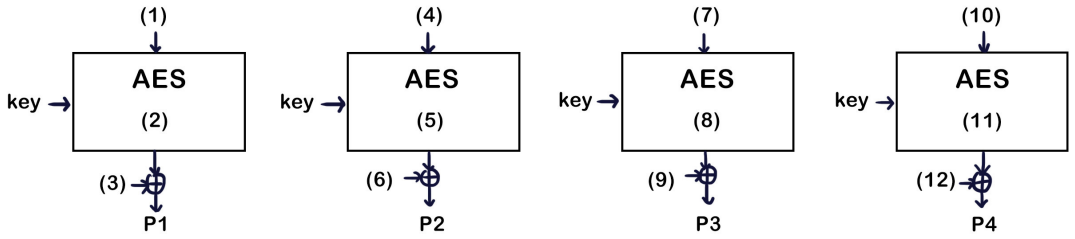
Solution: The attacker can still manipulate messages sent by the laptop.

Question 2 ***EvanBot's Last Creation*** **(15 min)**

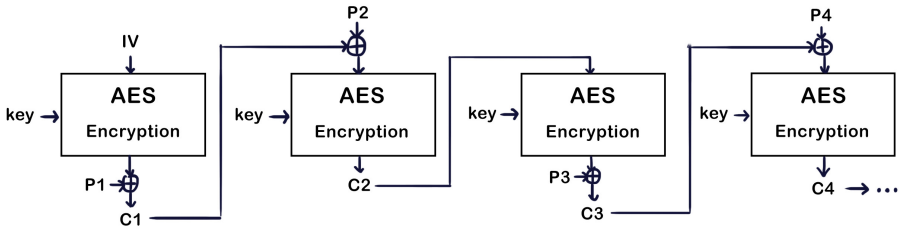
Inspired by different AES modes of operation, EvanBot creates an encryption scheme that combines two existing modes of operation and names it AES-DMO (Dual Mode Operation). Provided below is an encryption schematic of AES-DMO.

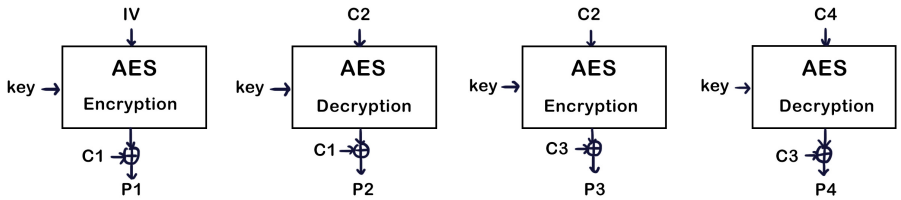


Q2.1 Fill in the numbered blanks for this incomplete decryption schematic of AES-DMO.



Solution:





Q2.2 Select all true statements about AES-DMO.

☐ (G) Encryption can be parallelized

■ (H) Decryption can be parallelized

■ (I) AES-DMO is IND-CPA secure

☐ (J) None of the above

☐ (K) ——

Solution: The diagram for encryption has a feedback from one block to the next, whereas the diagram for decryption has no such feedback. This makes decryption parallelizeable but not encryption.

DMO is IND-CPA because each block is either AES-CBC or AES-CFB, both of which are IND-CPA. You can do a proof by induction: C1 is secure since it's the first block of AES-CFB, and each subsequent block is AES-CFB or AES-CBC where the feedback from the previous block (ciphertext) is IND-CPA, in effect a random number.

Question 3 **(12 min)**

Alice comes up with a couple of schemes to securely send messages to Bob. Assume that Bob and Alice have known RSA public keys.

For this question, Enc denotes AES-CBC encryption, H denotes a collision-resistant hash function, $||$ denotes concatenation, and $\bigoplus$ denotes bitwise XOR.

Consider each scheme below independently and select whether each one guarantees confidentiality, integrity, and authenticity in the face of a MITM.

Q3.1 Alice and Bob share two symmetric keys k_1 and k_2. Alice sends over the pair $[Enc(k_1, Enc(k_2, m)), Enc(k_2, m)]$.

■ (A) Confidentiality ☐ (C) Authenticity ☐ (E) ——

☐ (B) Integrity ☐ (D) None of the above ☐ (F) ——

Solution: Note that Enc denotes AES-CBC, not AES-EMAC, so we can only provide confidentiality. An attacker can forge a pair $[Enc(k1, c1), c1]$ given $[Enc(k1, c1||c2), c1||c2]$.

