CS161 Summer 2026 — Introduction to Computer Security — Discussion 7

Q1 *Key Exchange Protocols* **(3 points)**

Recall that in a Diffie-Hellman key exchange, there are values a,b,g, and p. Alice computes $g^a \bmod p$ and Bob computes $g^b \bmod p$.

Q1.1 (1 point) Which of these values (a,b,g, and p) are publicly known and which must be kept private?

a	b	g	p
○ Public	○ Public	● Public	● Public
● Private	● Private	○ Private	○ Private

Q1.2 (1 point) Mallory can eavesdrop, intercept, and modify everything sent between Alice and Bob. Alice and Bob perform Diffie-Hellman to agree on a shared symmetric key K. After the exchange, Bob gets the feeling that something went wrong and calls Alice. He compares his value of K to Alice's and realizes that they are different. Explain what Mallory has done.

Solution: Mallory is performing a **man-in-the-middle attack**. Mallory pretends to be Bob when she talks to Alice, and Mallory also pretends to be Alice when she talks to Bob. In this way, both Alice and Bob are unknowingly talking to Mallory. Mallory can then decrypt/re-encrypt the traffic in both directions and modify it however she wishes to.

More technically, when Alice sends $A = g^a \bmod p$ to Bob, Mallory intercepts this (preventing it from going to Bob), and sends back to Alice: $M = g^c \bmod p$. Now when Alice sends a message to Bob, she uses $K_{\texttt{bad}} = M^a \bmod p$ which Mallory knows as $K_{\texttt{bad}} = A^c \bmod p$. Mallory can then decrypt all messages sent from Alice. She can also send messages to Alice which Alice thinks are from Bob. Mallory then does the same trick to Bob.

Q1.3 (1 point) Assume that K, the Diffie-Hellman exponents a and b, and the messages themselves are destroyed once all messages are sent. That is, these values are not stored on Alice and Bob's devices after they are done communicating.

Eavesdropper Eve records all communications between Alice and Bob, but is unable to decrypt them. At some point in the future, Eve is lucky and manages to compromise Bob's computer.

Is the confidentiality of Alice and Bob's prior **Diffie-Hellman**-based communication in jeopardy? Explain why.

○ Yes ● No

Solution: No. Since Alice and Bob destroy the DH exponents a and b after use, and since the key computed from them itself is never transmitted, there is no information present on Bob's computer that Eve can leverage to recover K. This means that with Diffie-Hellman key exchanges, later compromises in no way harm the confidentiality of previous communication, even if the ciphertext for that communication was recorded in full. This property is called *Perfect Forward Secrecy*.

Q2 *Ra's al Gamal* **(3 points)**

Recall the ElGamal scheme from lecture:

- $\mathsf{KeyGen}() = \big(b, B = g^b \bmod p\big)$
- $\mathsf{Enc}(B, M) = (C_1 = g^r \bmod p, C_2 = B^r \times M \bmod p)$

Q2.1 (1 point) Is the ciphertext (C_1, C_2) decryptable by someone who knows the private key b? If you answer yes, provide a decryption formula. You may use C_1, C_2, b, and any public values.

● Yes ○ No

Solution: The decryption formula is $M = C_1^{-b} \times C_2$.

Q2.2 (1 point) Consider an adversary that can efficiently break the discrete log problem. Can the adversary decrypt the ciphertext (C_1, C_2) without knowledge of the private key? If you answer yes, briefly state how the adversary can decrypt the ciphertext.

● Yes ○ No

Solution: An adversary that can break the discrete log problem can recover r from $C_1 = g^r$ or b from $B = g^b$, so they can compute g^{br} and recover the original message.

