CS 161 Spring 2026 — Introduction to Computer Security — Discussion 5

Q1 *MAC Madness* **(18 points)**

Evan wants to store a list of every CS161 student's `firstname` and `lastname`, but they are afraid that Mallory will tamper with their 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) $\mathsf{H}(\texttt{firstname}\|\texttt{lastname})$

- ◯ Mallory can modify a record to be a value of her choosing
- ◯ Mallory can modify a record to be a specific value (not necessarily of her choosing)
- ◯ Mallory cannot modify a record without being detected

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

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

- ◯ Mallory can modify a record to be a value of her choosing
- ◯ Mallory can modify a record to be a specific value (not necessarily of her choosing)
- ◯ Mallory cannot modify a record without being detected

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

- ◯ Mallory can modify a record to be a value of her choosing
- ◯ Mallory can modify a record to be a specific value (not necessarily of her choosing)
- ◯ Mallory cannot modify a record without being detected

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

- ◯ Mallory can modify a record to be a value of her choosing
- ◯ Mallory can modify a record to be a specific value (not necessarily of her choosing)
- ◯ Mallory cannot modify a record without being detected

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

- ◯ Mallory can modify a record to be a value of her choosing
- ◯ Mallory can modify a record to be a specific value (not necessarily of her choosing)
- ◯ Mallory cannot modify a record without being detected

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

- ◯ All 5 schemes
- ◯ Only the schemes with a hash
- ◯ Only the schemes with a MAC
- ◯ None of the above

Q2 *EU-CMA* **(9 points)**

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

- Symmetric Encryption:
  - Encryption: $\mathsf{Enc}(K, m)$
  - Decryption: $\mathsf{Dec}(K, m)$
- Cryptographic Hash Function: $\mathsf{Hash}(m)$
- MAC: $\mathsf{MAC}(K, m)$

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.

1. $t_i = \mathsf{Hash}(t_{i-1}, m_i), t_0 = \mathsf{Hash}(K)$
2. $t = \mathsf{Hash}(m_i)$
3. $t = \mathsf{MAC}(K, m)$

Q2.1 (3 points) Is scheme 1 EU-CMA secure?

◯ Yes (leave the boxes blank)

◯ No (fill in the boxes)

m : t :

Q2.2 (3 points) Is scheme 2 EU-CMA secure?

◯ Yes (leave the boxes blank)

◯ No (fill in the boxes)

m : t :

Q2.3 (3 points) Is scheme 3 EU-CMA secure?

◯ Yes (leave the boxes blank)

◯ No (fill in the boxes)

m : t :

Q3 *Why do RSA signatures need a hash?* **(3 points)**

To generate RSA signatures, Alice first creates a standard RSA key pair: (n, e) is the RSA public key and d is the RSA private key, where n is the RSA modulus. For standard RSA signatures, we typically set e to a small prime value such as 3; for this problem, let $e = 3$.

Suppose we used a **simplified** scheme for RSA signatures that skips using a hash function and instead uses message M directly, so the signature S on a message M is $S = M^d \bmod n$. In other words, if Alice wants to send a signed message to Bob, she will send (M, S) to Bob where $S = M^d \bmod n$ is computed using her private signing key d.

Q3.1 (1 point) With this **simplified** RSA scheme, how can Bob verify whether S is a valid signature on message M? In other words, what equation should he check, to confirm whether M was validly signed by Alice?

Q3.2 (1 point) Mallory learns that Alice and Bob are using the **simplified** signature scheme described above and decides to trick Bob into believing that one of Mallory's messages is from Alice. Explain how Mallory can find an (M, S) pair such that S will be a valid signature on M.

You should assume that Mallory knows Alice's public key n, but not Alice's private key d. The message M does not have to be chosen in advance and can be gibberish.

Q3.3 (1 point) Is the attack in Q1.2 possible against the **standard** RSA signature scheme (the one that includes the cryptographic hash function)? Why or why not?