Q3.2 Alice and Bob share a symmetric key k, have agreed on a PRNG, and implement a stream cipher as follows: they use the key k to seed the PRNG and use the PRNG to generate message-length codes as a one-time pad every time they send/receive a message. Alice sends the pair $[m \bigoplus \text{code}, HMAC(k, m \bigoplus \text{code})]$.

■ (G) Confidentiality ■ (I) Authenticity ☐ (K) ——

■ (H) Integrity ☐ (J) None of the above ☐ (L) ——

Solution: This stream cipher scheme has confidentiality since the attacker has no way of coming up with the pseudorandomly generated one-time pads. $HMAC$ provides the integrity and authentication.

Q3.3 Alice and Bob share a symmetric key k. Alice sends over the pair $[Enc(k, m), H(Enc(k, m))]$.

■ (A) Confidentiality ☐ (C) Authenticity ☐ (E) ——

☐ (B) Integrity ☐ (D) None of the above ☐ (F) ——

Solution: Public hash functions alone do not provide integrity or authentication. Anyone can forge a pair $c, H(c)$, which will pass the integrity check and can be decrypted.

Q3.4 Alice and Bob share a symmetric key k. Alice sends over the pair $[Enc(k, m), H(k||Enc(k, m))]$.

■ (G) Confidentiality ☐ (I) Authenticity ☐ (K) ——

☐ (H) Integrity ☐ (J) None of the above ☐ (L) ——

Solution: $H(k||Enc(k, m))$ is not a valid substitute for $HMAC$ because it is vulnerable to a length extension attack.

Question 4 **(12 min)**

EvanBot has decided to switch career paths and pursue creating new cryptographic hash functions. EvanBot proposes two new hash functions, E and B:

$$E(x) = H(x_1 x_2 \ldots x_{M-1})$$
$$B(x) = H(x_1 x_2 \ldots x_M || 0)$$

where H is a preimage-resistant and collision-resistant hash function, $x = x_1 x_2 \ldots x_M$, $x_i \in \{0, 1\}$ and $||$ denotes concatenation.

In other words, $E(x)$ calls H with the last bit of x removed, and $B(x)$ calls H with a 0 bit appended to x.

Q4.1 Is $E(x)$ preimage-resistant? Provide a counter-example if it is not.

● (A) Yes ○ (C) — ○ (E) —

○ (B) No ○ (D) — ○ (F) —

Counterexample:

Q4.2 Is $E(x)$ collision-resistant? Provide a counter-example if it is not.

○ (G) Yes ○ (I) — ○ (K) —

● (H) No ○ (J) — ○ (L) —

Counterexample:

Solution: $E(x)$ is preimage-resistant. Suppose not, i.e., given $E(x)$ we could find an x' such that $E(x) = E(x')$. We will argue this means that H is not preimage-resistant, either. Suppose we are given $H(y)$. Let $x = y0$, so that $E(x) = H(y)$. By assumption, we can find x' such that $E(x) = E(x')$. Let $y' = x'_1 \cdots x'_{M-1}$. Then it follows that $H(y) = E(x) = E(x') = H(y')$, so given $H(y)$ we can find y' such that $H(y) = H(y')$. This implies that H is not preimage resistant. That is a contradiction, so our assumption that E was not preimage-resistant must have been wrong.

$E(x)$ is not collision-resistant. Counter example: $E(1 \cdots 010) = E(1 \cdots 011)$,

Q4.3 Is $B(x)$ preimage-resistant? Provide a counter-example if it is not.

● (A) Yes ○ (C) — ○ (E) —

○ (B) No ○ (D) — ○ (F) —

Counterexample:

Q4.4 Is $B(x)$ collision-resistant? Provide a counter-example if it is not.

● (G) Yes ○ (I) — ○ (K) —

○ (H) No ○ (J) — ○ (L) —

Counterexample:

Solution:

$B(x)$ is preimage resistant, using the same reasoning as $E(x)$. (If there is an attack B's preimage-resistance, then we can construct an attack against H's preimage-resistance that succeeds half as often, which is often enough to show that H is not preimage-resistant — but we were promised that H is preimage-resistant, so it follows that B must be preimage-resistant, too.)

$B(x)$ is collision-resistant. If $B(x)$ was not collision resistant, then we can find $x \neq y$ such that $B(x) = B(y)$. This can be rewritten as $H(x||0) = H(y||0)$. Letting $x' = x'||0$ and $y' = y'||0$, this means we found $x' \neq y'$ such that $H(x') = H(y')$, which proves that $H(\cdot)$ is not collision-resistant, which is a contradiction. Thus $B(x)$ must be collision-resistant.