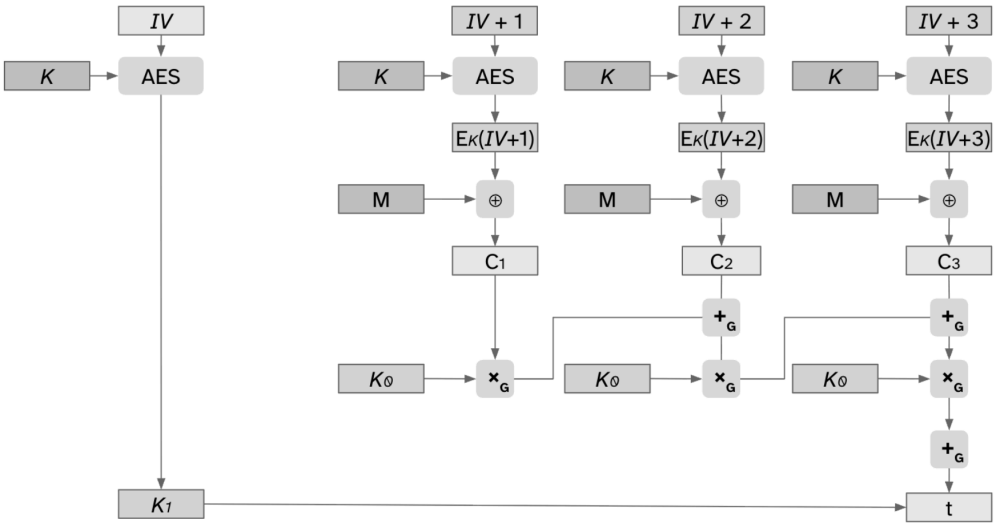
Q2.3 (1 point) Consider an adversary that can efficiently break the Diffie-Hellman problem. Can the adversary decrypt the ciphertext (C_1, C_2) without knowledge of the private key? If you answer yes, briefly state how the adversary can decrypt the ciphertext.

● Yes ○ No

Solution: An adversary that can break the Diffie-Hellman problem can recover g^{br} from $C_1 = g^r$ and $B = g^b$, so they can recover the original message.

Q3 *AES-GCM* **(6 points)**

Consider the **AES-GCM** tag-generation mechanism where plaintext blocks are encrypted under AES-CTR to produce ciphertext $C_1, C_2, ..., C_n$,. Then, the tag t is computed by combining these ciphertext blocks with the sub-keys $K_0 = H = E_K(0)$ and $K_1 = E_K(IV)$.



Assume there is no associated data, and assume the length block is omitted from the GHASH computation. So for an n-block ciphertext:

$$t = C_1 K_0^n \oplus C_2 K_0^{n-1} \oplus ... \oplus C_n K_0 \oplus K_1$$

Q3.1 (1 point) In AES-GCM, what is the value of t for a 3-block message (M_1, M_2, M_3)?

- ○ $t = C_3 K_0^3 \oplus C_2 K_0^2 \oplus C_1 K_0 \oplus K_1$
- ● $t = C_1 K_0^3 \oplus C_2 K_0^2 \oplus C_3 K_0 \oplus K_1$
- ○ $t = C_1 K_1^3 \oplus C_2 K_1^2 \oplus C_3 K_1 \oplus K_0$
- ○ $t = C_3 K_1^3 \oplus C_2 K_1^2 \oplus C_1 K_1 \oplus K_0$

Solution:

$t = C_1 K_0^n \oplus C_2 K_0^{n-1} \oplus ... \oplus C_n K_0 \oplus K_1$

Let n = 3

$t = C_1 K_0^3 \oplus C_2 K_0^2 \oplus C_3 K_0 \oplus K_1$

Q3.2 (5 points) Assume you have access to a encryption oracle which takes your message M and outputs $M \oplus E_K(IV + 1)$. Now suppose the IV is reused across every ciphertext in this question, so K_0 and K_1 are the same fixed (unknown) values throughout. You know that $C = (C_1, C_2, C_3)$ has tag t. K is unknown.

What should the first ciphertext be?

0

Solution:

For n = 1:
$t = C_1 K_0 \oplus K_1 = K_1$

If we use 0, we will get a symbolic representation of K_1

$$t_a = 0 \times K_0 \oplus K_1 = K_1$$
$$\therefore K_1 = t_a$$

The MAC of the message in the box above is t_a.

What should the second ciphertext be?

1

Solution:

If we use 1, we will get a symbolic representation of K_0 in terms of known variables.

$$t_b = 1 \times K_0 \oplus K_1 = K_0 \oplus K_1 = t_a \oplus K_1$$
$$\therefore K_0 = t_a \oplus t_b$$

The MAC of the message in the box above is t_b.

Now, construct a three-block message C' with the same tag as C (i.e. with the tag t):

$C_{\text{new}} = \Big($ [0] First block , [0] Second block , [$(t \oplus t_a) \div (t_a \oplus t_b)$] Third block $\Big)$

Solution: With both K_0 and K_1 known symbolically, a valid tag for any chosen ciphertext can be computed — a full key compromise, not just an extension trick. Choosing C_1' = C_2' = 0 for simplicity: $t = C_3' \times K_0 \oplus t$ a, solving to D3 = (t ⊕ t a) ÷ (ta ⊕ tb)