CS 161 Spring 2025 — Introduction to Computer Security — Discussion 5

Question 1 ***MAC Madness***

Evan wants to store a list of every CS161 student's firstname and lastname, but he is afraid Mallory will tamper with his list.

Evan is considering adding a cryptographic value to each record to ensure its integrity. For each scheme, determine what Mallory can do without being detected.

Assume MAC is a secure MAC, H is a cryptographic hash, and Mallory does not know Evan's secret key k. Assume that firstname and lastname are all lowercase and **alphabetic** (no numbers or special characters), and concatenation does not add any delimiter (e.g. a space or tab), so nick||weaver = nickweaver.

Q1.1 (3 points) H(firstname||lastname)

- (●) (A) Mallory can modify a record to be a value of her choosing
- (○) (B) Mallory can modify a record to be a specific value (not necessarily of her choosing)
- (○) (C) Mallory cannot modify a record without being detected
- (○) (D) ——
- (○) (E) ——
- (○) (F) ——

Solution: Anybody can hash a value, so Mallory could change a record to be whatever she wants and compute the hash of her new record.

Q1.2 (3 points) $\mathsf{MAC}(k, \mathsf{firstname} \| \mathsf{lastname})$

Hint: Can you think of two different records that would have the same MAC?

◯ (G) Mallory can modify a record to be a value of her choosing

● (H) Mallory can modify a record to be a specific value (not necessarily of her choosing)

◯ (I) Mallory cannot modify a record without being detected

◯ (J) ——

◯ (K) ——

◯ (L) ——

Solution: Because the concatenation doesn't have any indicator of where the first name ends and the last name begins, Mallory could shift some letters between the first name and last name. For example, she could change the name Nick Weaver to Ni Ckweaver, Nic Kweaver, Nickw Eaver, etc. Since the MAC would remain unchanged, this edit would be undetectable.

Q1.3 (3 points) $\mathsf{MAC}(k, \mathsf{firstname} \| \texttt{"-"} \| \mathsf{lastname})$, where `"-"` is a hyphen character.

◯ (A) Mallory can modify a record to be a value of her choosing

◯ (B) Mallory can modify a record to be a specific value (not necessarily of her choosing)

● (C) Mallory cannot modify a record without being detected

◯ (D) ——

◯ (E) ——

◯ (F) ——

Solution: Since the names are alphabetic, they would never include a dash in them. Hence, the dash serves as a separator between first name and last name, so the attack from the previous part is no longer possible.

Q1.4 (3 points) $\mathsf{MAC}(k, \mathsf{H}(\mathsf{firstname})\|\mathsf{H}(\mathsf{lastname}))$

◯ (G) Mallory can modify a record to be a value of her choosing

◯ (H) Mallory can modify a record to be a specific value (not necessarily of her choosing)

● (I) Mallory cannot modify a record without being detected

◯ (J) ——

◯ (K) ——

◯ (L) ——

Solution: Because the hashes produce a fixed-length value, concatenating them within the MAC without delimiters does not violate integrity.

Q1.5 (3 points) $\mathsf{MAC}(k, \mathsf{firstname})\|\mathsf{MAC}(k, \mathsf{lastname})$

◯ (A) Mallory can modify a record to be a value of her choosing

● (B) Mallory can modify a record to be a specific value (not necessarily of her choosing)

◯ (C) Mallory cannot modify a record without being detected

◯ (D) ——

◯ (E) ——

◯ (F) ——

Solution: Because the first name and last name have separate MACs, Mallory could swap the first name and last name, and swap the two halves of the MAC.

In other words, Mallory could change the name Nick Weaver to Weaver Nick, and change the MAC from $\mathsf{MAC}(k, \mathsf{nick})\|\mathsf{MAC}(k, \mathsf{weaver})$ to $\mathsf{MAC}(k, \mathsf{weaver})\|\mathsf{MAC}(k, \mathsf{nick})$.

Q1.6 (3 points) Which of Evan's schemes guarantee confidentiality on his records?

◯ (G) All 5 schemes

◯ (H) Only the schemes with a MAC

◯ (I) Only the schemes with a hash

● (J) None of the schemes

◯ (K) ——

◯ (L) ——

Solution: MACs and hashes do not have any confidentiality guarantees.

Question 2 ***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 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 $\mathsf{c}_2 = \mathsf{Hash}(\mathsf{c}_1)$
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 $\mathsf{c}_2 = \mathsf{MAC}(\mathsf{K}, \mathsf{c}_1)$

Q2.1 In which schemes can Bob successfully decrypt m given c?

> **Solution:** Bob cannot decrypt Scheme 1 because he cannot invert Hash.
>
> ***In sum:*** 2-4

Q2.2 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?

> **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

Q2.3 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?

> **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

Q2.4 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?

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

In sum: 4, we replace message m with timestamp $\|$ m.

Question 3 ***Key Exchange Protocols***

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 $g^b \bmod p$.

Q3.1 Which of these values (a, b, g, and p) are publicly known and which must be kept private?

> **Solution:**
>
> g and p are publicly known. Implementations of Diffie-Hellman often have carefully picked values of g and p which are known to everyone. Alice and Bob must keep a and b secret respectively.

Q3.2 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 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 $K_{bad} = M^a \bmod p$ which Mallory knows as $K_{bad} = A^c \bmod p$. 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.

Now consider the following key exchange protocol which can be used by Alice (A) and Bob (B) to agree upon a shared key, K.

Some additional details:

- 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.

Q3.3 Is the confidentiality of Alice and Bob's prior **Diffie-Hellman**-based communication in jeopardy?

> **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*.